SIEMENS

SIMATIC

SIMIT Simulation Platform (V10.0)

Bedienhandbuch

Vorwort

Grundlagen von SIMIT 1

Kopplungen 2

Simulationsmodell 3

Automatische Modellerstellung 4

Diagnose & Visualisierung 5

Skripte 6

Komponententypeditor 7

Bibliotheken 8

Menüs und Dialogfelder 9

06/2018
A5E44876249-AA

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

⚠ GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠ WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠ VORSICHT
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

⚠ WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Siemens AG
Division Process Industries and Drives
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

A5E44876249-AA
℗ 03/2019 Änderungen vorbehalten

Copyright © Siemens AG 2018.
Alle Rechte vorbehalten

Vorwort

Security-Disclaimer für Simulationsprodukte

Mit den Simulationsprodukten von Siemens können Planung und Betrieb einer Anlage/ Maschine simuliert und/oder optimiert werden. Bei den Simulations- und Optimierungsergebnissen handelt es sich lediglich um unverbindliche Empfehlungen, die von der Vollständigkeit und Richtigkeit der Eingangsdaten abhängig sind. Die Eingangsdaten und die Ergebnisse sind daher bei jeder Simulation/Optimierung vom Anwender auf Plausibilität zu prüfen.

Security-Hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z. B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie im Internet (http://www.siemens.com/industrialsecurity).

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed (http://www.siemens.com/industrialsecurity).

Was ist neu?

SIMIT V10.0

Die Version V10.0 enthält gegenüber der Vorgängerversion V9.1 folgende Erweiterungen oder Änderungen:

- **Neues Lizenzmodell**
  Eine Lizenz nach der Anzahl der Simulation-Tags in einem Simulationsprojekt ersetzt die bisherigen Produktvarianten SIMIT Standard, SIMIT Professional und SIMIT Ultimate. Lizenzen werden auf einem Wibu-Dongle verwaltet.
- **Änderungen in DEMO-Modus**
  Die DEMO-Version starten Sie durch Doppelklick auf die Bildschirm-Verknüpfung SIMIT SP Demo. Die DEMO-Version entspricht dem Funktionsumfang von SIMIT-Variante "S".
- **Neue Betriebsart "bus-synchron"**
  Die bus-synchrone Betriebsart ergänzt die bisherigen Betriebsarten asynchron und synchron. Die neue Betriebsart steht für die Kopplungen PLCSIM Advanced und MCD zur Verfügung. Bei bus-synchroner Betriebsart stellt SIMIT sicher, dass alle an der Simulation beteiligten Komponenten den gleichen synchronisierten Simulationsfortschritt haben.
- **Zeitscheiben**
  Die minimal einstellbare Zykluszeit ändert sich von 10 ms auf 1 ms.
  Für jede Zeitscheibe gibt es jetzt acht Unterzeitscheiben. Die Unterzeitscheiben ermöglichen eine bessere Aufteilung der Simulation, um Mehrkernprozessoren besser auszulasten.
- **Neue Funktionalität "Verteilte Simulation"**
  Die verteilte Simulation ermöglicht das Verteilen der Rechenlast auf mehrere Rechner, die vom Master-Rechner ferngesteuert werden.
- **Änderungen bei der Kopplung "SIMIT Unit"**
  Shared Devices sind jetzt mit SIMIT möglich.
  Alarme simulieren Sie entweder in den Eigenschaften einer Station oder über die Komponente "SetInterrupt".
  Der Gerätestatus wird aus der SIMIT Unit-Hardware ermittelt und im Hardware-Baum dargestellt.
- **Änderungen bei der Kopplung "Virtual Controller"**
  Der Virtual Controller unterstützt jetzt auch die Systemfunktion SFC 60 und das Protokoll S7H.
- **Neue Kopplung "MCD"**
  SIMIT Simulation Platform V10.0 unterstützt die Kopplung zu einer Mechatronics Concept Designer-Anwendung.
- **Exerne Kopplungen**
  Sie können eigene programmierte Kopplungen an SIMIT anbinden.
- **Vorlagen instanziieren**
  Das Dialogfeld "Instanziiere Vorlagen" fasst den Tabellenimport, den IEA-Import, den CMT-Import und die Generierung der Geräteebene zusammen.
- **Tabellenimport**
  Beim Tabellenimport gibt es keine Profile zur Auswahl mehr. Das frühere Profil 2 mit Platzhaltern in der ersten Zeile ist jetzt Standard.

- **Änderungen in der Basisbibliothek**
  Neue Komponenten:
  - DataAdaption
  - DP-DP-ID
  - DynamicServoControl
  - GeneralDrive
  - InFeed_Tel_370
  - Multi HART 8
  - Multi HART 16
  - PROFIdrive1
  - PROFIdrive2
  - Safety30
  - Safety31
  - Safety32
  - SafetyProcess
  - Sensor
  - SensorProcessLinear
  - SensorProcessRotatory
  - SetInterrupt
  - SiemensMomentumReduction

  Geänderte Komponenten:
  - Sinamics
  - Universal

  Entfernte Komponenten:
  - DCMaster
  - Masterdrive
  - Masterdrive3
  - Masterdrive4
  - PROFIdrive
- **Änderungen in der Bibliothek "CHEM-BASIC"**
  Neue Komponenten:
  - Column Tower – Kolonnenturm
  - Evaporator – Verdampfer
  - Flash – 2-Phasen Abscheider

  Geänderte Komponenten:
  - Condenser – Kondensator

– StorageTankLiquid – Speichertank für flüssige Medien

Lizenzierung

Lizenzmodell

Grundlage des Lizenzmodells von SIMIT sind Simulation-Tags. Simulation-Tags berechnen sich aus der Summe der Ein- und Ausgänge sowie der Zustandsgrößen aller verwendeten Komponenten im Simulationsprojekt. Da eine implizite Signalverschaltung zwischen zwei Kopplungen die Funktion eines Diagramms übernimmt, auf dem diese Signale über eine Komponente mit mindestens einem Ein- und einem Ausgang verbunden sind, werden solche impliziten Verschaltungen in Kopplungen entsprechend mit 2 Simulation-Tags gezählt.
Die folgende Tabelle zeigt die verfügbaren SIMIT-Varianten:

SIMIT-Variante	Anzahl Simulation-Tags
XS[1]	bis zu 250
S	bis zu 2.500
M	bis zu 15.000
L	bis zu 200.000
XL	bis zu 1.000.000

[1] Nur in Verbindung mit SIMIT Unit-Hardware

Zusätzliche Lizenzen

Zusätzliche Lizenzen werden für Folgendes benötigt:

- Eine Aktualisierung auf eine neue SIMIT-Version
- Weitere Bibliotheken (CHEM-BASIC, FLOWNET oder CONTEC)
- Component Type Editor (CTE)
- SIMIT Virtual Controller (VC)

Verhalten in SIMIT

Während des Startens von SIMIT wählen Sie die Größenvariante. Sie können die Größenvariante auch nachträglich ändern über Portalansicht > Start > Größenvariante.

Wie viel Prozent der erlaubten Simulation-Tags Ihr Simulationsprojekt enthält, wird in der Statuszeile angezeigt.

Anzahl Simulation-Tags ausgeschöpft zu	Folge
80 %	Einmaliger Hinweis
110 % und mehr[1]	Hinzufügen von Simulation-Tags möglich: Nein Speichern des Simulationsprojekts möglich: Nein Starten der Simulation möglich: Nein

[1] SIMIT-Variante XL: Keine Beschränkung. Die Leistungsfähigkeit des benutzten PC bestimmt die Maximalgröße eines Simulationsprojekts.

WIBU-Lizenzserver einrichten

Einleitung

Wenn der Dongle am lokalen Computer steckt, greift SIMIT direkt darauf zu.

Wenn der Dongle an einem Computer im Netzwerk steckt, richten Sie den Zugriff auf den Dongle ein. Nutzen Sie dazu auf dem Server und auf dem lokalen Computer das Programm CodeMeter.

Voraussetzung

- Der Dongle steckt nicht am lokalen Computer, sondern an einem Computer im Netzwerk.

Vorgehen

Um die Hilfe zu öffnen, gehen Sie folgendermaßen vor:

1. Navigieren Sie zu "Start > CodeMeter > CodeMeter Control Center".
   Das "CodeMeter Kontrollzentrum" wird geöffnet.
2. Klicken Sie auf "WebAdmin".
3. Klicken Sie auf das Icon "?".
   Die "CodeMeter Anwenderhilfe" wird geöffnet.
4. Führen Sie die unter "Server-Zugriff" und unter "Lizenz-Zugriffsberechtigungen" genannten Schritte durch.

Ergebnis

SIMIT greift auf eine Lizenz zu, die sich auf einem Dongle befindet, der an einem Computer im Netzwerk steckt.

Inhaltsverzeichnis

Vorwort **3**

1 Grundlagen von SIMIT **21**

1.1 Funktionsweise 21

1.2 DEMO-Modus 21

1.3 Starten von SIMIT 23

1.4 Die Oberfläche von SIMIT 26

1.5 Erstellen einer Simulation 28
1.5.1 Aufbau einer Simulation 28
1.5.2 Visualisieren einer Simulation 31
1.5.3 Visualisieren von Kopplungssignalen 34

1.6 Ablauf einer Simulation 36
1.6.1 Die Zustände einer Simulation 36
1.6.2 Bedienen und Beobachten der Simulation 39
1.6.2.1 Aktionen bei laufender Simulation 39
1.6.2.2 Anzeige der Simulationslast 40
1.6.2.3 Änderungen im Simulationsprojekt bei laufender Simulation 42
1.6.3 Einstellungen zum Simulationsablauf 43
1.6.3.1 Betriebsarten 43
1.6.3.2 Zeitscheiben 48
1.6.3.3 Beschleunigung und Verzögerung der Simulation 49
1.6.3.4 Einstellbare Time-Out-Zeiten 50

1.7 Verteilte Simulation 51
1.7.1 Funktionsweise der verteilten Simulation 51
1.7.2 Projektierung einer verteilten Simulation 52
1.7.3 Datenaustausch bei einer verteilten Simulation 53
1.7.4 Firewall einrichten 54
1.7.5 Bildschirm-Verknüpfung "SIMIT Client" anlegen 55
1.7.6 Verteilte Simulation starten 56
1.7.7 Verteilte Simulation auf einem Client-PC ändern 57

2 Kopplungen **59**

2.1 Kopplungskonzept 59
2.1.1 Kopplungen in der Architektur von SIMIT 59
2.1.2 Kopplungen mit SIMATIC-Steuerungen 61

2.2 Kopplungseditor 63

2.3 HWCNExport-Datei erstellen 65

2.4 Kopplungen deaktivieren 66

2.5 SIMIT Unit 67
2.5.1 Funktionsweise der SIMIT Unit-Kopplung 67
2.5.2 Unterstützte PROFIBUS DP-Konfigurationen 69

2.5.3 Unterstützte PROFINET IO-Konfigurationen 76
2.5.4 Redundante und fehlersichere Systeme 81
2.5.4.1 Redundante Konfigurationen (H-Systeme) 81
2.5.4.2 Fehlersichere Konfigurationen (F-Systeme) 86
2.5.4.3 Redundante, fehlersichere Konfigurationen (HF-Systeme) 87
2.5.5 SIMIT Unit einrichten 87
2.5.5.1 SIMIT Unit konfigurieren 87
2.5.5.2 Gerätebeschreibungsdatei in SIMIT importieren 88
2.5.5.3 Firmware der SIMIT Unit aktualisieren 88
2.5.6 Kopplung vom Typ "SIMIT Unit" projektieren 89
2.5.6.1 Kopplung vom Typ "SIMIT Unit" anlegen 89
2.5.6.2 Station anlegen 90
2.5.6.3 Kopieren der Systemdatenbausteine 91
2.5.6.4 Hardwarekonfiguration in Station importieren 91
2.5.6.5 Hardwarekonfiguration für Shared Devices importieren 93
2.5.6.6 Station konfigurieren 94
2.5.6.7 Simulation von Alarmen 96
2.5.6.8 Gerätestatus 98
2.5.6.9 Strang, Gerät oder Modul deaktivieren 98
2.5.6.10 Konfiguration in Station laden 99
2.5.6.11 Eigenschaften der SIMIT Unit-Kopplung 100
2.5.6.12 XML-Importschnittstelle für Hardware-Konfiguration von Drittsystemen 103

2.6 Virtual Controller 109
2.6.1 Funktionsweise des Virtual Controller 109
2.6.1.1 Einleitung 109
2.6.1.2 Anforderungen an ein Simulationsnetzwerk 113
2.6.1.3 Funktionen 114
2.6.1.4 Grundlagen zur Virtual Controller-Kommunikation 115
2.6.1.5 Unterstützte Systemfunktionen 117
2.6.1.6 Unterstützte S7-Bausteine 119
2.6.1.7 Unterstützte Dienste 120
2.6.1.8 Behandlung von Synchronfehlern 123
2.6.1.9 Ablaufsteuerung 124
2.6.1.10 Datensatzkommunikation 126
2.6.1.11 Ersatzmechanismus für Datenaustausch 126
2.6.1.12 Hinweise zum Einsatz eines Virtual Controller 127
2.6.2 Konfigurieren des Virtual Controller 128
2.6.2.1 Virtual Controller projektieren 128
2.6.2.2 Virtual Controller anlegen 130
2.6.2.3 STEP 7-/PCS 7-Projekt importieren 131
2.6.2.4 Virtual Controller konfigurieren 133
2.6.2.5 Verteilungseditor (Virtual Controller) 133
2.6.2.6 Zusätzlichen PC konfigurieren 137
2.6.2.7 Virtual Controller während Simulation 138
2.6.3 Editieren von Signalen 140
2.6.3.1 Eigenschaften der Virtual Controller-Kopplung 140
2.6.3.2 DB-Daten aus AWL-Quellen importieren 143

2.7 PLCSIM Advanced-Kopplung 145
2.7.1 Funktionsweise der PLCSIM Advanced-Kopplung 145
2.7.2 Kopplung vom Typ "PLCSIM Advanced" anlegen 145
2.7.3 Hardwarekonfiguration in Station importieren 146

2.7.4 Station einer PLCSIM Advanced-Kopplung konfigurieren 147
2.7.5 Verteilungseditor (PLCSIM Advanced) 148

2.8 PLCSIM-Kopplung 150
2.8.1 Funktionsweise der PLCSIM-Kopplung 150
2.8.2 Konfigurieren der PLCSIM-Kopplung 150
2.8.2.1 Anlegen einer PLCSIM-Kopplung 150
2.8.2.2 Konfigurieren der E/A-Signale in der PLCSIM-Kopplung 150
2.8.2.3 Eigenschaften der PLCSIM-Kopplung 151

2.9 OPC-Kopplung 152
2.9.1 Funktionsweise der OPC-Kopplung 152
2.9.2 Signalübertragung zwischen OPC-Server und OPC-Client 153
2.9.3 Abbildung der Signale und Datentypen eines OPC-Servers 155
2.9.4 Status von OPC-Servern 155
2.9.5 OPC DA-Client-/Server-Kopplung projektieren 156
2.9.5.1 Funktionsweise des SIMIT OPC DA-Servers 156
2.9.5.2 SIMIT OPC DA-Server-Kopplung konfigurieren 158
2.9.5.3 OPC DA-Client-Kopplung konfigurieren 158
2.9.5.4 Eigenschaften der SIMIT OPC DA-Server-Kopplung 159
2.9.5.5 Eigenschaften der OPC DA-Client-Kopplung 160
2.9.5.6 Hinweise zur DCOM-Konfiguration 161
2.9.6 OPC UA-Client-Kopplung projektieren 163
2.9.6.1 Funktionsweise der OPC UA-Client-Kopplung 163
2.9.6.2 Zertifikate 163
2.9.6.3 OPC UA-Client-Kopplung konfigurieren 165
2.9.6.4 Fehlermeldungen beim Verbindungsaufbau zum OPC UA-Server 166
2.9.6.5 Eigenschaften der OPC UA-Client-Kopplung 167

2.10 Shared Memory-Kopplung 168
2.10.1 Funktionsweise der SHM-Kopplung 168
2.10.1.1 Zugriff auf den Speicherbereich 168
2.10.1.2 Struktur des Speicherbereichs 169
2.10.1.3 Anlegen des Speicherbereichs 172
2.10.2 Konfigurieren der SHM-Kopplung 173
2.10.2.1 Anlegen einer SHM-Kopplung 173
2.10.2.2 Konfigurieren der Signale in der SHM-Kopplung 173
2.10.2.3 Signaleigenschaften in der SHM-Kopplung 175
2.10.2.4 Eigenschaften der SHM-Kopplung 175
2.10.2.5 Import und Export der Signale 176

2.11 PRODAVE-Kopplung 177
2.11.1 Funktionsweise der PRODAVE-Kopplung 177
2.11.2 Konfigurieren der PRODAVE-Kopplung 178
2.11.2.1 Anlegen einer PRODAVE-Kopplung 178
2.11.2.2 Editieren von Signalen in der PRODAVE-Kopplung 178
2.11.2.3 Eigenschaften der PRODAVE-Kopplung 179
2.11.2.4 Importieren der Signaleigenschaften 180
2.11.2.5 Exportieren der Signaleigenschaften 180

2.12 MCD-Kopplung 180
2.12.1 Funktionsweise der MCD-Kopplung 180
2.12.2 Einsatzszenarien der MCD-Kopplung 181
2.12.3 Konfigurieren der MCD-Kopplung 183
2.12.3.1 MCD-Kopplung anlegen 183

2.12.3.2 Signale importieren 184
2.12.3.3 Signale verknüpfen 185
2.12.3.4 Signaleigenschaften in der MCD-Kopplung bearbeiten 185
2.12.3.5 Bus-synchrone MCD-Kopplung konfigurieren 186
2.12.3.6 Eigenschaften der MCD-Kopplung 187

2.13 Signale einer Kopplung bearbeiten 188
2.13.1 Grundlagen zu Signalen 188
2.13.1.1 Datenrichtung der Kopplung 188
2.13.1.2 Bedeutung des Kopplungsnamens 188
2.13.1.3 Sortieren und Filtern von Signalen im Kopplungseditor 189
2.13.1.4 Adressierung von Signalen 190
2.13.1.5 Fixieren von Signalen im Kopplungseditor 190
2.13.1.6 Abbildung von SIMATIC-Datentypen in SIMIT 193
2.13.1.7 Zugriff auf einen Datensatz oder Speicherbereich 194
2.13.1.8 Umwandeln der Datenbreite von Signalen 196
2.13.1.9 Normierung von analogen Signalen 196
2.13.2 Import und Export der Signale 199
2.13.2.1 Dateiformate für Signale 199
2.13.2.2 Symboltabelle 200
2.13.2.3 Variablentabelle 201
2.13.2.4 Signaltabelle 202
2.13.2.5 INI-Format der OPC-Kopplungen 203
2.13.2.6 Signaleigenschaften importieren 204
2.13.2.7 Import von PLC-Variablen-Listen 205
2.13.2.8 Signaleigenschaften exportieren 206
2.13.3 Verwendung von Peripheriesignalen 207
2.13.3.1 Peripheriesignale zur Verschaltung auf Diagrammen 207
2.13.3.2 Peripheriesignale für Animationen 208
2.13.3.3 Peripheriesignale zur Bedienung auf Diagrammen 209
2.13.3.4 Peripheriesignale in Kurvenbildern 210
2.13.3.5 Mehrfachverwendung von Peripherie-Konnektoren 210
2.13.4 Signal verschalten 210
2.13.5 Signal aufspalten 211
2.13.6 Signale zusammenfassen 212
2.13.7 Signal vorbelegen 213
2.13.8 Signal fixieren 214
2.13.9 Signal symbolisch adressieren 214
2.13.10 Signal normieren 215
2.13.11 Normierung auf anderes Signal übertragen 216
2.13.12 Signal begrenzen 217
2.13.13 Signale von einem OPC-Server lesen 218
2.13.14 Signal als "Rücklesbar" konfigurieren 219

3 Simulationsmodell 221

3.1 Projektmanager 221
3.1.1 Ansicht und Funktionen des Projektmanagers 221
3.1.2 Versionierung 222
3.1.3 Schreibschutz 223

3.2 Diagrammeditor 225
3.2.1 Erstellen und Bearbeiten von Diagrammen 225
3.2.2 Grafiken visualisieren 227

3.2.3 Signale visualisieren 231
3.2.4 Drucken von Diagrammen 231

3.3 Task-Cards....232
3.3.1 Task-Card "Komponenten" 232
3.3.2 Task-Card "Controls" 234
3.3.3 Task-Card "Makros" 236
3.3.4 Task-Card "Grafik" 238
3.3.5 Task-Card "Vorlagen" 239
3.3.6 Task-Card "Projekte"....240
3.3.7 Task-Card "Signale"....242

3.4 Makrokomponenteneditor 243
3.4.1 Makroeditor 243
3.4.2 Trennlinien zwischen Anschlüssen von Makrokomponenten einfügen....246
3.4.3 Topologische Anschlüsse von Makrokomponenten....247
3.4.4 Eingänge von Makrokomponenten vorbelegen....248
3.4.5 Parameter von Makrokomponenten definieren 248
3.4.6 Eigenschaften von Makrokomponenten....249
3.4.7 Suchen und Ersetzen in Makrokomponenten 249
3.4.8 Makrokomponenten verwenden....250

3.5 Migration von Projekten aus älteren SIMIT-Versionen....250

4 Automatische Modellerstellung....251

4.1 Vorlagen....252
4.1.1 Suchen und Ersetzen in Vorlagen....255
4.1.2 Vorlagen instanziieren mit Eingabe von Ersetzungen....255
4.1.3 Anlegen von Tabellen aus der Vorlage heraus 256
4.1.4 Angabe einer Ordnerhierarchie für Vorlagen 257
4.1.5 Adressierung eines Moduls über E/A-Adresse 258
4.1.6 Indirekte Adressierung 258
4.1.7 Öffnen von Basisvorlagen im Editor....259

4.2 Instanziierung von Vorlagen aus Dateien oder dem Simulationsmodell....259
4.2.1 Vorlagen aus Dateien instanziieren 259
4.2.2 Tabellenimport 261
4.2.3 IEA-Import....264
4.2.4 CMT-Import....266
4.2.5 Vorlagen aus dem Simulationsmodell instanziieren....269

4.3 Automatisierter Import....270
4.3.1 Diagramme aus ZIP- oder XML-Dateien importieren....270
4.3.2 Syntax der XML-Datei....271
4.3.3 Beispiele für XML-Dateien 276
4.3.3.1 Beispiel für eine einfache Vorlageninstanziierung 276
4.3.3.2 Beispiel für eine gruppierte Vorlageninstanziierung....277
4.3.3.3 Beispiel für eine Diagrammerstellung 277
4.3.4 Aufbau der Steuerdatei "procedure.cfg" für Import aus ZIP-Datei 278

4.4 Massenbearbeitung....280
4.4.1 Daten exportieren....280
4.4.2 Inhalt der Exportdatei....281
4.4.3 Daten importieren....282

5 Diagnose & Visualisierung 285

5.1 Trend and Messaging Editor 285
5.1.1 Funktionen des Trend and Messaging Editors 285
5.1.2 Meldesystem 285
5.1.2.1 Meldungsklassen 285
5.1.2.2 Meldungseditor 286
5.1.2.3 Meldungen bearbeiten 286
5.1.2.4 Message – die Meldekomponente 287
5.1.2.5 Komponentenspezifische Meldungen 288
5.1.2.6 Meldungen in der Statusleiste 288
5.1.2.7 Begrenzungen des Meldesystems 288
5.1.3 Archiv 288
5.1.4 Kurvenbilder 290
5.1.4.1 Signaldarstellung mit Kurvenbildern 290
5.1.4.2 Kurvenbilder anlegen und konfigurieren 290
5.1.4.3 Kurvenverläufe anzeigen 293
5.1.4.4 Kurvenbildeditor 298

5.2 Suchen & Ersetzen 299
5.2.1 Suchen 300
5.2.1.1 Suchen mit dem Editor Suchen & Ersetzen 300
5.2.1.2 Suchen über die Eigenschaften von Signalen und Konnektoren 302
5.2.1.3 Suchen von fixierten Signalen 302
5.2.2 Ersetzen 303
5.2.2.1 Ersetzen mit dem Editor Suchen & Ersetzen 303
5.2.2.2 Aktualisieren 304

5.3 Konsistenzprüfung 304

5.4 Analysefunktionen 306
5.4.1 Grundlagen zur Analysefunktion 306
5.4.2 Auswertung generieren 308

6 Skripte 309

6.1 Funktionsweise von Skripten 309

6.2 Handhabung von Skripten 310
6.2.1 Erstellen eines Skripts 310
6.2.2 Ausführen eines Skriptes 311
6.2.3 Externe Skripte 312

6.3 Skriptsyntax 313
6.3.1 Steuern des Skripts 313
6.3.2 Skripte zusammensetzen 314
6.3.3 Skripte kommentieren 315
6.3.4 Signale in Skripten 315
6.3.5 Steuern des Ablaufs der Simulation 316
6.3.5.1 Initialisieren der Simulation 317
6.3.5.2 Starten der Simulation 317
6.3.5.3 Starten und Warten auf einen absoluten Zeitpunkt 317
6.3.5.4 Starten und Warten auf einen relativen Zeitpunkt 318
6.3.5.5 Starten und eine bestimmte Anzahl von Zyklen warten 318
6.3.5.6 Starten und Warten auf ein Ereignis 318
6.3.5.7 Anhalten der Simulation 318

6.3.5.8 Ausführen eines Einzelschritts 318
6.3.5.9 Speichern eines Schnappschusses 319
6.3.5.10 Laden eines Schnappschusses 319
6.3.5.11 Rücksetzen der Simulationszeit 320
6.3.6 Protokollierung 320
6.3.6.1 Protokolldatei öffnen und schließen 320
6.3.6.2 Unformatierte Ausgabe 321
6.3.6.3 Formatierte Ausgabe 321
6.3.6.4 Ausgabe von Uhrzeit und Datum 322
6.3.6.5 Ausgabe von Versionsinformation 322
6.3.6.6 Die Systemfunktion _printlog 322
6.3.7 Signalverläufe 323
6.3.7.1 Plot-Datei öffnen und schließen 323
6.3.7.2 Signale festlegen 324
6.3.7.3 Zyklus festlegen 324
6.3.8 Setzen von Signalen 325
6.3.8.1 Einzelwerte setzen 325
6.3.8.2 Verbundene Signale trennen 326
6.3.8.3 Vorgeben eines Signalverlaufs 326
6.3.9 Bedingte Abarbeitung 327
6.3.10 Zugriff auf die Simulationszeit 329

7 Komponententypeditor 331

7.1 Die Benutzeroberfläche 331
7.1.1 Starten des CTE 331
7.1.2 Gliederung der Benutzeroberfläche 332
7.1.3 Die Menü- und Symbolleiste 334
7.1.4 Der Projektbaum 334
7.1.5 Tastaturkürzel 335
7.1.6 Die Task-Cards 336
7.1.6.1 Die Task-Card "Signale" 336
7.1.6.2 Die Task-Card "Controls" 337
7.1.6.3 Die Task-Card "Komponenten" 338
7.1.6.4 Die Task-Card "Aufzählungstypen" 339
7.1.6.5 Die Task-Card "Verbindungstypen" 340

7.2 Grundlagen der Komponententypen 342
7.2.1 Das Typ-Instanz-Konzept von SIMIT 342
7.2.2 Eigenschaften von Komponententypen 343
7.2.3 Die allgemeinen Eigenschaften der Komponententypen 344
7.2.3.1 Übersicht 344
7.2.3.2 Die Verwaltungseigenschaften 345
7.2.3.3 Schutz des Komponententyps 345
7.2.3.4 Besondere Eigenschaften 346
7.2.3.5 Änderungsvermerke 346

7.3 Die Anschlüsse und Parameter von Komponententypen 347
7.3.1 Der Anschlusseditor 347
7.3.2 Spezielle Vorbelegung implizit verschaltbarer Eingänge 350
7.3.3 Komplexe Verbindungstypen 352
7.3.4 Die Parameter von Komponententypen 354

7.4 Das Verhalten von Komponententypen 355
7.4.1 Einleitung 355

7.4.2 Die Zustandsgrößen ... 356
7.4.3 Initialisierung, zyklische Berechnung und Funktionen ... 358
7.4.4 Topologie ... 359

7.5 Die Visualisierung von Komponententypen ... 360
7.5.1 Einleitung ... 360
7.5.2 Das Grundsymbol ... 361
7.5.2.1 Editieren des Grundsymbols ... 361
7.5.2.2 Editieren von Grafiken ... 361
7.5.2.3 Editieren der Anschlüsse ... 362
7.5.2.4 Editieren der Eigenschaften ... 362
7.5.2.5 Maßstäblichkeit ... 365
7.5.3 Das Verknüpfungssymbol ... 365
7.5.4 Das Bedienfenster ... 366

7.6 Syntax der Verhaltensbeschreibung ... 369
7.6.1 Übersicht ... 369
7.6.2 Umsetzung der Verhaltensbeschreibung auf C#-Code ... 369
7.6.3 Der gleichungsorientierte Ansatz ... 370
7.6.3.1 Übersicht ... 370
7.6.3.2 Lokale Variablen ... 370
7.6.3.3 Konstanten ... 371
7.6.3.4 Die Berechnungsreihenfolge ... 371
7.6.3.5 Operatoren ... 373
7.6.3.6 Bedingte Zuweisungen ... 373
7.6.3.7 Aufzählungstypen ... 374
7.6.3.8 Vektoren ... 375
7.6.3.9 Funktionsaufrufe mathematischer Standardfunktionen ... 376
7.6.3.10 Selbstdefinierte Funktionen ... 377
7.6.3.11 Differentialgleichungen ... 378
7.6.3.12 Zugriff auf diskrete Zustandsgrößen ... 379
7.6.4 Der anweisungsorientierte Ansatz ... 380
7.6.4.1 Einleitung ... 380
7.6.4.2 Funktionen ... 380
7.6.4.3 Blöcke ... 381
7.6.4.4 Lokale Variablen ... 381
7.6.4.5 Felder ... 382
7.6.4.6 Konstanten ... 382
7.6.4.7 Schleifen ... 382
7.6.4.8 Verzweigungen ... 383
7.6.4.9 Systemfunktionen ... 384
7.6.4.10 Operatoren ... 386
7.6.4.11 Zugriff auf Zustandsgrößen ... 388
7.6.5 Interne Variablen und Konstanten ... 388
7.6.6 Der Parametertyp characteristic ... 389

8 Bibliotheken ... 391

8.1 Basisbibliothek ... 391
8.1.1 Allgemeines ... 391
8.1.1.1 Einführung ... 391
8.1.1.2 Symbole der Komponenten ... 391
8.1.1.3 Symbole der Controls ... 393
8.1.1.4 Aufbau des Simulationsmodells ... 394

8.1.1.5 Anschlüsse von Komponenten 394
8.1.1.6 Anschlüsse von Controls 395
8.1.1.7 Verbinden von Anschlüssen 395
8.1.1.8 Setzen von Eingängen 398
8.1.1.9 Eigenschaften von Komponenten 399
8.1.1.10 Fehlermeldungen von Komponenten 406
8.1.1.11 Eigenschaften von Controls 407
8.1.2 Konnektoren 408
8.1.2.1 Globaler Konnektor 409
8.1.2.2 Peripherie-Konnektoren 410
8.1.2.3 Topologischer Konnektor 411
8.1.2.4 Unit-Konnektor 411
8.1.3 Standardkomponenten 412
8.1.3.1 Unterscheidung der Standardkomponenten 412
8.1.3.2 Analoge Funktionen 412
8.1.3.3 Ganzzahlige Funktionen 435
8.1.3.4 Mathematische Funktionen 441
8.1.3.5 Binäre Funktionen 446
8.1.3.6 Konvertierung von Werten 456
8.1.3.7 Allgemeine Komponenten im Verzeichnis Misc 465
8.1.4 Antriebskomponenten 474
8.1.4.1 Antriebe als Komponententypen 474
8.1.4.2 Ventilantriebe 475
8.1.4.3 Pumpenantriebe und Lüfterantriebe 478
8.1.4.4 Motor 480
8.1.4.5 PROFIdrive-Antriebe 486
8.1.4.6 Motorsteuergeräte SIMOCODE pro 527
8.1.5 Sensorkomponenten 545
8.1.5.1 SIWAREXU-Komponenten 545
8.1.5.2 Zählerbaugruppe FM350-1 552
8.1.6 Kommunikationskomponenten 559
8.1.6.1 Komponenten für SIMATIC 559
8.1.6.2 Komponenten für SINUMERIK 573
8.1.7 Controls 577
8.1.7.1 Controls zur Anzeige von Signalwerten 577
8.1.7.2 Controls zur Eingabe von Signalwerten 583
8.1.7.3 Sonstige Controls 593
8.1.7.4 3D Viewer-Control 596

8.2 Bibliotheken CHEM-BASIC und FLOWNET 610
8.2.1 Einleitung 610
8.2.2 Topologischer Konnektor 611
8.2.3 Geodätische Höhe 612
8.2.4 Flussnetze 614
8.2.4.1 Grundlagen der Flussnetze 615
8.2.4.2 In Flussnetzen verwendete Größen 618
8.2.4.3 Modellierung der Zweige von Flussnetzen 618
8.2.4.4 Modellierung der Knoten von Flussnetzen 619
8.2.4.5 Wärmeaustausch mit der Umgebung 622
8.2.4.6 Parametrierung von Flussnetzen 623
8.2.4.7 Kopplung Simulationsmodell mit Aktor-Sensor-Ebene 627
8.2.5 Komponenten der Bibliothek CHEM-BASIC 630
8.2.5.1 Burner 630

8.2.5.2 Fittings ... 631
8.2.5.3 Graphics ... 638
8.2.5.4 Heatexchanger ... 639
8.2.5.5 Measurements ... 653
8.2.5.6 Mixing Apparatuses ... 664
8.2.5.7 Pump ... 667
8.2.5.8 Separators ... 672
8.2.5.9 System ... 703
8.2.5.10 Tanks ... 710
8.2.5.11 Valves ... 732
8.2.6 Komponenten der Bibliothek FLOWNET ... 742
8.2.6.1 Allgemeine Komponenten ... 742
8.2.6.2 Messkomponenten ... 754
8.2.6.3 Komponententypen für Medium "Wasser/Dampf" ... 760
8.2.6.4 Komponententypen für Flüssigkeiten als Medium ... 777
8.2.6.5 Komponententypen für Gase als Medium ... 790
8.2.7 Komponententypen für Flussnetze selbst erstellen ... 803
8.2.7.1 Topologische Eigenschaften ... 803
8.2.7.2 Anbindung an das Lösungsverfahren ... 807
8.2.7.3 Konstanten und Funktionen ... 814
8.2.7.4 Initialisierung von Flussnetzsimulationen ... 819

8.3 Bibliothek CONTEC ... 819
8.3.1 Einleitung ... 819
8.3.2 Fördertechniksimulation ... 820
8.3.2.1 Grundlagen der Fördertechniksimulation ... 822
8.3.2.2 Modellierung der Fördergüter ... 823
8.3.2.3 Modellierung des Streckennetzes ... 825
8.3.2.4 Besondere Aspekte der Fördertechniksimulation ... 829
8.3.2.5 Maßstäblichkeit ... 830
8.3.2.6 Generierung der Simulation von Antrieben und Sensoren ... 833
8.3.3 Komponenten der Bibliothek CONTEC ... 838
8.3.3.1 Topologischer Konnektor in der CONTEC-Bibliothek ... 838
8.3.3.2 Komponententypen für Fördersysteme mit Fahrzeugen ... 840
8.3.3.3 Komponententypen für fahrzeuglose Fördersysteme ... 855
8.3.3.4 Komponententypen zur Simulation von Fördergütern ... 889
8.3.3.5 Komponenten zur Simulation von Identifikationssystemen ... 896
8.3.4 Komponententypen für Fördertechniksimulation selbst erstellen ... 925
8.3.4.1 Topologische Eigenschaften ... 926
8.3.4.2 Anbindung an das Lösungsverfahren ... 929
8.3.4.3 Systemfunktionen ... 938
8.3.4.4 Systemvariablen ... 947

9 Menüs und Dialogfelder ... 949

9.1 Menüs ... 949
9.1.1 Portalansicht > Start ... 949
9.1.2 Portalansicht > Kopplungen ... 950
9.1.3 Portalansicht > Simulationsmodell ... 950
9.1.4 Portalansicht > Automatische Modellerstellung ... 950
9.1.5 Portalansicht > Diagnose & Visualisierung ... 951
9.1.6 Projekt > Neues Projekt... ... 952
9.1.7 Projekt > Öffnen... ... 953
9.1.8 Projekt > Schließen ... 953

9.1.9 Projekt > Alles speichern....954
9.1.10 Projekt > Speichern unter....954
9.1.11 Projekt > Archivieren....954
9.1.12 Projekt > Dearchivieren....955
9.1.13 Projekt > Analyse....956
9.1.14 Projekt > Beenden....956
9.1.15 Bearbeiten > Ausschneiden....956
9.1.16 Bearbeiten > Kopieren....956
9.1.17 Bearbeiten > Einfügen....956
9.1.18 Simulation > Initialisieren....956
9.1.19 Simulation > Starten....956
9.1.20 Simulation > Anhalten....957
9.1.21 Simulation > Einzelschnitt....957
9.1.22 Simulation > Beenden....957
9.1.23 Simulation > Schnappschuss....957
9.1.24 Fenster > Horizontal teilen....957
9.1.25 Fenster > Vertikal teilen....957
9.1.26 Fenster > Teilung aufheben....957
9.1.27 Fenster > Alle schließen....958
9.1.28 Automatische Modellerstellung > Vorlagen instanziieren....958
9.1.29 Automatische Modellerstellung > Automatisierter Import....958
9.1.30 Automatische Modellerstellung > Massenbearbeitung Import....958
9.1.31 Extras > Zoom....958
9.1.32 Extras > SU-Verwaltung....958
9.1.33 Extras > Kopplungssignale zuweisen....959
9.1.34 Hilfe > Hilfe anzeigen....959
9.1.35 Hilfe > Info....959
9.1.36 CTE > Komponente > Neue Komponente....959
9.1.37 CTE > Komponente > Öffnen....959
9.1.38 CTE > Komponente > Schließen....959
9.1.39 CTE > Komponente > Speichern....960
9.1.40 CTE > Komponente > Speichern unter....960
9.1.41 CTE > Komponente > Beenden....960

9.2 Dialogfelder....961
9.2.1 Dialogfeld "SU Import"....961
9.2.2 Dialogfeld "PLCSIM Advanced-Import"....962
9.2.3 Dialogfeld "Virtual Controller Import"....964
9.2.4 Dialogfeld "AWL-Import"....966
9.2.5 Dialogfeld "Import von Signaleigenschaften"....967
9.2.6 Dialogfeld "Export von Signaleigenschaften"....970
9.2.7 Dialogfeld "Projektauswahl"....972
9.2.8 Dialogfeld "SU-Verwaltung"....973
9.2.9 Dialogfeld "Instanziiere Vorlagen"....974
9.2.10 Dialogfeld "Massenbearbeitung Import"....976
9.2.11 Editor "Kennlinie"....976
9.2.12 Dialogfeld "Suchen & Ersetzen"....977
9.2.13 Dialogfeld "Info"....978
9.2.14 Dialogfeld "Drucken"....978
9.2.15 Dialogfeld "Vorlage instanziieren"....979
9.2.16 Dialogfeld "Auswahl"....980
9.2.17 Dialogfeld "Aufzählung" (CTE)....981
9.2.18 Dialogfeld "Komponente verwalten" (CTE)....982

9.2.19 Dialogfeld "Verbindungstyp" (CTE) 983

Grundlagen von SIMIT

1

1.1 Funktionsweise

SIMIT hat folgende Verwendungsmöglichkeiten:

- Komplette Anlagensimulation
  Simulation von Signalen, Geräten und Anlagenverhalten
- Ein- und Ausgabesimulator von Testsignalen für eine Automatisierungssteuerung
- Test und Inbetriebnahme von Automatisierungssoftware

Auch wenn zunächst z. B. nur die Bedienoberfläche für den Signaltest genutzt wird, können später jederzeit Simulationsmodelle hinzugefügt werden, um das Anlagenverhalten zu simulieren und dynamische Tests durchzuführen.

Prinzipiell arbeiten Sie mit den folgenden Bestandteilen, um eine Simulation zu erstellen:

- Diagramm
  Zum Aufbau einer Simulation setzen Sie die in den Bibliotheken vorhandenen Komponenten auf dem Diagrammeditor zusammen und tragen passende Parameter ein.
  Weitere Informationen hierzu finden Sie im Kapitel: Erstellen und Bearbeiten von Diagrammen (Seite 225).
- Visualisierung
  Visualisierungen geben einen Überblick über die Signale Ihrer Anlage. Signale werden mit Controls (Eingabe- und Anzeigeobjekte) und grafischen Objekten visualisiert.
  Weitere Informationen hierzu finden Sie im Kapitel: Visualisieren einer Simulation (Seite 31).
- Kopplung
  Die Kopplung ist die Schnittstelle zum Automatisierungssystem und wird zum Signalaustausch benötigt. Den Signalumfang, den SIMIT verarbeiten soll, können Sie individuell festlegen.
  Weitere Informationen zu Kopplungen finden Sie im Kapitel: Kopplungen (Seite 59).

1.2 DEMO-Modus

Mit dem DEMO-Modus können Sie sich einen Eindruck von der Handhabung und Leistungsfähigkeit von SIMIT verschaffen.

Starten von SIMIT im DEMO-Modus

Starten Sie SIMIT DEMO durch Doppelklick auf die Bildschirm-Verknüpfung SIMIT SP Demo. Ein Dongle ist nicht notwendig.

Im DEMO-Modus sind nur die folgenden Funktionsmodule von SIMIT verfügbar:

- Makrokomponenteneditor
  Weitere Informationen hierzu finden Sie im Kapitel: Makrokomponenteneditor (Seite 243).
- Diagrammeditor
  Weitere Informationen hierzu finden Sie im Kapitel: Erstellen und Bearbeiten von Diagrammen (Seite 225).
- Trend and Messaging Editor
  Weitere Informationen hierzu finden Sie im Kapitel: Trend and Messaging Editor (Seite 285).
- Automatische Modellerstellung
  Weitere Informationen hierzu finden Sie im Kapitel: Automatische Modellerstellung (Seite 251).

SIMIT hat im DEMO-Modus in den folgenden Punkten nur eingeschränkte Funktionalität:

- **Virtual Controller**
  Sie können SIMIT im DEMO-Modus laufen lassen und den Virtual Controller lizenziert, sowie umgekehrt. Die Einschränkungen des DEMO-Modus gelten dann aber sowohl für SIMIT als auch für den Virtual Controller.
- **Speichern und Archivieren**
  Projekte, Vorlagen und Makrokomponenten können im DEMO-Modus gespeichert werden. Im DEMO-Modus erstellte Projekte, Vorlagen und Makrokomponenten können aber nur auf dem Rechner verwendet werden, auf dem diese erstellt wurden.

  Hinweis

  Im DEMO-Modus erstellte Projekte, Vorlagen und Makrokomponenten sind nicht kompatibel mit der Vollversion von SIMIT.

  Im DEMO-Modus erstellte Projekte sind nicht upgradefähig.

  Im DEMO-Modus können Sie Projekte nicht archivieren.

- **Öffnen und Dearchivieren**
  Sie können im DEMO-Modus unter folgenden Voraussetzungen ein Projekt öffnen:
  - Projekt wurde auf diesem Rechner im DEMO-Modus erstellt.
  - Version des Projekts entspricht der installierten Software-Version.

  Projekte, die mit einer Vollversion erstellt wurden, können nicht geöffnet werden.
  Sie können Projekte dearchivieren, die in einer Vollversion archiviert wurden. Wenn das dearchivierte Projekt im DEMO-Modus verändert wurde, kann es aber nicht in der Vollversion verwendet werden.
- **Adressbereich**
  Im DEMO-Modus ist der Adressbereich für das Verschalten von Signalen auf 30 Byte beschränkt. Wenn Sie eine OPC-Kopplung verwenden, können Sie bis zu 30 Signale verschalten.
- **Laufzeit**
  Sie können SIMIT beliebig lange im DEMO-Modus benutzen, allerdings ist die Laufzeit einer Simulation auf 45 Minuten begrenzt. Nach Ablauf dieser 45 Minuten endet die Simulation automatisch. Sie können die Simulation nach dem Beenden wieder starten.

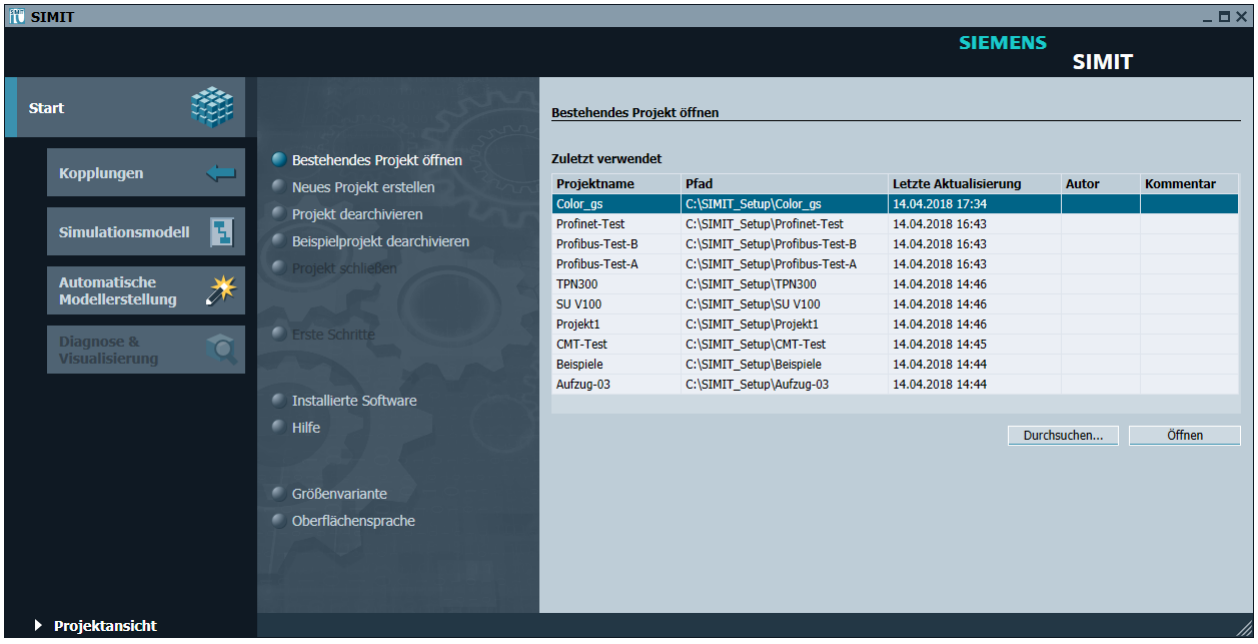
- **Anzahl der Kopplungen**
  Sie können im DEMO-Modus maximal eine Kopplung in einem SIMIT-Projekt angelegen. Die Programmierschnittstelle "Externe Kopplungen" wird nicht unterstützt.
- **Projektordner**
  Sie können im DEMO-Modus Projekte nur in einem vorgegebenen Speicherort im Arbeitsbereich von SIMIT ablegen.
- **Bibliotheken für Makrokomponenten und Vorlagen**
  Sie können Makrokomponenten und Vorlagen im DEMO-Modus nur innerhalb des Arbeitsbereiches von SIMIT ablegen. Weitere Bibliotheksordner können Sie nicht öffnen.
- **Schnappschüsse**
  Im DEMO-Modus ist keine Aufnahme von Schnappschüssen möglich.
- **Simulationsablauf**
  Im DEMO-Modus kann die Simulation nur in Echtzeit ablaufen. Schneller oder langsamer ist nicht möglich.
- **Anzahl Virtual Controller**
  Sie können maximal 32 VCs pro SIMIT-System laufen lassen.
- **Größenvariante**
  Sie können im DEMO-Modus nur die Größenvariante S nutzen.

1.3 Starten von SIMIT

Starten Sie SIMIT entweder über das Windows-Startmenü "**Alle Programme > Siemens Automation > SIMIT > SIMIT Simulation Platform**" oder über die Verknüpfung auf dem Bildschirm.

Während des Startens von SIMIT wählen Sie die Größenvariante (XS, S, M, L, XL). Sie können die Größenvariante auch nachträglich ändern über Portalansicht > Start > Größenvariante.

Nach dem Startvorgang öffnet sich die Portalansicht:



Die Portalansicht erleichtert den Einstieg in SIMIT mit einer übersichtlichen Darstellung. Die wichtigsten Grundfunktionen können direkt angewählt werden:

- Projekte verwalten
- Kopplungen anlegen
- Simulationsmodelle erstellen
- Automatischer Datenimport aus unterschiedlichen Dateiformaten

Weitere Funktionen sind:

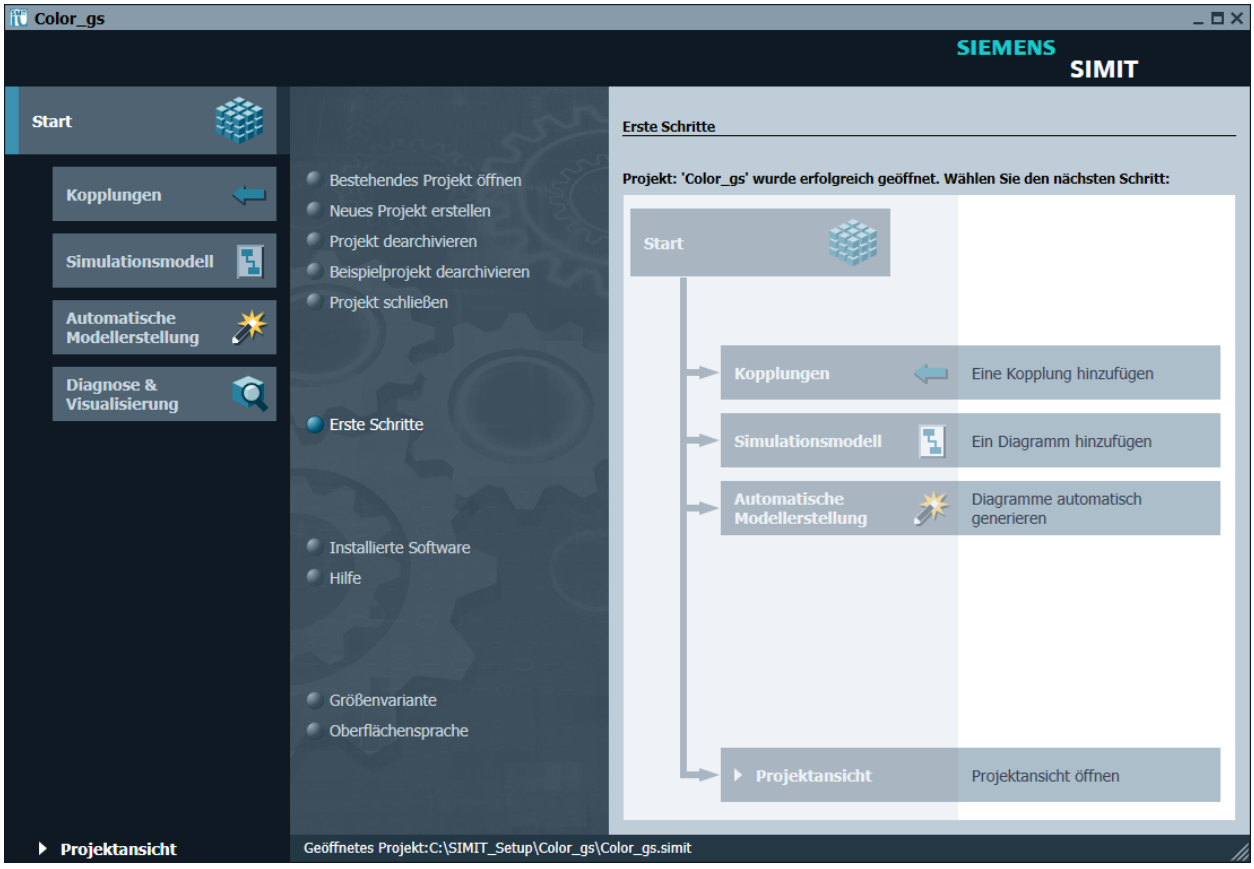
- Durchführen von Konsistenzprüfungen
- Aufrufen der Funktion "Suchen & Ersetzen"
- Erstellen eines neuen Kurvenbildes
- Bearbeiten von Archiven

Über die Schaltfläche "Projektansicht" wechseln Sie zur Projektansicht. Der volle Funktionsumfang von SIMIT steht nur in der Projektansicht zur Verfügung.

Erste Schritte

Nach dem Öffnen, Anlegen oder Dearchivieren eines Projekts öffnet sich der Bereich "Erste Schritte". Hier können Sie die grundlegenden Objekte hinzufügen, die zur Funktion eines SIMIT-Projekts nötig sind.

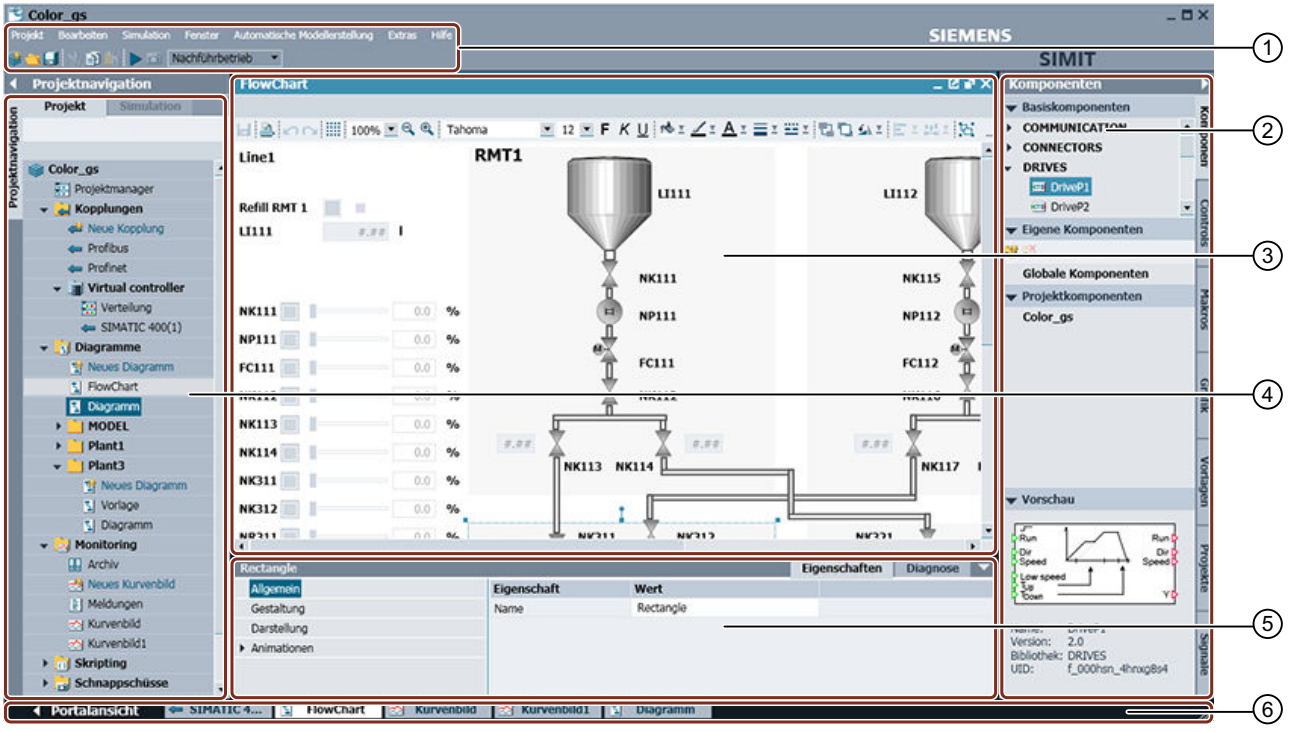
- Kopplung hinzufügen
- Diagramm hinzufügen
- Diagramme automatisch generieren



Sie können diese Schritte hier durchführen oder zur **Projektansicht** wechseln, um das Projekt weiter zu bearbeiten und die Simulation zu starten.

1.4 Die Oberfläche von SIMIT

Die Oberfläche von SIMIT gliedert sich in die folgenden Bereiche:



①	Menüleiste und Symbolleiste	Zugriff auf die Funktionen von SIMIT. Weitere Funktionen stehen in Kontextmenüs in der Projektnavigation zur Verfügung.
②	Task-Cards	Hier werden Objekte wie Bibliothekskomponenten, Controls und Grafikobjekte aufgelistet, die im aktuell geöffneten Editor verwendet werden können. Diese Objekte sind in Task-Cards sortiert.
③	Arbeitsbereich	Hier werden die Editoren zum Bearbeiten geöffnet.
④	Projektnavigation	Hier wird das aktuelle Projekt in einer Baumansicht angezeigt.
⑤	Eigenschafts-fenster	Hier werden die Eigenschaften des ausgewählten Objekts angezeigt.
⑥	Editorleiste und Statuszeile	Hier können Sie zwischen geöffneten Editoren und in die Portalansicht umschalten. Wenn erforderlich, werden hier Informationen zum aktuellen Status von SIMIT angezeigt.

Alle Editoren werden im Arbeitsbereich geöffnet. Task-Cards werden für jeden Editor spezifisch zur Verfügung gestellt. Der Arbeitsbereich lässt sich teilen, um zwei Editoren nebeneinander oder untereinander im Arbeitsbereich öffnen zu können.

Die Fensterschaltflächen eines Editors:



Über die Fensterschaltflächen eines Editors lassen sich folgende Funktionen durchführen:

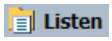
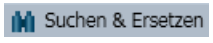
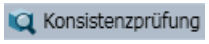
- Minimieren, d. h. auf den Eintrag in der Editorleiste reduzieren
- Auf einen Teil des Arbeitsbereichs reduzieren

- Auf den gesamten Arbeitsbereich vergrößern
- Schließen
- Einen Editor mit Task-Cards und Eigenschaftsfenster aus dem Arbeitsbereich als eigenes Fenster herauslösen
- Ein Editorfenster, das aus dem Arbeitsbereich herausgelöst wurde, wieder in den Arbeitsbereich einfügen
- Ein herausgelöstes Editorfenster im Vordergrund fixieren

Die Projektnavigation

Projekte werden in der Projektnavigation verwaltet. Ein SIMIT-Projekt gliedert sich hier in die folgenden Baumeinträge, wobei ein Projekt nicht alle hier aufgeführten Objekte enthalten muss:

Color_gs	aktueller Projektname
Projektmanager	Baumeintrag zum Projektmanager. Im Projektmanager können Elemente aus anderen Projekten in das aktuell geöffnete Projekt kopiert werden. Weitere Informationen hierzu finden Sie im Abschnitt: Ansicht und Funktionen des Projektmanagers (Seite 221).
Kopplungen	Ordner "Kopplungen": Ablageordner für die Kopplungen des aktuellen Projekts. Weitere Informationen hierzu finden Sie im Kapitel: Kopplungen (Seite 59).
	Symbol für Kopplungen. Kopplungen stellen die Verbindung zwischen SIMIT und einer Steuerung oder anderen Applikationen her.
Neue Kopplung	Baumeintrag zum Anlegen einer neuen Kopplung.
Diagramme	Ordner "Diagramme": Ablageordner für die Diagramme.
	Symbol für Diagramme. Diagramme enthalten ein mit Hilfe von Bibliothekskomponenten und Controls erstelltes Simulationsmodell.
Neues Diagramm	Baumeintrag zum Erstellen eines neuen Diagramms. Weitere Informationen hierzu finden Sie im Abschnitt: Aufbau einer Simulation (Seite 28).
Monitoring	Ordner "Monitoring": Dieser Ordner enthält folgende Funktionen und ist auch deren Ablageort: • Neues Kurvenbild • Meldungen • Archiv Weitere Informationen hierzu finden Sie im Abschnitt: Funktionen des Trend and Messaging Editors (Seite 285).
Skripting	Ordner "Skripting": Dieser Ordner enthält die Funktion "Neues Skript" und ist der Ablageort für vorhandene Skripte. Weitere Informationen hierzu finden Sie im Abschnitt: Erstellen eines Skripts (Seite 310).

	Ordner "Listen": Dieser Ordner enthält die Funktion "Neue Liste" und ist der Ablageort für vorhandene Materiallisten. Dieser Ordner ist nur relevant, wenn die Bibliothek "CONTEC" installiert ist. Weitere Informationen hierzu finden Sie im Abschnitt: Modellierung der Fördergüter (Seite 823).
	Ordner "Schnappschüsse": Ablageordner für die aufgezeichneten Momentaufnahmen der Simulation. Ist die Simulation aktiv, enthält der Ordner zusätzlich die Funktion "Neuer Schnappschuss". Weitere Informationen hierzu finden Sie im Abschnitt: Aktionen bei laufender Simulation (Seite 39).
	Baumeintrag zur Funktion "Suchen & Ersetzen". Gesucht und ersetzt werden können Signale, Komponenten, Konnektoren und Grafiktexte. Weitere Informationen hierzu finden Sie im Kapitel: Suchen & Ersetzen (Seite 299).
	Baumeintrag zum Durchführen der Konsistenzprüfung. Die Konsistenzprüfung prüft das Projekt auf formale Fehler. Weitere Informationen hierzu finden Sie im Abschnitt: Konsistenzprüfung (Seite 304).
	Baumeintrag zum Starten der Simulation.

Diese Funktionen werden jeweils durch einen Doppelklick ausgeführt.

Alternativ stehen die Funktionen für die einzelnen Projektelemente auch im jeweiligen Kontextmenü des Baumeintrags zur Verfügung.

1.5 Erstellen einer Simulation

1.5.1 Aufbau einer Simulation

Eine Simulation wird mit Komponenten, Controls und Verbindungen auf Diagrammen zusammengestellt und mit Konnektoren mit der Kopplung verbunden.

Im Folgenden finden Sie eine Kurzbeschreibung dieser Bestandteile. Für weitergehende Informationen beachten Sie die Links.

- **Diagramm**
  Das Diagramm beinhaltet das Simulationsmodell. Damit werden Komponenten, Controls und Verbindungen grafisch dargestellt. Diagramme werden im Diagrammeditor erstellt und bearbeitet. Ein Simulationsmodell kann aus mehreren Diagrammen bestehen.
  Weitere Informationen hierzu finden Sie im Kapitel: Diagrammeditor (Seite 225).
  Die folgende Abbildung zeigt ein Beispiel für ein Diagramm mit Komponenten, Verbindungen und Controls:

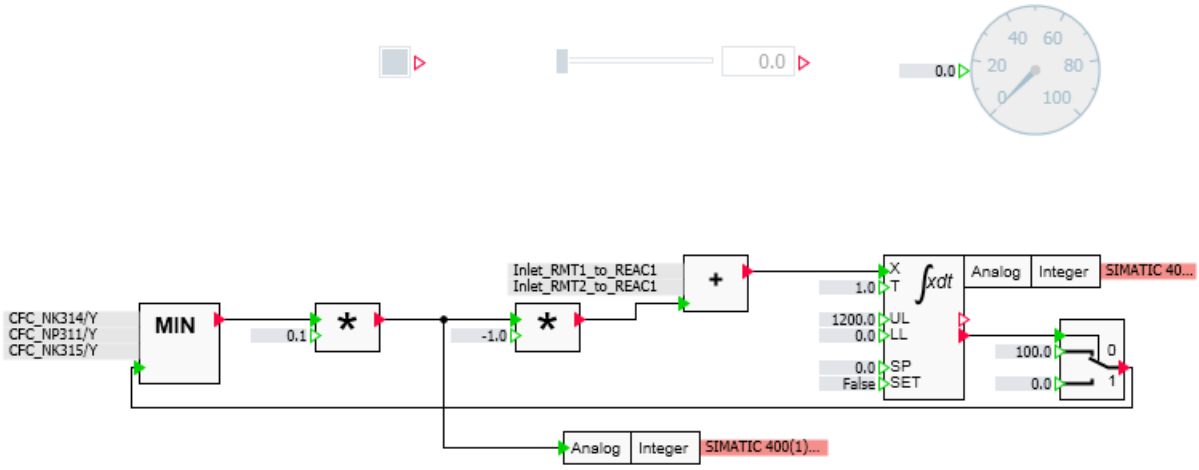


Diagramme bestehen prinzipiell aus den folgenden Objekten:

- **Komponenten**
  Die entsprechenden Komponenten für logische und arithmetische Funktionen, für Antriebe, Sensoren, Verbindungen und Kommunikation finden Sie in der Task-Card "Komponenten" im Bereich "Basiskomponenten".
  Um Komponenten in ein Diagramm einzufügen, markieren Sie diese und ziehen Sie sie auf das Diagramm.
  Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Komponenten" (Seite 232)
- **Controls**
  Die Controls zur Eingabe und zur Anzeige von Werten finden Sie in der Task-Card "Controls". Im Bereich "Anzeige" finden Sie Objekte zum dynamischen Anzeigen von Werten aus einer laufenden Simulation. Im Bereich "Eingabe" finden Sie Objekte zur Vorgabe von bestimmten Werten in eine laufende Simulation.
  Um Controls in ein Diagramm einzufügen, markieren Sie dieses und ziehen Sie sie auf das Diagramm.
  Weitere Informationen hierzu finden Sie im Kapitel: Controls (Seite 577)
- **Verbindungen**
  Verbindungen werden über Signale definiert und im Diagramm entweder über direkte Verbindungen oder als Konnektoren dargestellt.
  Signale verbinden Komponenten und Controls mit der Kopplung und die Kopplung mit dem Automatisierungssystem. Die Signale der Steuerung werden im Kopplungseditor bearbeitet.
  - **Konnektoren**
    Auf einem Diagramm werden Signale durch Konnektoren dargestellt: Ausgangssignale durch grüne Ausgangskonnektoren (*Output*), Eingangssignale durch rote Eingangskonnektoren (*Input*). Ziehen Sie Konnektoren per Drag & Drop aus der Kopplung auf Diagramme. Teilen Sie den Arbeitsbereich über den Menüpunkt "Fenster > Horizontal teilen" und öffnen Sie die Kopplung und das Diagramm. Ziehen Sie dann das gewünschte Signal aus der Kopplung auf das Diagramm, indem Sie es im Kopplungsfenster am linken Rand anfassen und <Shift> gedrückt halten. Verbinden Sie den Anschluss des Konnektors mit dem Anschluss einer Komponente.
    Kopplungssignale können auch aus der Task-Card "Signale" auf ein Diagramm gezogen werden. Filtern Sie dazu die Signale entsprechend, halten Sie <Shift> gedrückt und ziehen Sie die gewünschten Signale auf das Diagramm.

Um ein neues Diagramm zu erstellen, ziehen Sie die Komponenten und Controls aus den entsprechenden Task-Cards auf das Diagramm, verbinden Sie deren Anschlüsse und geben Sie die Parameterwerte ein. Weitere Informationen finden Sie im Abschnitt: Erstellen und Bearbeiten von Diagrammen (Seite 225).

- **Kopplung**
  Über Kopplungen kommuniziert SIMIT mit dem Automatisierungssystem.
  Kopplungssignale können in verschiedenen Formaten importiert und können im Kopplungseditor editiert werden.
  Eine Kopplung wird in der Projektnavigation erstellt, indem Sie auf den Baumeintrag "Neue Kopplung" doppelklicken. Wählen Sie den gewünschten Kopplungstyp aus und öffnen Sie mit einem weiteren Doppelklick auf die neue Kopplung den Kopplungseditor. Hier können Sie Signale importieren und editieren.
  Weitere Informationen hierzu finden Sie im Kapitel: Kopplungen (Seite 59).

Hinweis

Die Konfiguration der Kopplung muss gespeichert sein, um ein Signal herausziehen zu können.

1.5.2 Visualisieren einer Simulation

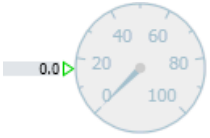
Signalwerte und Signalzustände einer Simulation werden mit Grafiken und Controls grafisch dargestellt.

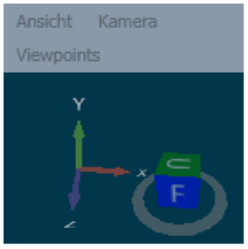
Controls sind vorgefertigte Objekte zur Eingabe oder Anzeige von Signalen. Sie müssen nur noch mit dem gewünschten Signal verbunden werden.

Grafiken werden individuell erstellt. Es gibt statische und animierte Grafiken.

Controls

Controls unterscheiden sich nach Eingabe oder Anzeige eines Signals und nach Datentyp. Folgende Controls stehen zur Verfügung:

Name und Symbol	Datentyp	Verwendung	Link zu weiterführenden Informationen
Binäranzeige 	Binär	Signalanzeige	Binäranzeige (Seite 577)
Analoganzeige 	Analog, Integer	Signalanzeige	Analoganzeige (Seite 578)
Digitalanzeige 	Analog, Integer	Signalanzeige	Digitalanzeige (Seite 580)
Balkenanzeige 	Analog, Integer	Signalanzeige	Balkenanzeige (Seite 582)
Taster 	Binär	Signaleingabe	Taster (Seite 583)
Taster mit Bild 	Binär	Signaleingabe	Taster mit Bild (Seite 584)
Schalter 	Binär	Signaleingabe	Schalter (Seite 585)

Name und Symbol	Datentyp	Verwendung	Link zu weiterführenden Informationen
Schalter mit Bild 	Binär	Signaleingabe	Schalter mit Bild (Seite 586)
Stufenschalter 	Integer	Signaleingabe	Stufenschalter (Seite 587)
Stufenschalter mit Bild 	Integer	Signaleingabe	Stufenschalter mit Bild (Seite 588)
Digitaleingabe 	Analog, Integer	Signaleingabe	Digitaleingabe (Seite 590)
Schieber 	Analog	Signaleingabe	Schieber (Seite 592)
3D-Viewer 	–	Einbinden von dreidimensionalen Grafiken in ein Diagramm	3D Viewer-Control (Seite 596)
Signaltrenner	–	Fixieren von Signalen, muss mit anderen Controls verbunden werden	Signaltrenner (Seite 593)
Aktion	–	Öffnen von Diagrammen und Kurvenbildern.	Aktion (Seite 595)

Um ein Control in das aktuelle Diagramm einzubinden, gehen Sie folgendermaßen vor:

1. Wählen Sie die Task-Card "Controls".
2. Klicken Sie auf das gewünschte Control.
   Informationen zum Control werden jetzt im Bereich "Vorschau" angezeigt.
3. Ziehen Sie das Control per Drag & Drop auf das Diagramm.
4. Verschalten Sie das Control.
   Um das Control zu verschalten, gibt es folgende Möglichkeiten:
   - Verbinden Sie das Control direkt mit einer Komponente.
   - Verbinden Sie das Control über einen Konnektor.
   - Verbinden Sie das Control über ein Kopplungssignal aus der Task-Card "Signale".

Grafiken

Mit Grafiken stellen Sie Signale mit einer individuellen Ansicht dar. Statische Grafiken geben eine bestimmte Darstellung vor; animierte Grafiken werden innerhalb der Simulation in Bezug auf Größe, Farbe und Position je nach ihrem aktuellen Zustand verändert.

Folgende statische Grafiken stehen zur Verfügung:

Grafikelement	Kurzbeschreibung	Einstellmöglichkeiten
Text	Eingabe von Text an beliebiger Stelle im Diagramm	Schriftart Schriftgröße Schriftfarbe Hintergrundfarbe Rahmenfarbe
Linie	Zeichnen einer geraden Linie. Halten Sie <Shift> gedrückt, um eine horizontale oder vertikale Line zu erstellen.	Linienfarbe Linienstärke
Rechteck	Zeichnen eines Rechtecks. Halten Sie <Shift> gedrückt, um ein Quadrat zu erstellen.	Rahmenfarbe Rahmenstärke Füllfarbe
Ellipse	Zeichnen einer Ellipse. Halten Sie <Shift> gedrückt, um einen Kreis zu erstellen.	Rahmenfarbe Rahmenstärke Füllfarbe
Polygonzug	Zeichnen eines Polygonzugs. Doppelklicken Sie oder drücken Sie <Space>, um das Zeichnen zu beenden.	Linienfarbe Linienstärke
Ellipsenbogen	Zeichnen eines Ellipsenbogens. Halten Sie <Shift> gedrückt, um einen Kreisbogen zu erstellen.	Linienfarbe Linienstärke
Bezierkurve	Zeichnen einer Bezierkurve	Linienfarbe Linienstärke

Um eine Grafik in das aktuelle Diagramm einzubinden, gehen Sie folgendermaßen vor:

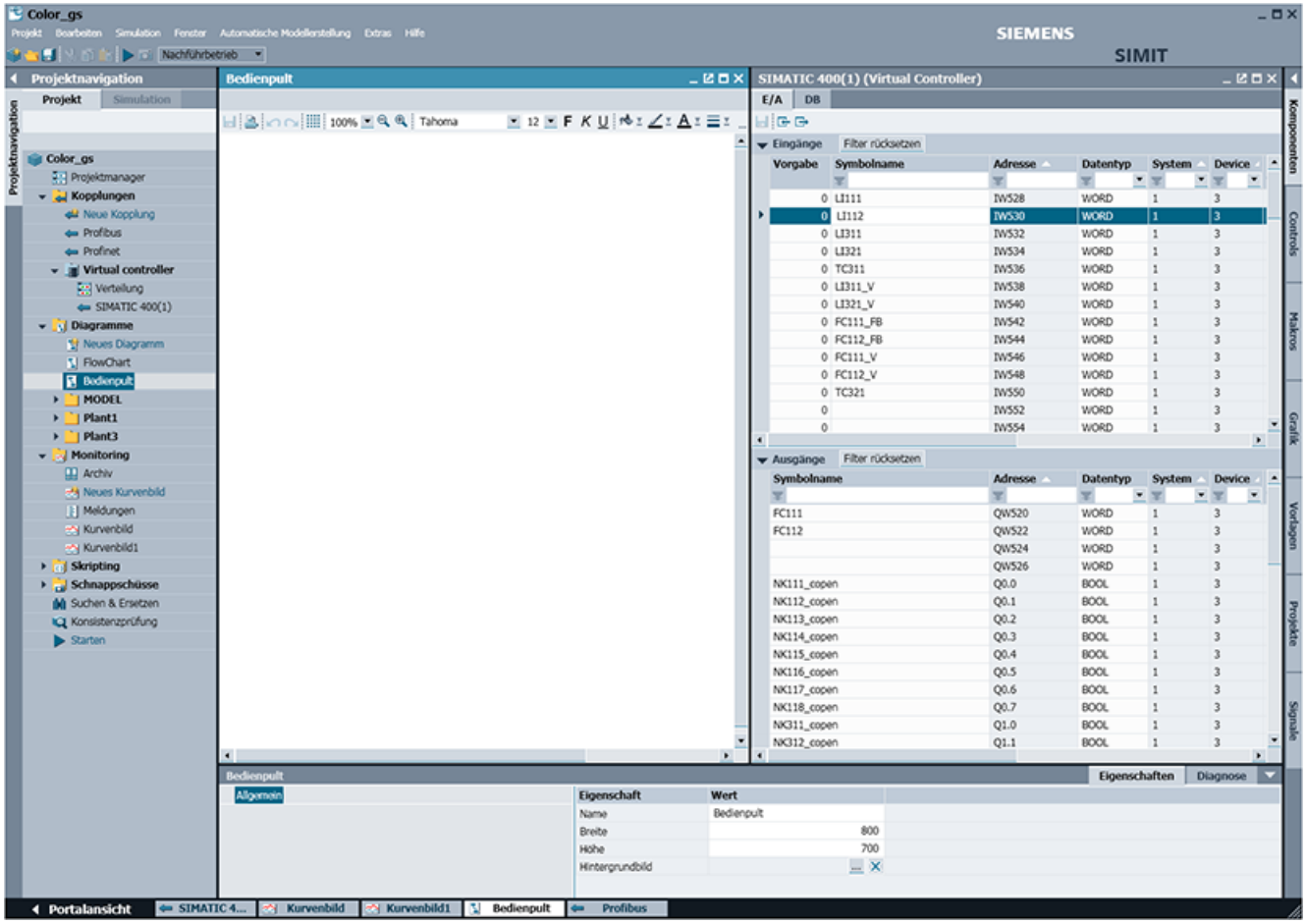
1. Wählen Sie die Task-Card "Grafik".
2. Klicken Sie auf die gewünschte Grafik.
   Informationen zur Grafik werden jetzt im Bereich "Vorschau" angezeigt.
3. Ziehen Sie die Grafik per Drag & Drop auf das Diagramm.
4. Verschalten Sie die Grafik.
   Um eine Grafik zu verschalten, gibt es folgende Möglichkeiten:
   - Verbinden Sie die Grafik direkt mit einer Komponente.
   - Verbinden Sie die Grafik über einen Konnektor.
   - Verbinden Sie die Grafik über ein Kopplungssignal aus der Task-Card "Signale".

Informationen zur Verwendung von animierten Grafiken finden Sie im Abschnitt: Grafiken visualisieren (Seite 227).

1.5.3 Visualisieren von Kopplungssignalen

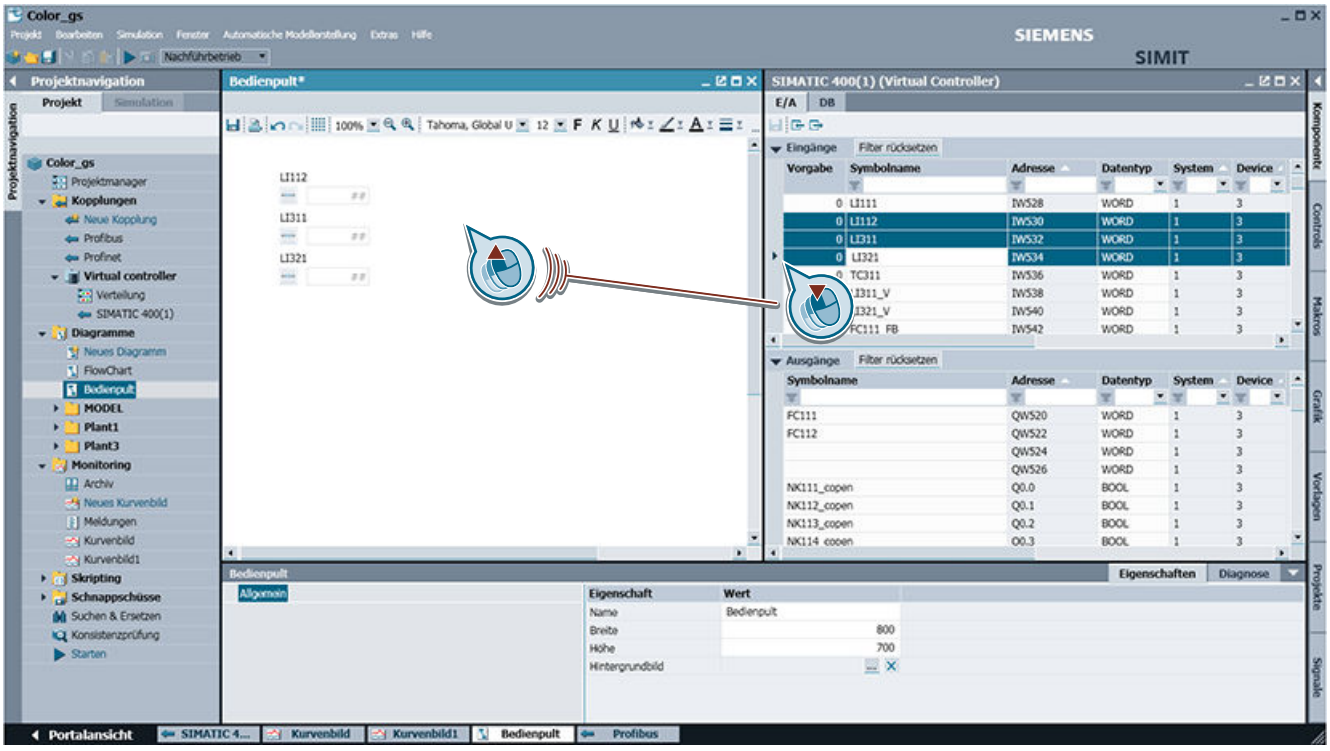
In Kopplungen wird bei laufender Simulation für jedes Signal ein Control mit Signaltrenner zur Verfügung gestellt. Um dieses Control auf einem Diagramm anzuordnen, gehen Sie folgendermaßen vor:

1. Teilen Sie den Arbeitsbereich mit dem Menübefehl "Fenster > Horizontal teilen".
2. Öffnen Sie die Kopplung und das Diagramm im Arbeitsbereich.
3. Legen Sie ein neues Diagramm an.
4. Öffnen Sie das Diagramm.
5. Öffnen Sie die Kopplung.
   Sie sehen das Diagramm und die Kopplung wie in der folgenden Abbildung im Arbeitsbereich:



6. Setzen Sie passende Filtereinstellungen im Kopplungsfenster und selektieren Sie das gewünschte Signal mit einem Mausklick.
   Um mehrere Signale zu selektieren, drücken Sie <Strg> (Einzelselektion) oder <Shift> (Bereichsselektion).

7. Selektieren Sie das Symbol "▶" in der ersten Spalte des Kopplungsfensters.
8. Ziehen Sie die selektierten Signale per Drag & Drop auf das Diagramm, wie in der folgenden Abbildung dargestellt:



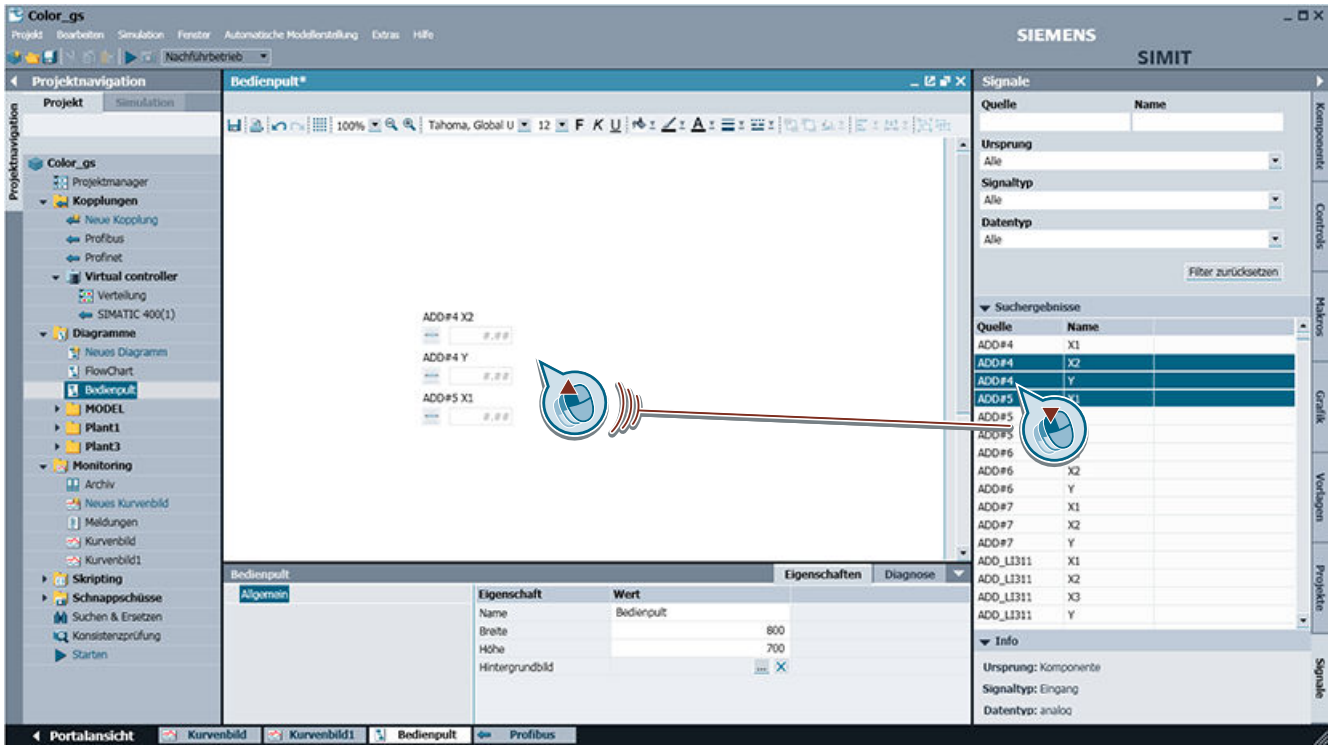
Für Signale mit dem Datentyp "Binär" wird ein Schalter, für alle anderen Signale eine Digitaleingabe als Control in der Kopplung erzeugt.

Kopplungssignale aus der Task-Card "Signale" auf das Diagramm ziehen

Alternativ können Sie Kopplungssignale auch aus der Task-Card "Signale" auf Diagramme ziehen.
Gehen Sie hierzu folgendermaßen vor:

1. Öffnen Sie das Diagramm im Arbeitsbereich.
2. Öffnen Sie die Task-Card "Signale".
   In dieser Task-Card finden Sie alle im Projekt gespeicherten Signale.

3. Zur Auswahl von Signalen setzen Sie entsprechende Filter für die Signalliste ein. Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Signale" (Seite 242).
4. Selektieren Sie dann die Signale in der Liste und ziehen Sie sie per Drag & Drop auf das Diagramm, wie in der folgenden Abbildung dargestellt:



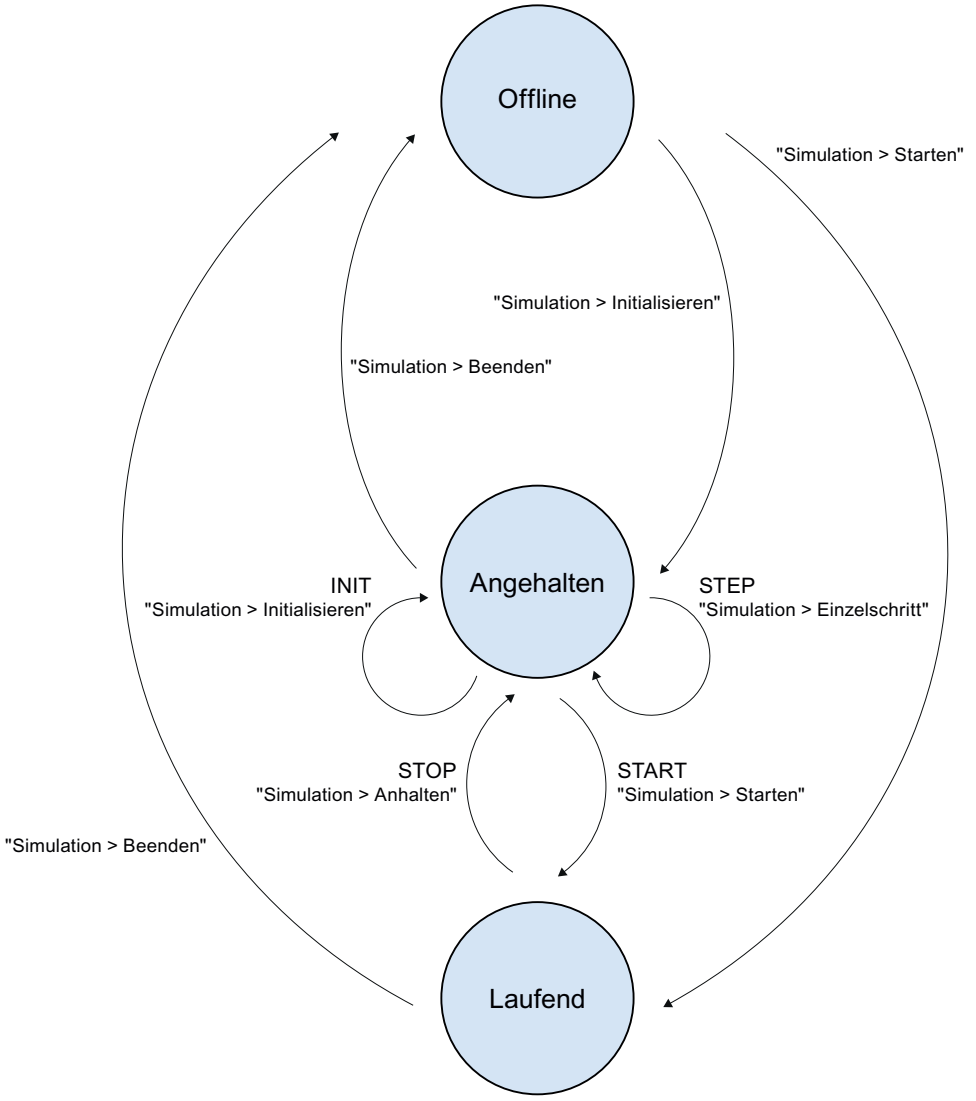
1.6 Ablauf einer Simulation

1.6.1 Die Zustände einer Simulation

Eine Simulation befindet sich immer in einem der folgenden Zustände:

- Offline
- Angehalten
- Laufend

Diese Zustände werden über das Menü "Simulation" verändert. In der folgenden Grafik werden die Zustände und die zugehörigen Menübefehle angezeigt, mit denen von einem Zustand in einen anderen gewechselt wird:



Die Simulation initialisieren

Die Simulation wird über den Menübefehl "Simulation > Initialisieren" initialisiert.

Bevor Sie die Simulation starten, können Sie eine Initialisierung der Simulation durchführen. Der Zustand wechselt auf "Angehalten". Die Initialisierung führt die Konsistenzprüfung durch, im Fehlerfall werden entsprechende Meldungen angezeigt. Weitere Informationen hierzu finden Sie im Abschnitt: Konsistenzprüfung (Seite 304).

Von einer initialisierten Simulation aus kann die Funktion "Einzelschritt" ausgeführt werden.

Die Simulation starten

Die Simulation wird mit folgenden Befehlen gestartet:

- Über das Symbol "" in der Symbolleiste
- Über den Menübefehl "Simulation > Starten"
- Über die Projektnavigation durch Doppelklick auf 

Nachdem die Simulation gestartet wurde, verändert SIMIT das Farbschema der Oberfläche von *blau* nach *orange*, der Zustand wechselt in "Laufend".

Weitere Informationen zur laufenden Simulation finden Sie im Abschnitt: Aktionen bei laufender Simulation (Seite 39).

Die Simulation anhalten

Die Simulation wird über den Menübefehl "Simulation > Anhalten" angehalten.

Eine Simulation kann zu jedem Zeitpunkt angehalten werden. Die Simulation wird auf dem aktuellen Stand angehalten und pausiert so lange, bis sie wieder gestartet oder beendet wird.

Beim Einsatz der SIMIT Unit führt das Anhalten der Simulation für mehr als ca. 30 Sekunden zum Abbruch der Verbindung zwischen der SIMIT Unit und SIMIT.

Einen Einzelschritt ausführen

Ein Einzelschritt ist nur dann verfügbar, wenn die Simulation sich im Zustand "Angehalten" befindet.

Ein Einzelschritt kann mit den folgenden Befehlen ausgeführt werden:

- Über den Menübefehl "Simulation > Einzelschritt"
- Über die Funktionstaste <F12>

Ein Einzelschritt führt die jeweils kleinste Zeitscheibe des Projekts aus und pausiert dann, bis ein weiterer Einzelschritt angestoßen wird oder einer der Befehle "Starten" oder "Beenden" ausgeführt wird.

Die Simulation beenden

Die Simulation wird mit den folgenden Befehlen beendet:

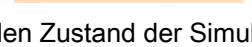
- Über das Symbol "" in der Symbolleiste
- Über den Menübefehl "Simulation > Beenden"

Der Zustand wechselt auf "Offline", das Farbschema der Oberfläche wechselt zurück auf *blau*.

1.6.2 Bedienen und Beobachten der Simulation

1.6.2.1 Aktionen bei laufender Simulation

Während die Simulation läuft, können Sie folgende Aktionen ausführen:

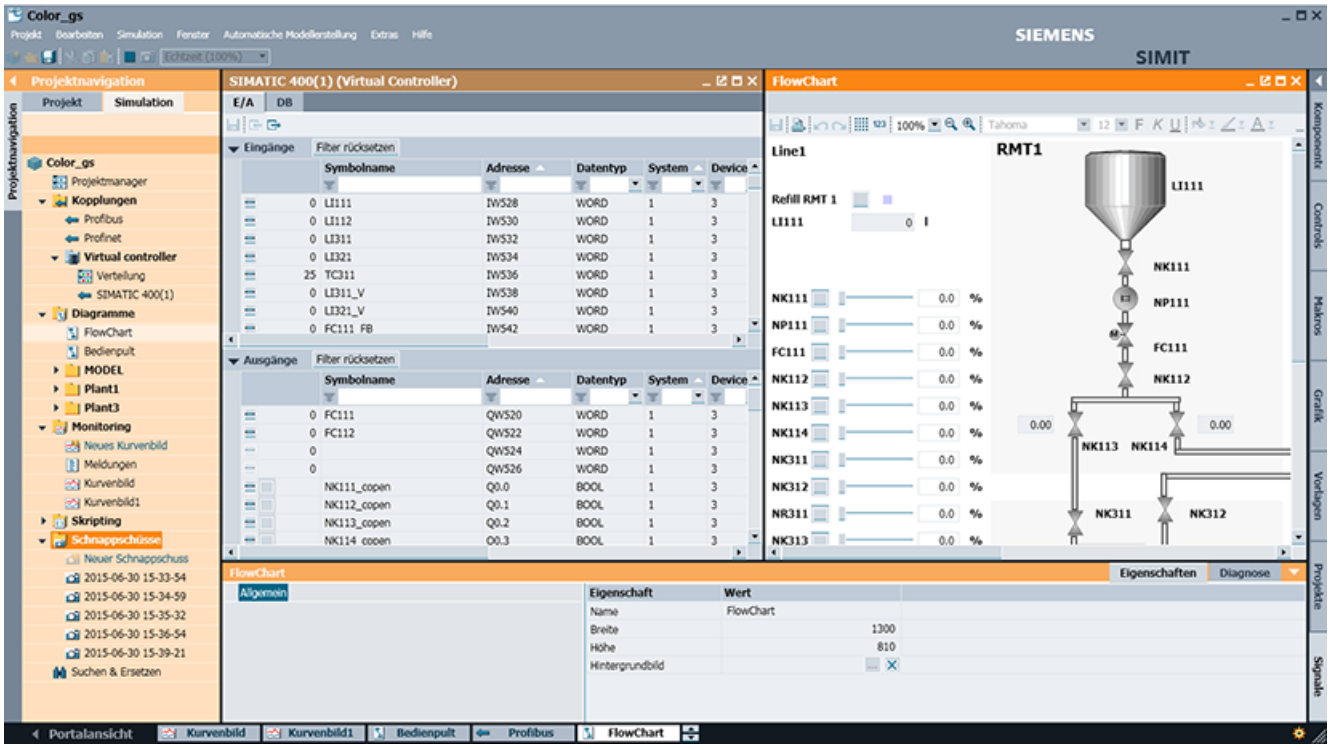
- Öffnen und Schließen von Diagrammen und Kopplungen
- Wertänderungen an den Ein- und Ausgängen markierter Komponenten im Eigenschaftsfenster verfolgen
  Aktivieren Sie über das Symbol " 123 " in der Symbolleiste die Anzeige von Wertänderungen von Ausgängen und verschalteten Eingängen im Diagramm. Die Wertänderungen werden zusätzlich auch in den Eigenschaften der Komponente angezeigt.
- Ein Kurvenbild erzeugen und anzeigen
  Weitere Informationen hierzu finden Sie im Kapitel: Kurvenbilder (Seite 290)
- Ein Skript erstellen
  Weitere Informationen hierzu finden Sie im Kapitel: Handhabung von Skripten (Seite 310)
- Einen Schnappschuss speichern
  Ein Schnappschuss wird mit einem der folgenden Befehle erstellt:
  - Über den Menübefehl "Simulation > Schnappschuss"
  - Über das Symbol "" in der Symbolleiste
  - Mit einem Doppelklick auf den Eintrag "

    " in der Projektnavigation

  Der Schnappschuss speichert den aktuellen Zustand der Simulation. Jeder Schnappschuss wird im Ordner "Schnappschüsse" (

) in der Projektnavigation gespeichert und erhält als Namen das Datum und die Uhrzeit zum Speicherzeitpunkt. Sie können jeden Schnappschuss umbenennen, in Unterordner verschieben oder löschen.
  Wenn Sie einen Schnappschuss laden, wird die Simulation auf dem im Schnappschuss gespeicherten Zustand gesetzt und die Simulation wird aus diesem Zustand heraus fortgesetzt. Dieser Vorgang startet automatisch die Simulation.

Die Oberfläche erhält in den Zuständen "Angehalten" und "Laufend" das Farbschema *orange*, wie in der folgenden Abbildung dargestellt:



1.6.2.2 Anzeige der Simulationslast

Simulationen werden zyklisch berechnet. Die Zykluszeit wird über eine Zeitscheibe eingestellt. Jede Komponente eines Diagramms und jede Kopplung wird einer von mehreren verfügbaren Zeitscheiben zugeordnet. Die Lösungsverfahren werden in derjenigen Zeitscheibe berechnet, in der die zugehörigen Komponenten liegen. Die absolute Zykluszeit für eine Zeitscheibe wird im Projektmanager eingestellt.

Die Zykluszeit gibt an, in welchem Zeitraster die Berechnungen ausgeführt werden sollen, z. B. alle 100 ms. Wenn aber innerhalb der Zykluszeit nicht alle Berechnungen der Zeitscheibe durchgeführt werden können, kann dieses Zeitraster nicht eingehalten werden.

Um zu beurteilen, wie stark der Rechner ausgelastet ist, ermittelt SIMIT die Simulationslast. Sie gibt das prozentuale Verhältnis von tatsächlicher Rechenzeit zu eingestellter Zykluszeit an. Damit kann ermittelt werden, ob die Zykluszeit für alle Berechnungen ausreicht. Es wird nur der insgesamt ungünstigste Wert angezeigt.

Beispiel:

	Zeitscheibe 1	Zeitscheibe 2
Eingestellte Zykluszeit	100 ms	200 ms
Berechnungsdauer	60 ms	80 ms
Simulationslast	60 %	40 %
Simulationslast gesamt (Lastwert)	60 %	

Bedeutung des prozentualen Lastwerts:

- **Lastwert deutlich unter 50 %**:
  Die Simulation kann in den eingestellten Zykluszeiten berechnet werden.
  Die Daten werden planmäßig zwischen dem Simulationsmodell und den Kopplungen ausgetauscht.
- **Lastwert erreicht oder überschreitet 50 %:**
  Die Simulation kann in den eingestellten Zykluszeiten berechnet werden.
  Die Daten werden nicht mehr zeitversetzt zwischen Simulationsmodell und Kopplungen ausgetauscht. Es entstehen zusätzliche Zyklusversätze.
  Die Simulationswerte werden erst in einem späteren Rechenschritt an die Kopplungen weitergereicht bzw. aus den Kopplungen gelesen.
  Wenn in einem Simulationsprojekt keine Kopplungen enthalten sind, dann ist die 50 %-Grenze irrelevant.
- **Lastwert erreicht oder überschreitet 100 %:**
  Die Simulation ist in Überlast.
  Das Simulationsmodell kann nicht mehr in der vorgesehenen Zykluszeit berechnet werden. Rechenschritte fallen aus.

Die Simulationslast wird bei laufender Simulation über ein Symbol im unteren rechten Bereich dargestellt. Die aktuelle Simulationslast kann anhand des grünen Kreissegments abgelesen werden (Vollkreis entspricht 100 %):



Lastschwankungen

Der Lastwert ist nicht immer konstant. Eine schwankende Last kann folgende Ursachen haben:

- Das Simulationsmodell selbst kann projektierungsbedingt in jedem Rechenschritt eine andere Last erzeugen.
- Eine Zeitscheibe wird durch eine Zeitscheibe mit höherer Priorität unterbrochen.
- Neben SIMIT laufen noch weitere Anwendungen im Hintergrund.
- Interne Vorgänge sowohl im Betriebssystem als auch in der von SIMIT genutzten .NET-Umgebung können die Simulationsberechnung verzögern.

Um die grafische Darstellung an der Oberfläche ruhig zu halten, werden die Lastwerte über mehrere Zeitschritte hinweg geglättet.

Hinweis

Falls der Lastwert immer wieder zwischen zwei Werten hin- und herspringt, dann kann das daran liegen, dass der Prozessor Ihres Rechners aus Energiespargründen ständig die Taktfrequenz ändert. Sie können das vermeiden, indem Sie beispielsweise die System-Einstellungen Ihres Rechners anpassen. Dazu wählen Sie "Systemsteuerung > Energieoptionen > Energiesparplan > Erweiterte Energieeinstellungen > Höchstleistung".

Hinweis

Der Windows-Taskmanager ist nicht dazu geeignet, die Simulationslast zu beurteilen. Zum einen ist die Verteilung des Simulationsmodells auf mehrere Prozessoren oder Prozessorkerne zu berücksichtigen, zum anderen sagt der Taskmanager nur etwas über die Rechenleistung aus, nicht jedoch, ob beispielsweise aufgrund von Verzögerungen in der Kommunikation die vorgesehene Zykluszeit nicht eingehalten werden kann.

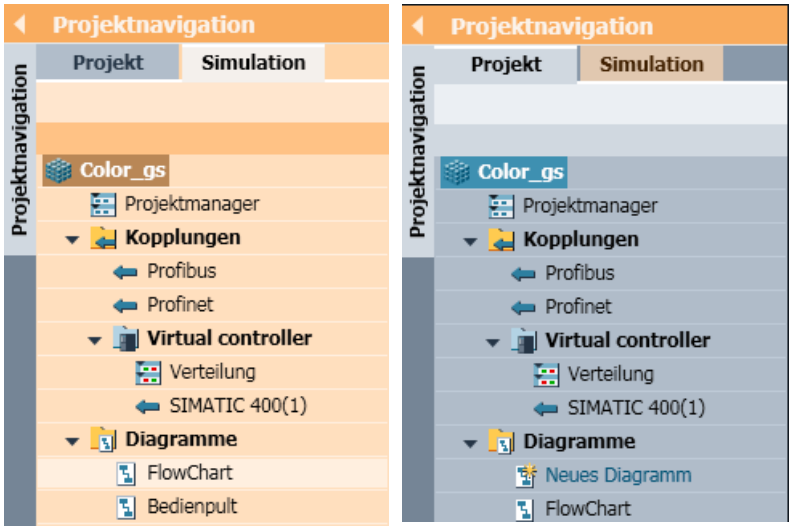
Hinweis

Wenn die Simulationslast zu hoch ist, können Sie Informationen zur Ursache über die Komponente "SimulationLoad" erhalten.

Weitere Informationen zur Komponente finden Sie im Abschnitt: SimulationLoad – Simulationslast (Seite 469)

1.6.2.3 Änderungen im Simulationsprojekt bei laufender Simulation

Sie können bei laufender Simulation Änderungen am Simulationsprojekt vornehmen. Zum Ändern bei laufender Simulation schalten Sie in der Projektnavigation zwischen "Simulation" und "Projekt" um.



In beiden Projektfenstern können Sie Diagramme öffnen, allerdings mit unterschiedlicher Bedeutung:

- Wenn Sie sich im blau dargestellten Projektfenster befinden, werden die Diagramme zum Bearbeiten geöffnet. Die Änderungen, die Sie vornehmen, werden beim nächsten Simulationsstart oder mit dem Menübefehl "Änderungen aktivieren" wirksam.
- Wenn Sie sich im orange dargestellten Simulationsfenster befinden, werden die Diagramme zum Bedienen geöffnet. Sie entsprechen in ihrem Inhalt der laufenden Simulation und können nicht verändert werden.

Das gleiche Diagramm kann in beiden Modi gleichzeitig geöffnet sein. Beim Beenden der Simulation wird das aus dem Simulationsbaum geöffnete Diagramm automatisch geschlossen.

Während die Simulation läuft, gibt es folgende Einschränkungen in der Bedienung von SIMIT:

- Es können keine neuen Kopplungen angelegt oder vorhandene gelöscht werden.
- In den Kopplungen können ausschließlich implizite Verschaltungen bearbeitet werden.
- Alle Funktionen aus der "Automatischen Modellerstellung" sind deaktiviert.

Modelländerungen vorbereiten und aktivieren

Modelländerungen können auch während der Simulation aktiviert werden.

Gehen Sie hierzu folgendermaßen vor:

1. Wechseln Sie bei laufender Simulation in das Projektfenster.
2. Nehmen Sie die gewünschten Änderungen am Simulationsmodell vor.
3. Klicken Sie auf "Änderungen vorbereiten". Die Vorbereitung kann einige Minuten dauern. Nachdem die Funktion "Änderungen vorbereiten" beendet ist, wechselt der Menübefehl in "Änderungen aktivieren".
4. Klicken Sie auf "Änderungen aktivieren", um die Änderungen in die laufende Simulation zu übernehmen.

Hinweis

Während die Änderungen aktiviert werden, verarbeitet das Simulationsmodell für einige Zeit keine Werte. Während dieser Zeit zeigen Peripheriegeräte eventuell keine Reaktion. Die Zeitspanne hängt von der Größe des Simulationsmodells ab.

Wenn der aktuelle Simulationszustand auf die neue Projektierung übertragen wird, werden die Werte anhand des Signalnamens identifiziert und zugewiesen. Werte, für die es keine Entsprechung gibt, werden mit Initialwerten belegt.

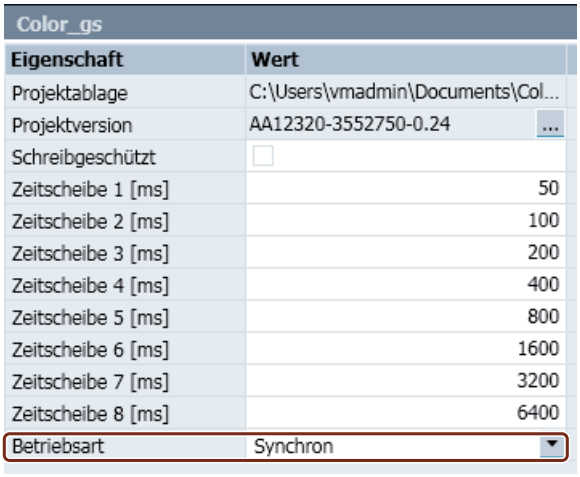
Werte von online änderbaren Parametern und Vorbelegungen werden bei einer Aktivierung nicht berücksichtigt, sondern die Werte aus der laufenden Simulation beibehalten. Um solche Wertänderungen zu aktivieren, starten Sie die Simulation neu.

1.6.3 Einstellungen zum Simulationsablauf

1.6.3.1 Betriebsarten

Das Simulationsmodell und die Kopplungen können nach den drei unterschiedlichen Betriebsarten *asynchron, synchron* und *bus-synchron* bearbeitet werden.

Welche Betriebsart aktuell geeignet ist, hängt vom Projekt ab. Sie wird im Eigenschaftsfenster im Projektmanager eingestellt:



Color_gs	
Eigenschaft	**Wert**
Projektablage	C:\Users\vmadmin\Documents\Col...
Projektversion	AA12320-3552750-0.24 ...
Schreibgeschützt	☐
Zeitscheibe 1 [ms]	50
Zeitscheibe 2 [ms]	100
Zeitscheibe 3 [ms]	200
Zeitscheibe 4 [ms]	400
Zeitscheibe 5 [ms]	800
Zeitscheibe 6 [ms]	1600
Zeitscheibe 7 [ms]	3200
Zeitscheibe 8 [ms]	6400
Betriebsart	Synchron

Weitere Informationen zu den Betriebsarten finden Sie in den Abschnitten:

- Asynchrone Betriebsart (Seite 44)
- Synchrone Betriebsart (Seite 46)
- Bus-synchrone Betriebsart (Seite 47)

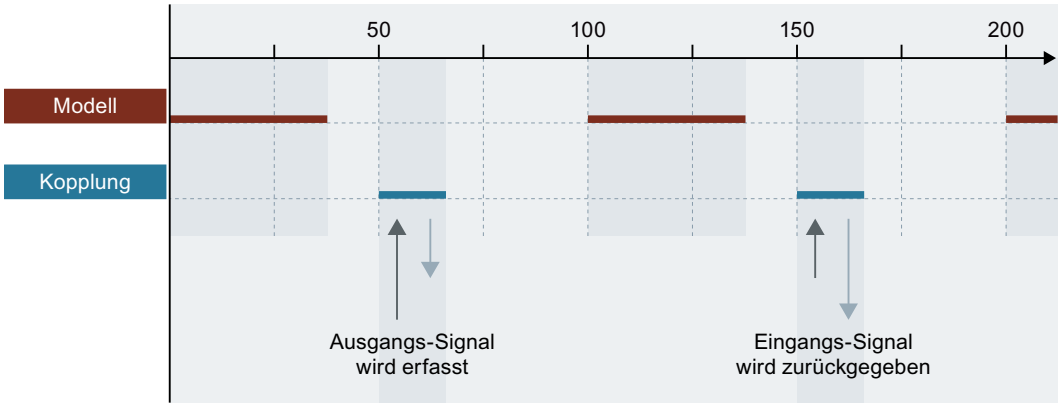
Asynchrone Betriebsart

Die Modellberechnung der einzelnen Zeitscheiben und die Bearbeitung der Kopplungen erfolgen zeitgesteuert.

Wenn das Simulationsmodell einer Zeitscheibe nicht in der vorgesehenen Zeit abschließt, dann fallen ein oder mehrere Bearbeitungszyklen aus und die anderen Zeitscheiben werden nicht berechnet. Erst wenn alle Zeitscheiben wieder rechnen können, wird die Berechnung des Simulationsmodells fortgesetzt.

Wenn SIMIT eine Kopplung nicht in der vorgesehenen Zeit berechnet, dann fallen zwar auch hier ein oder mehrere Bearbeitungszyklen aus, aber die Berechnung der Kopplungen in anderen Zeitscheiben und die Berechnung des Simulationsmodells behindert das nicht. Dadurch blockieren Kopplungen durch eventuelle Kommunikationsprobleme nicht die Gesamtbearbeitung.

Um möglichst schnelle Reaktionszeiten zu erzielen, werden Kopplungen zeitversetzt gegenüber der Modellberechnung eingeplant:



Wenn die Modellberechnung oder der Datenaustausch der Kopplungen länger als die Hälfte der Zykluszeit dauert, dann führt diese Zeitversetzung nicht zu schnelleren Reaktionszeiten.

Signalaustausch über Kopplungen bei asynchroner Betriebsart

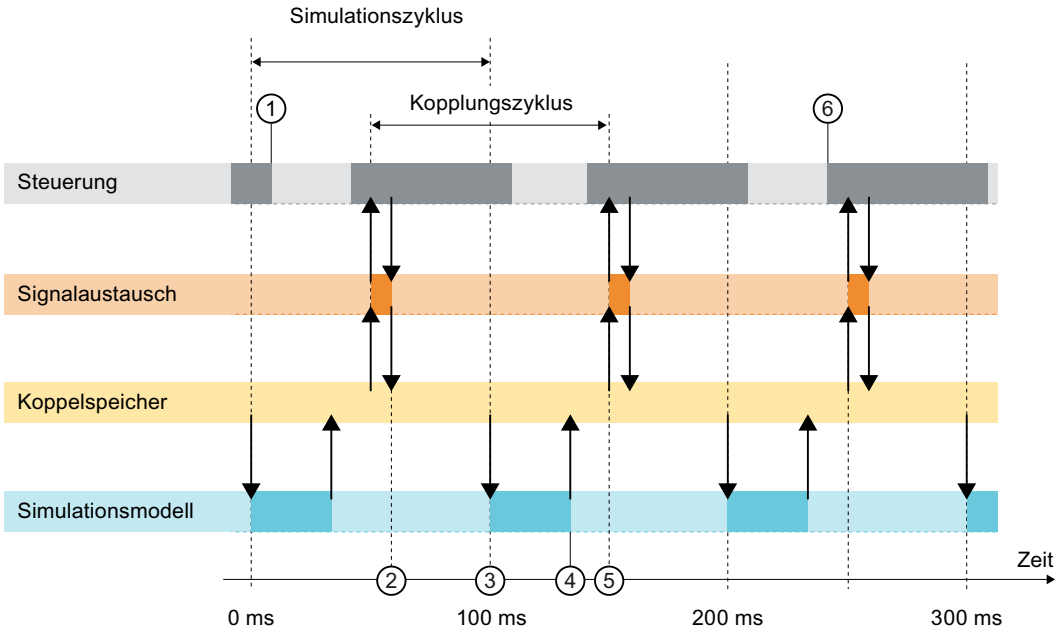
Die Modellberechnung wie auch der Signalaustausch in der Kopplung werden zyklisch vom Steuersystem angestoßen. Die Zykluszeit, mit der das Modell berechnet wird, ist in den Komponenten bzw. den Projekteigenschaften festgelegt. In den Kopplungseigenschaften (Kopplungskonfigurator) wird unabhängig davon die Zykluszeit festgelegt, mit der der Signalaustausch erfolgt. Verzögerungen im Signalaustausch einer Kopplung haben damit keine Rückwirkung auf die Modellberechnung, sondern nur auf andere Kopplungen, die in der gleichen Zykluszeit abgearbeitet werden.

Um Verzögerungen im Signalaustausch möglichst klein zu halten, wird der Triggerzeitpunkt der Kopplung pauschal gegen den Triggerzeitpunkt des Simulationsmodells um die Hälfte des eingestellten Simulationszyklus verschoben.

Hinweis

Die Zykluszeit der Kopplung (Kopplungszyklus) und die Zykluszeit für die Berechnung des Simulationsmodells (Simulationszyklus) sollten beide auf den gleichen Wert eingestellt werden.

Im folgenden Beispiel wird sowohl die Steuerung als auch die Modellberechnung mit einer Zykluszeit von 100 ms getaktet. Da Steuerung und Simulation nicht synchronisiert sind, erfolgt die Taktung in der Regel zeitlich gegeneinander versetzt. Dabei dauert der Simulationszyklus nicht länger als die Hälfte des Modellberechnungszyklus:



① Zu diesem Zeitpunkt hat die Steuerung ihren Rechenzyklus beendet. An den Ausgängen der Steuerung stehen weitere Signale zur Verfügung.

② Die Ausgangssignale der Steuerung werden in den Koppelspeicher von SIMIT übernommen. Hier stehen sie für die Modellberechnung zur Verfügung.

③ Die Modellberechnung wird wieder angestoßen.

④ Ende des Modellberechnungszyklus. Die berechneten Eingangssignale für die Steuerung sind jetzt im Koppelspeicher verfügbar.

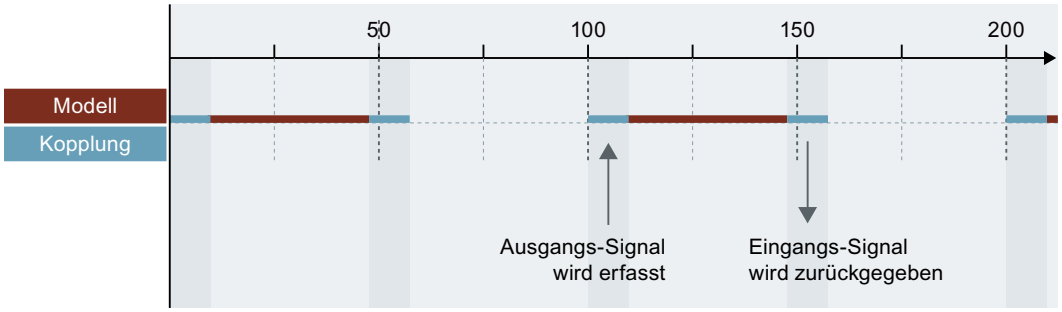
⑤ Mit dem nächsten Takt der Kopplung werden diese Eingangssignale an die Steuerung übertragen.

⑥ Falls die Steuerung ihre Eingänge nur zu Beginn eines Zyklus auswertet, werden die übertragenen Eingangssignale erst jetzt im Steuerprogramm verwendet. Die Reaktion der Simulation findet damit erst zwei Steuerungszyklen später statt.

Synchrone Betriebsart

Bei synchroner Betriebsart werden Teilmodelle und Kopplungen in einer genau vorgegebenen Reihenfolge berechnet. Die nächste Aktion startet erst, wenn die vorherige Aktion fertig ist.

Die Kopplung ist aufgetrennt in das Erfassen von Ausgangssignalen und das Schreiben von Eingangssignalen:



Das führt bei kleinen Zykluszeiten zu besseren Reaktionszeiten. Allerdings kann jedes Teilmodell und jede Kopplung die gesamte Simulationsausführung blockieren.

Signalaustausch über Kopplungen bei synchroner Betriebsart

Bei synchroner Betriebsart erfolgen Modellberechnung und Signalaustausch in vorgegebener Reihenfolge. Die Kopplung erfasst das Ausgangssignal und schreibt erst nach einer Modellberechnung das Eingangssignal zurück. Im Gegensatz zur asynchronen Betriebsart können keine Steuerungszyklen ausfallen.

Bus-synchrone Betriebsart

Bei bus-synchroner Betriebsart stellt SIMIT sicher, dass alle an der Simulation beteiligten Komponenten den gleichen synchronisierten Simulationsfortschritt haben. Deshalb ist diese Betriebsart für Anwendungen mit Echtzeitanforderung geeignet. Für Anwendungen ohne Echtzeitanforderung genügt die asynchrone Betriebsart oder die synchrone Betriebsart.

Voraussetzung:

- Die bus-synchrone Betriebsart steht für die Kopplungen PLCSIM Advanced und MCD zur Verfügung.
- Beim bus-synchronen Betrieb muss eine Zeitscheibe der Taktzeit entsprechen.

In der bus-synchronen Betriebsart sind nur folgende Steuerbefehle aus dem Menü "Simulation" verfügbar:

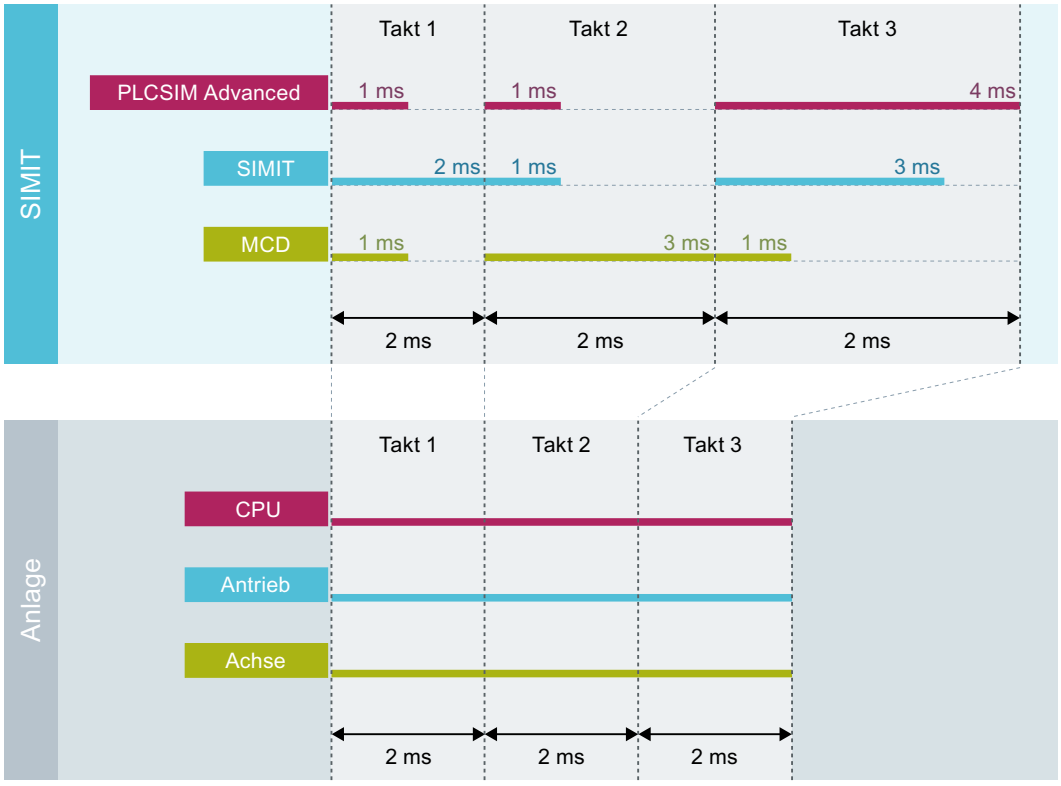
- Starten
- Beenden

Beispiel

Im abgebildeten Beispiel beträgt die Taktzeit 2 ms und eine Zeitscheibe des Projekts muss auf 2 ms eingestellt sein.

Im oberen Teil der Abbildung sehen Sie, wie lange PLCSIM Advanced, SIMIT und MCD für ihre Berechnungen benötigen. Erst wenn alle drei Berechnungen abgeschlossen sind und die Ergebnisse ausgetauscht wurden, beginnen die Berechnungen für den nächsten Takt. Auch wenn die Berechnungen in der Simulation länger als die Taktzeit von 2 ms benötigen, entspricht die Berechnungsdauer in der realen Anlage immer der eingestellten Taktzeit.

Im unteren Teil der Abbildung sehen Sie die Taktung von CPU, Antrieb und Achse in der realen Anlage.



1.6.3.2 Zeitscheiben

Definition

Simulationen werden zyklisch berechnet. Die Zykluszeit gibt an, in welchem Zeitraster die Berechnungen ausgeführt und Daten austauscht werden sollen. Die Zykluszeit wird über eine von acht verfügbaren Zeitscheiben eingestellt. Die absolute Zykluszeit für eine Zeitscheibe wird im Projektmanager eingestellt und ist für das gesamte Projekt gültig. Die minimal einstellbare Zykluszeit beträgt 1 ms.

Zeitscheiben von Diagrammen

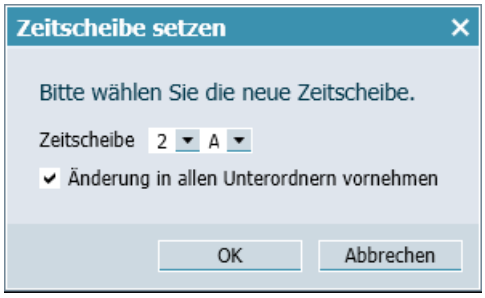
Jede Komponente eines Diagramms wird einer Zeitscheibe zugeordnet. Die Lösungsverfahren werden in derjenigen Zeitscheibe berechnet, in der die zugehörigen Komponenten liegen. Die Zykluszeit gibt an, in welchem Zeitraster die Berechnungen ausgeführt werden sollen, z. B. alle 100 ms. Wenn aber innerhalb der Zykluszeit nicht alle Berechnungen der Zeitscheibe durchgeführt werden können, kann dieses Zeitraster nicht eingehalten werden.

Komponenten eines Diagramms haben für jede Zeitscheibe acht verfügbare Unterzeitscheiben A bis H. Alle Unterzeitscheiben laufen in der Zykluszeit der zugehörigen Hauptzeitscheibe. Die Unterzeitscheiben ermöglichen eine bessere Aufteilung der Simulation, um Mehrkernprozessoren besser auszulasten.

Die Zeitscheibe und die Unterzeitscheibe einer Komponente eines Diagramms setzen Sie in den Eigenschaften der Komponente unter "Allgemein".



Die Zeitscheibe und die Unterzeitscheibe eines Diagramms oder eines Diagrammordners setzen Sie über das Kontextmenü im Dialogfeld "Zeitscheibe setzen".



Zeitscheiben von Kopplungen

Jede Kopplung wird einer Zeitscheibe zugeordnet. Die Zeitscheibe bestimmt den Zyklus, mit dem die Kopplung Daten austauscht. Kopplungen laufen immer in der Hauptzeitscheibe.

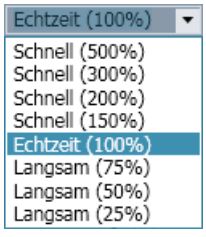
Die Zeitscheibe einer Kopplung setzen Sie in den Eigenschaften der Kopplung.

Die Zeitscheibe und die Unterzeitscheibe eines Kopplungsordners setzen Sie über das Kontextmenü im Dialogfeld "Zeitscheibe setzen".

1.6.3.3 Beschleunigung und Verzögerung der Simulation

Um den Ablauf der Simulation zu verlangsamen oder zu beschleunigen, können Sie das Verhältnis von Simulationszeit zu tatsächlichem Zeitablauf verändern.

Dieser Wert wird über eine Klappliste in der Symbolleiste eingestellt:



Die maximale Beschleunigung ist so begrenzt, dass die kürzeste Zykluszeit des Projekts unter Berücksichtigung der Beschleunigung mindestens 1 ms beträgt. Nach Beendigung der Zeitverschiebung wird die Zeit wieder auf die Rechnerzeit synchronisiert.

Je nach Zykluszeit ist also keine Beschleunigung möglich. Es werden nur passende Werte zur Auswahl angeboten.

Hinweis

Zeitverhalten

Timer OBs im Virtual Controller und Zeitfunktionen, die von Timer-OBs im Virtual Controller abhängig sind, können zu irregulärem Zeitverhalten führen.

1.6.3.4 Einstellbare Time-Out-Zeiten

Verschiedene Vorgänge werden von SIMIT überwacht und beim Überschreiten einer voreingestellten Zeit abgebrochen, um die Bedienbarkeit von SIMIT zu gewährleisten.

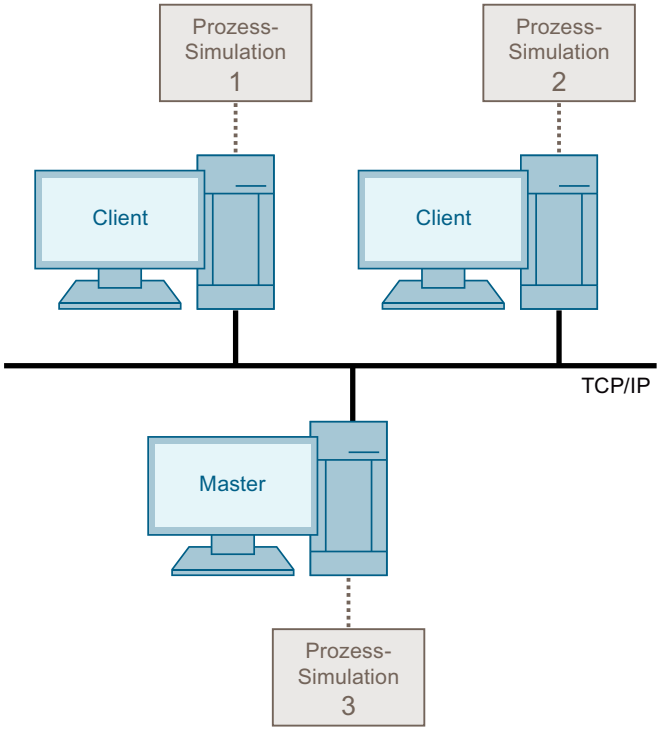
Diese Time-Out-Zeiten sind so dimensioniert, dass sie in der Regel nicht überschritten werden. Sollten Sie dennoch die Meldung erhalten, dass ein Time Out überschritten wurde, prüfen Sie zunächst, ob irgendein Fehlverhalten von SIMIT vorliegt. Falls nicht, können Sie die Time-Out-Zeiten in der Registry von Windows erhöhen. Sie finden die entsprechenden Schlüssel unter "HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\SIMIT\8.0\Timeout" bzw. "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Siemens\SIMIT\8.0\Timeout" auf 64-Bit Betriebssystemen.

1.7 Verteilte Simulation

1.7.1 Funktionsweise der verteilten Simulation

Funktionsweise

Die Komplexität eines Simulationsprojekts ist begrenzt durch die Rechenleistung des Simulations-PCs. Um komplexe Simulationsprojekte mit SIMIT zu realisieren, verteilen Sie ein Simulationsprojekt auf bis zu 16 Client-PCs. Die Simulation steuern Sie zentral vom Master-PC aus.



Typische Szenarien für den Einsatz der verteilten Simulation sind z. B.

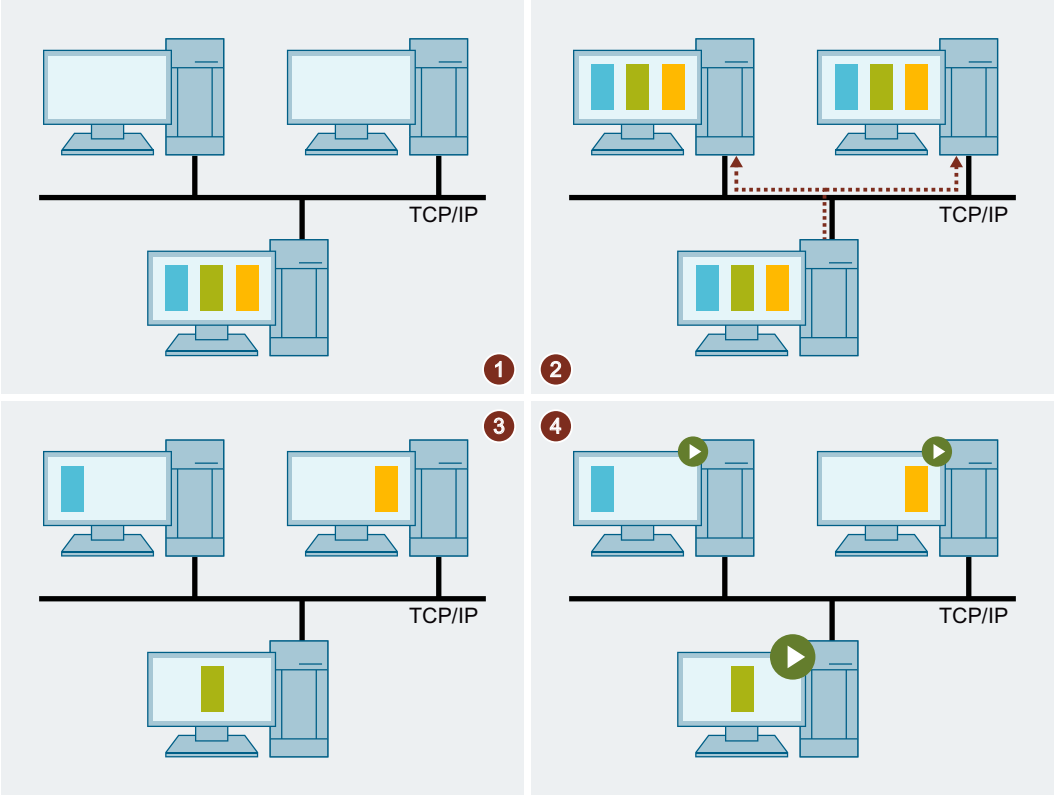
- Anlagenteile agieren möglichst unabhängig voneinander
- Wenig bis kein zeitkritischer Datenaustausch zwischen den Anlagenteilen
- Kraftwerk mit mehreren Blöcken
- Herstellungsprozess mit mehreren Linien

1.7.2 Projektierung einer verteilten Simulation

Beispiel zur Projektierung einer verteilten Simulation

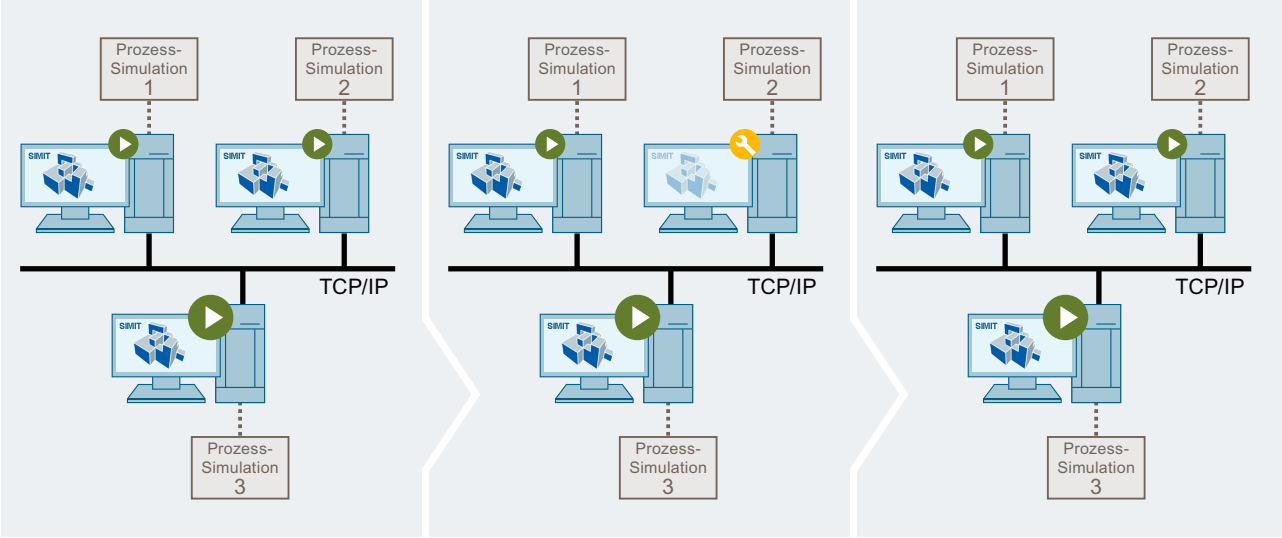
Ein Simulationsprojekt wird auf Grundlage der Anlagenstruktur in mehrere Teilprojekte aufgeteilt. Dazu wird das Simulationsprojekt auf die beteiligten Client-PCs kopiert und im gleichen lokalen Pfad wie auf dem Master-PC abgelegt. Auf jedem Client-PC wird das Simulationsprojekt individuell angepasst. Jedes Simulationsprojekt ist eigenständig lauffähig.

Die folgende Abbildung zeigt das prinzipielle Vorgehen:



① Simulationsprojekt auf Master-PC anlegen
② Simulationsprojekt auf Client-PCs kopieren
③ Simulation projektieren und testen pro Client-PC und Master-PC
④ Simulation auf Master-PC starten

Ablauf einer verteilten Simulation



① Wenn Sie die Simulation auf dem Master-PC starten, starten auch die Teilsimulationen auf den Client-PCs.

② Um ein Teilprojekt auf einem Client-PC zu ändern, deaktivieren Sie die Fernsteuerung des Client-PCs durch den Master-PC. Anschließend ändern Sie das Modell auf dem Client-PC.

③ Wenn Sie danach die Fernsteuerung des Client-PCs durch den Master-PC wieder herstellen, laufen alle Teilsimulationen wieder synchron.

Siehe auch

Verteilte Simulation auf einem Client-PC ändern (Seite 57)

Verteilte Simulation starten (Seite 56)

1.7.3 Datenaustausch bei einer verteilten Simulation

Verteilte Simulation mit Datenaustausch zwischen Client-PCs und Master-PC

Wenn sich Ihre Simulation nicht in vollständig unabhängige Teilsimulationen aufteilen lässt, nutzen Sie die verteilte Simulation mit Datenaustausch. Teilen Sie Ihr Projekt so in Teilprojekte auf, dass die Teilsimulationen möglichst wenig Daten austauschen.

Für den Datenaustausch nutzen Sie z. B. die OPC DA-Kopplung. Im folgenden Beispiel ist der Master-PC der verteilten Simulation gleichzeitig der OPC DA-Server. Die Client-PCs der verteilten Simulation sind die OPC DA-Clients.

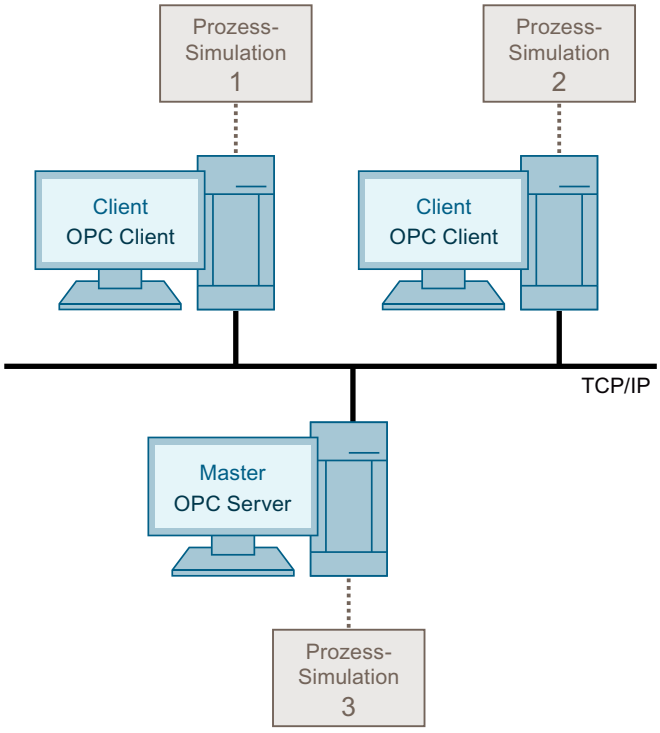


Bild 1-1 Verteilte Simulation mit Datenaustausch zwischen Client-PCs und Master-PC

1.7.4 Firewall einrichten

Einleitung

Stellen Sie die Firewall ein, damit SIMIT SP sich mit folgenden Systemen verbinden kann:

- Weitere Instanzen von SIMIT SP in einer verteilten Simulation
- Verteilte Virtual Controller
- Fremdsysteme, die eine Verbindung mit SIMIT SP über die Simulationssteuerung (RCI) herstellen

ACHTUNG
Beachten Sie die Security-Hinweise im Vorwort
Änderungen an der Firewall können die Sicherheit Ihres Systems beeinflussen!
Kontaktieren Sie gegebenenfalls Ihren Systemadministrator, bevor Sie die nachfolgend beschriebenen Schritte durchführen.

Voraussetzung

- SIMIT ist installiert
- Alle Teilnehmer befinden sich im gleichen Netzwerk

Vorgehen

Um die Firewall einzurichten, gehen Sie folgendermaßen vor:

1. Öffnen Sie die "Windows Defender Firewall mit erweiterter Sicherheit", z. B. indem Sie "wf.msc" in der Eingabeaufforderung eingeben und mit "Enter" bestätigen.
2. Klicken Sie auf "Eingehende Regeln".
3. Doppelklicken Sie "SIMIT-CS Manager".
   Das Fenster "Eigenschaften von SIMIT-CS Manager" wird geöffnet.
4. Wählen Sie den Reiter "Bereich".
5. Wählen Sie unter "Remote-IP-Adresse" "Beliebige IP-Adresse". Alternativ fügen Sie die IP-Adressen der beteiligten Systeme in die Liste der zugelassenen Remote-IP-Adressen ein.
6. Bestätigen Sie mit "OK".

Ergebnis

Die Firewall ist so eingestellt, dass sich SIMIT SP mit anderen Systemen verbinden kann.

Siehe auch

Vorwort (Seite 3)

1.7.5 Bildschirm-Verknüpfung "SIMIT Client" anlegen

Einleitung

Um SIMIT im Client-Modus zu starten, erzeugen Sie eine Bildschirm-Verknüpfung mit einem entsprechenden Übergabeparameter.

Syntax des Übergabeparameters: /[M|m]:<IP-Adresse>:<Port>

- /M: Der Bediener des Client-PCs kann die Fernsteuerung deaktivieren.
- /m: Der Bediener des Client-PCs kann die Fernsteuerung nicht deaktivieren.
- <IP-Adresse>: IP-Adresse des Master-PCs
- <Port>: Portnummer des Master-PCs, Standardwert: 50800.
  Wenn ein Administrator den voreingestellten Port geändert hat, finden Sie den aktuellen Port auf dem Master-PC in der Registry unter [HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Siemens\SIMIT\8.0] im Schlüssel "uri".

Beispiel für vollständigen Übergabeparameter:

"C:\Program Files (x86)\Siemens\Automation\SIMIT\SIMIT\SIMIT SF\bin\SIMIT.exe / M:192.168.0.1:50800".

Voraussetzung

- SIMIT ist installiert.
- Die Firewall-Konfiguration des Master-PC lässt eingehende Verbindungen auf der Portnummer zu.

Vorgehen

Um eine Bildschirm-Verknüpfung "SIMIT Client" anzulegen, gehen Sie folgendermaßen vor:

1. Kopieren Sie auf dem Bildschirm die Verknüpfung "SIMIT SP".
2. Ergänzen Sie in den Eigenschaften der kopierten Verknüpfung den Übergabeparameter.
3. Benennen Sie die Verknüpfung in "SIMIT Client" um.

Ergebnis

Die Bildschirm-Verknüpfung "SIMIT Client" ist angelegt.

1.7.6 Verteilte Simulation starten

Voraussetzung

- Der Master-PC und die Client-PCs befinden sich im selben Netzwerk.
- Auf dem Master-PC und den Client-PCs ist ein Projekt gleichen Namens unter dem gleichen lokalen Pfad abgelegt.
- Die Bildschirm-Verknüpfung (Seite 55) "SIMIT Client" ist auf allen Client-PCs angelegt.

Vorgehen

Um eine verteilte Simulation zu starten, gehen Sie folgendermaßen vor:

1. Starten Sie SIMIT auf dem Master-PC.
2. Starten Sie auf den Client-PCs SIMIT im Client-Modus.
3. Öffnen Sie das Projekt auf dem Master-PC.
4. Starten Sie die Simulation auf dem Master-PC.

Ergebnis

Die Simulation läuft auf dem Master-PC und auf den Client-PCs.

Siehe auch

Projektierung einer verteilten Simulation (Seite 52)

1.7.7 Verteilte Simulation auf einem Client-PC ändern

Einleitung

Während einer laufenden verteilten Simulation sind auf dem Client-PC keine Modelländerungen möglich. Um das Modell zu ändern, unterbrechen Sie die Verbindung zwischen Client-PC und Master-PC.

Voraussetzung

- Eine verteilte Simulation läuft auf dem Master-PC und den Client-PCs.
- Der Client-PC wurde mit dem Übergabeparameter "/M" gestartet.

Vorgehen

Um eine verteilte Simulation auf einem Client-PC zu ändern, gehen Sie folgendermaßen vor:

1. Deaktivieren Sie die Option "Fernsteuerung aktiv" in den Projekteigenschaften.
   Die Verbindung zum Master-PC wird getrennt. Die Simulation auf dem Client-PC läuft zunächst weiter.
2. Ändern Sie das Modell.
3. Starten Sie die Simulation mit Ihren Änderungen.
4. Wenn die Simulation lauffähig ist, bringen Sie die Simulation auf dem Client-PC in denselben Zustand wie die Simulation auf dem Master-PC.
5. Aktivieren Sie die Option "Fernsteuerung aktiv" in den Projekteigenschaften.

Ergebnis

Der Client-PC läuft wieder unter der Kontrolle des Master-PCs.

Siehe auch

Projektierung einer verteilten Simulation (Seite 52)

Kopplungen

2

2.1 Kopplungskonzept

2.1.1 Kopplungen in der Architektur von SIMIT

Einleitung

Eine Kopplung ist eine Schnittstelle zwischen SIMIT und einem Kopplungspartner, z. B. einem Automatisierungssystem. Eine Kopplung hat folgende Aufgaben:

- Signalaustausch mit dem Kopplungspartner
  Über die Kopplung werden die Ein-/Ausgangssignale (E/A-Signale) des Kopplungspartners mit SIMIT ausgetauscht.
- Koordination des Signalaustauschs zwischen SIMIT und dem Kopplungspartner

Kopplungsarten

In SIMIT werden drei Kopplungsarten unterschieden:

- Kopplung zu einem realen Controller der SIMATIC-Familie
- Kopplung zu einem emulierten Controller der SIMATIC-Familie
- Kopplung zu einem externen Partner

	Kopplung zu einem realen Controller der SIMATIC-Familie		Kopplung zu einem emulierten Controller der SIMATIC-Familie			Kopplung zu einem externen Partner
	SIMIT Unit	PRODAVE	Virtual Controller	PLCSIM	PLCSIM Advanced	–
Simulation des Busverhaltens	ja	nein	nein			nein
Simulation von Taktsynchronität	nein		nein		ja	nein[1]
Zugriff auf das Prozessabbild	ja		ja			nein
Verwendung von Hardware-Konfigurationsdaten aus STEP 7	ja	nein	ja	nein	nein	nein
Import und Export von Signalen	ja		ja			ja[2]

[1] Die MCD-Kopplung unterstützt Taktsynchronität.

[2] Abhängig vom externen Kopplungspartner

Jede Kopplung besitzt einen eigenen Konfigurator zum Anlegen und Bearbeiten der Kopplung.

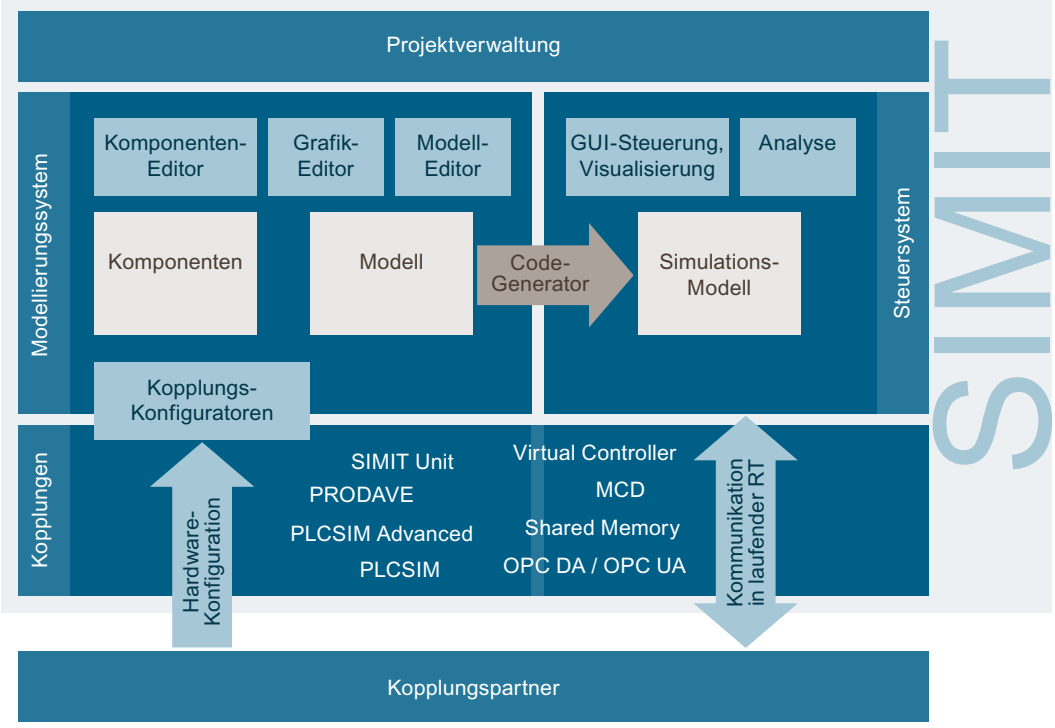
Hinweis

In einem SIMIT-Projekt sind maximal 16 externe Kopplungen verwendbar, unabhängig davon von welchem Typ die Kopplungen sind.

Kopplungen in der Architektur von SIMIT

Jedes Simulationssystem besteht aus zwei Komponenten:

- Modellierungssystem
  Im Modellierungssystem erstellen Sie das Simulationsmodell und konfigurieren die Schnittstellen zu den Kopplungspartnern.
- Steuersystem
  Im Steuersystem führen Sie das Simulationsmodell aus. Das Simulationsmodell kommuniziert dazu mit den Kopplungspartnern.



Eine Kopplung stellt die Verbindung zwischen dem vom Steuersystem ausgeführten Simulationsmodell und dem Kopplungspartner her.

- Die Schnittstelle zum Simulationsmodell
  Diese Schnittstelle ist für alle Kopplungen gleich. Damit ist das Simulationsmodell unabhängig vom Typ der verwendeten Kopplung. Kopplungen können somit ohne Änderungen im Simulationsmodell gegeneinander ausgetauscht werden.
- Die Schnittstelle zum Kopplungspartner
  Die Schnittstelle muss für jede Kopplung einmalig konfiguriert werden. Die Schnittstelle ist für jede Kopplung verschieden.

Konfiguration von Kopplungen

Um den Signalaustausch zwischen dem Simulationsmodell und einem Automatisierungssystem herzustellen, müssen in der Kopplung folgende Informationen enthalten sein:

- Informationen, die für den Verbindungsaufbau notwendig sind
- Signale, die auszutauschen sind

Diese Informationen werden für jede Kopplung über einen Konfigurator angelegt.

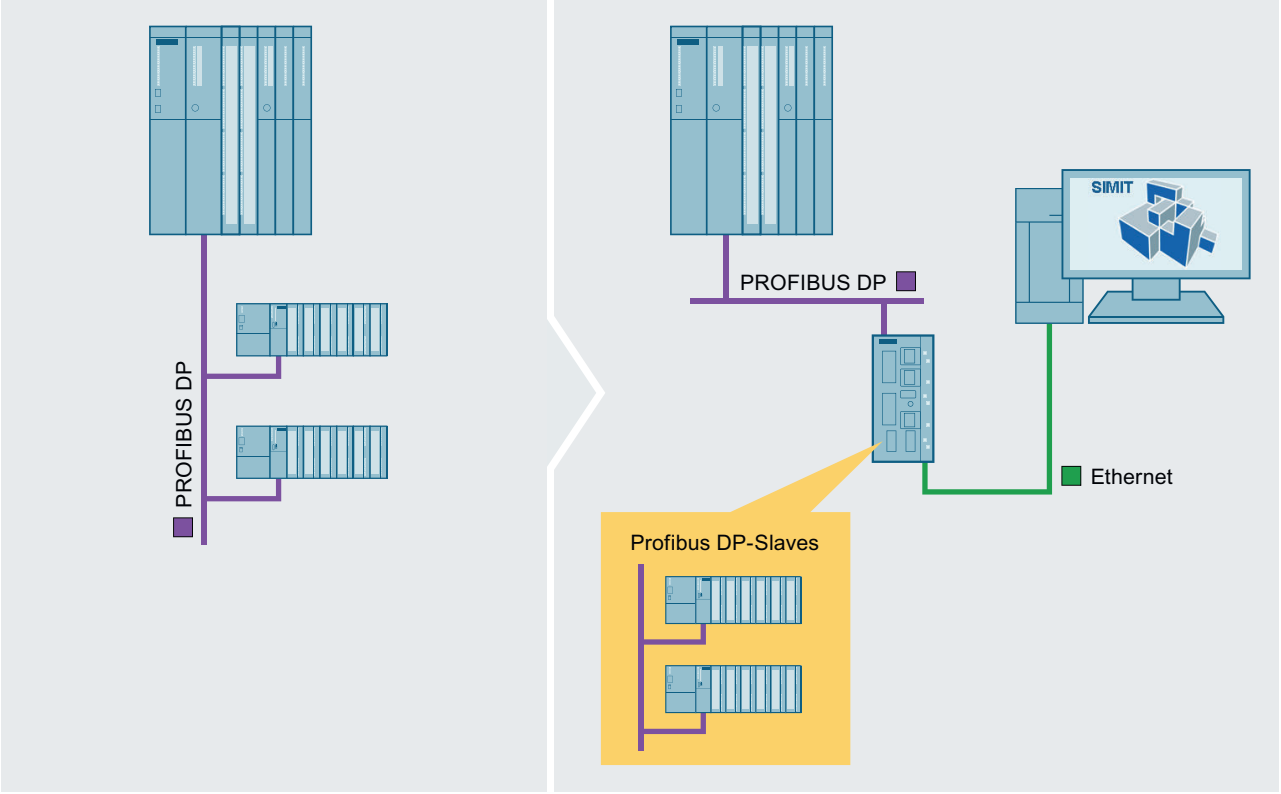
2.1.2 Kopplungen mit SIMATIC-Steuerungen

Einleitung

SIMIT stellt Kopplungen mit einer SIMATIC-Steuerung wie z. B. einer S7-400 über PROFIBUS DP, PROFINET IO oder PRODAVE sowie eine Kopplung zu PLCSIM / PLCSIM Advanced und zu SIMIT VC zur Verfügung. Die zur Konfiguration erforderlichen Daten können aus den SIMATIC-Projekten übernommen werden.

Kopplung zu einem SIMATIC-Controller

Eine SIMATIC-Steuerung mit einem Feldbus PROFIBUS DP oder PROFINET IO koppeln Sie direkt auf Feldbusebene mit SIMIT:



In SIMIT werden dazu "SIMIT Units" eingesetzt, welche die Feldgeräte simulieren. Eine solche SIMIT Unit wird in der einfachsten Konfiguration mit dem PROFIBUS DP-Master bzw. PROFINET IO-Controller der Steuerung verbunden. Die Steuerung kommuniziert dann mit dieser SIMIT Unit und mit SIMIT auf dieselbe Weise wie in einer realen Anlage mit den Feldgeräten. Die Schnittstelle von SIMIT zur SIMATIC-Steuerung ist in diesem Fall das Feldbuskabel.

Weitere Informationen hierzu finden Sie unter SIMIT Unit (Seite 67).

Kopplung mit Virtual Controller

Ein Virtual Controller simuliert eine Station mit Prozessabbild der Ein- und Ausgänge. Weitere Informationen hierzu finden Sie unter Virtual Controller (Seite 109).

Kopplung mit PLCSIM Advanced

Die PLCSIM Advanced-Kopplung von SIMIT ermöglicht einen dynamischen Datenaustausch zwischen PLCSIM Advanced und SIMIT. Dafür müssen PLCSIM Advanced und SIMIT auf demselben Rechner installiert sein. Weitere Informationen hierzu finden Sie unter PLCSIM Advanced-Kopplung (Seite 145).

Kopplung mit PLCSIM

Die PLCSIM-Kopplung von SIMIT ermöglicht einen dynamischen Datenaustausch zwischen PLCSIM und SIMIT. Dafür müssen PLCSIM und SIMIT auf demselben Rechner installiert sein. Weitere Informationen hierzu finden Sie unter PLCSIM-Kopplung (Seite 150).

Kopplung mit PRODAVE

Eine Alternative zur Kopplung von SIMIT an eine SIMATIC-Steuerung über Feldbus ist die Kopplung über die PRODAVE-Schnittstelle. Die physikalische Verbindung zwischen der Steuerung und dem SIMIT-PC erfolgt dabei über eine der folgenden Schnittstellen:

- Serielle Schnittstelle
- USB
- MPI-Schnittstellenkarte
- Ethernet

Durch die PRODAVE-Schnittstelle bedingt ist diese Kopplung weniger performant als die Feldbuskopplung.

Weitere Informationen hierzu finden Sie unter PRODAVE-Kopplung (Seite 177).

Hinweis

IP-Adressraum

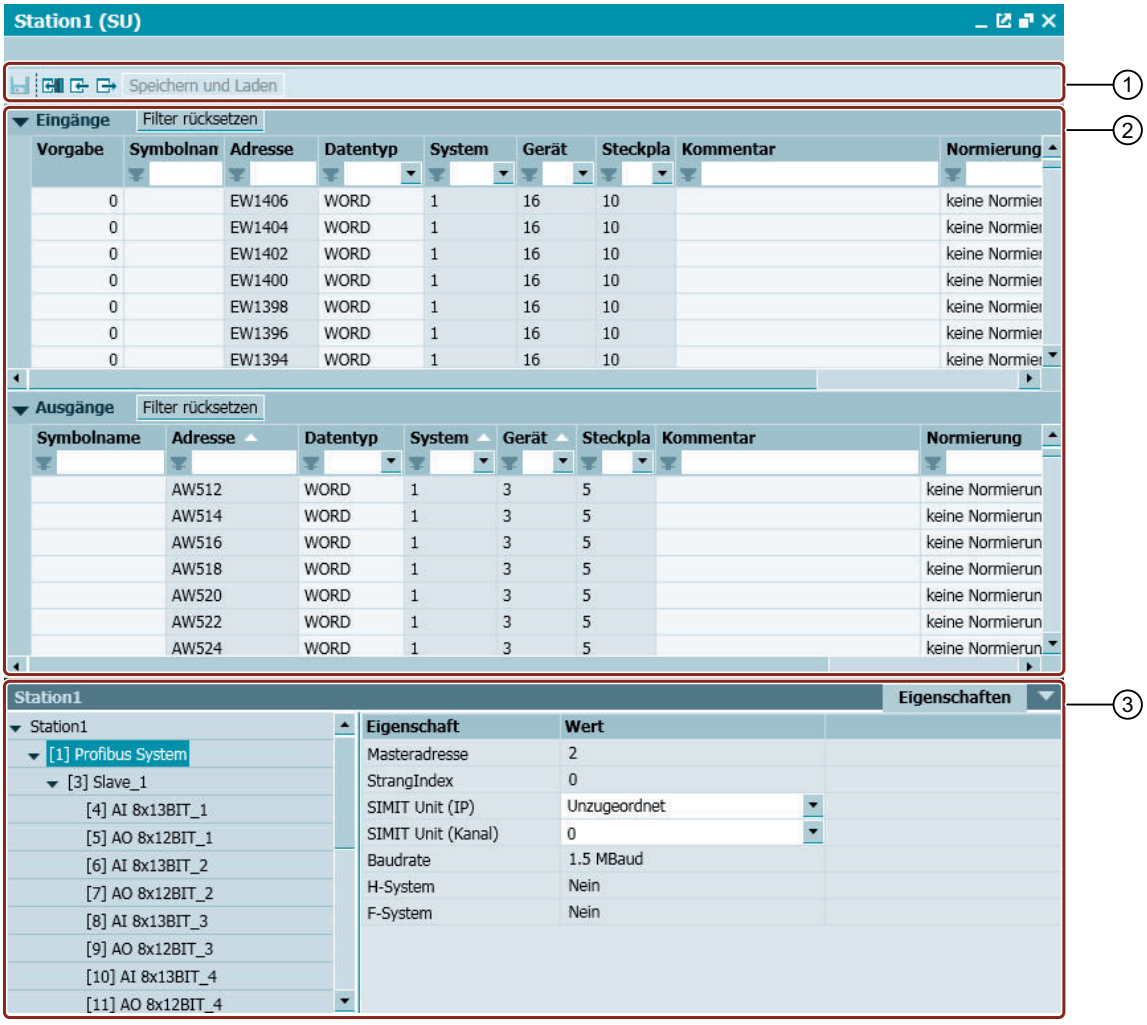
SIMIT unterstützt nur PCS 7-Projekte mit einem IP-V4-Adressraum.

2.2 Kopplungseditor

Im Kopplungseditor führen Sie folgende Aufgaben aus:

- Eigenschaften einer Kopplung konfigurieren
- Eigenschaften von Signalen konfigurieren

Der Kopplungseditor ist für jeden Kopplungstyp spezifisch, besitzt aber prinzipiell immer den in der folgenden Abbildung gezeigten Aufbau:



① Menüzeile

Der Inhalt der Menüzeile ist abhängig vom Kopplungstyp. Standardmäßig enthält die Menüzeile die Befehle zum Importieren und Exportieren von Signalen. Weitere Informationen finden Sie unter Import und Export der Signale (Seite 199)

② Eingangs- und Ausgangssignale

Listet die Signale mit ihren zugehörigen Parametern und Eigenschaften getrennt nach "Eingängen" und "Ausgängen" auf:

- Signale unter "Eingänge" verbinden Sie mit Ausgangskonnektoren im Simulationsmodell
- Signale unter "Ausgänge" verbinden Sie mit Eingangskonnektoren im Simulationsmodell

Informationen zum Bearbeiten der Signale im Kopplungseditor finden Sie im Kapitel: Grundlagen zu Signalen (Seite 188).

③ Eigenschaftsfenster

Im Eigenschaftsfenster bearbeiten Sie die Eigenschaften eines Signals oder der in der Projektnavigation gewählten Kopplung.

Welche Eigenschaften angezeigt werden, hängt vom Kopplungstyp ab.

Den Kopplungseditor öffnen Sie in der Projektnavigation, indem Sie auf die entsprechende Kopplung doppelklicken.

2.3 HWCNExport-Datei erstellen

Einleitung

Um auf einem SIMIT-PC ohne PCS 7-Installation dennoch Stationen aus einem STEP 7-/PCS 7-Projekt zu importieren, erstellen Sie aus einem STEP 7-/PCS 7-Projekt eine HWCNExport-Datei. Zum Erstellen der HWCNExport-Datei verwenden Sie die Anwendung "HWCNExport", die standardmäßig mit SIMIT installiert wird.

Hinweis

Die Anwendung "HWCNExport" ist ohne zusätzliche Installation lauffähig.

Sie finden die Anwendung unter dem Startmenü "Start > Alle Programme >Siemens Automation > SIMIT > Tools".

Wenn Sie diese Anwendung auf einem PC ohne SIMIT-Installation verwenden wollen, kopieren Sie den kompletten Ordner auf einen beliebigen Ort des PCs und starten Sie von dort die "HWCNExport".

Folgende Kopplungen unterstützen den Import aus einer HWCNExport-Datei:

Kopplung	Unterstützte STEP 7-Version
Virtual Controller	• Bis einschließlich V5.5
PLCSIM Advanced	• V14 SP1 bis V15.1

Voraussetzung

- Erstellen einer HWCNExport-Datei für einen Virtual Controller:
  - STEP 7 bis V5.5 ist installiert.
- Erstellen einer HWCNExport-Datei für PLCSIM Advanced:
  - STEP 7 (V14 SP1 bis V15.1) ist installiert.
  - TIA Portal Openness ist installiert.
  - STEP 7-Projekt ist geschlossen.
- .NET-Framework 3.5 oder höher ist installiert.
- Ordner mit Anwendung "HWCNExport" ist vorhanden.

Vorgehen

Um eine HWCNExport-Datei zu erstellen, gehen Sie folgendermaßen vor:

1. Starten Sie die Anwendung "HWCNExport".
2. Um die Hardware-Konfiguration aus einem Projekt mit Version STEP 7 V5.5 zu exportieren:
   - Wählen Sie die Registerkarte "STEP 7 V5.x".
   - Aktivieren Sie den gewünschten STEP 7-/PCS 7-Projekttyp.
   - Wählen Sie das STEP 7-/PCS 7-Projekt.

3. Um die Hardware-Konfiguration aus einem Projekt mit Version STEP 7 V14 SP1 zu exportieren:
   - Wählen Sie die Registerkarte "STEP 7 V14 SP1".
   - Wählen Sie das STEP 7-Projekt.
4. Wählen Sie den Ordner, in dem die HWCNExport-Datei abgelegt wird.
5. Klicken Sie auf "Create XML".

Ergebnis

Die HWCNExport-Datei im Format "*.XML" wird erstellt.

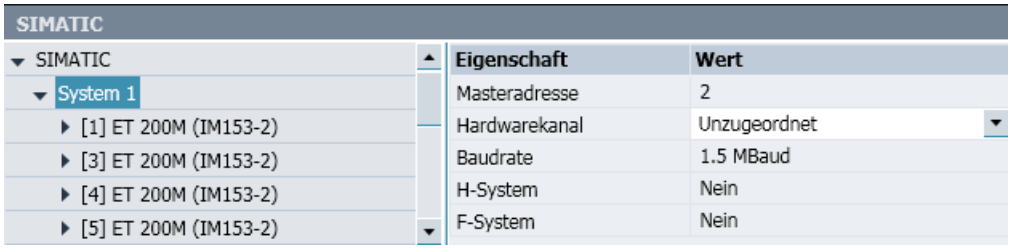
Siehe auch

Dialogfeld "Virtual Controller Import" (Seite 964)

Dialogfeld "PLCSIM Advanced-Import" (Seite 962)

2.4 Kopplungen deaktivieren

Wenn Sie eine Kopplung in Ihrem Simulationsprojekt konfiguriert haben, die Simulation aber ohne Kopplung starten möchten, stellen Sie den Hardwarekanal im Eigenschaftsfenster der Kopplung auf "Unzugeordnet":



Beim nächsten Starten der Simulation wird die Kopplung dann nicht aktiviert. Alle Werte der in dieser Kopplung enthaltenen Signale werden auf "0" gesetzt. Ein Signalaustausch findet nicht statt.

Diese Funktion ist nur bei den folgenden Kopplungstypen verfügbar:

- SIMIT Unit
- PLCSIM
- PRODAVE
- OPC Client

2.5 SIMIT Unit

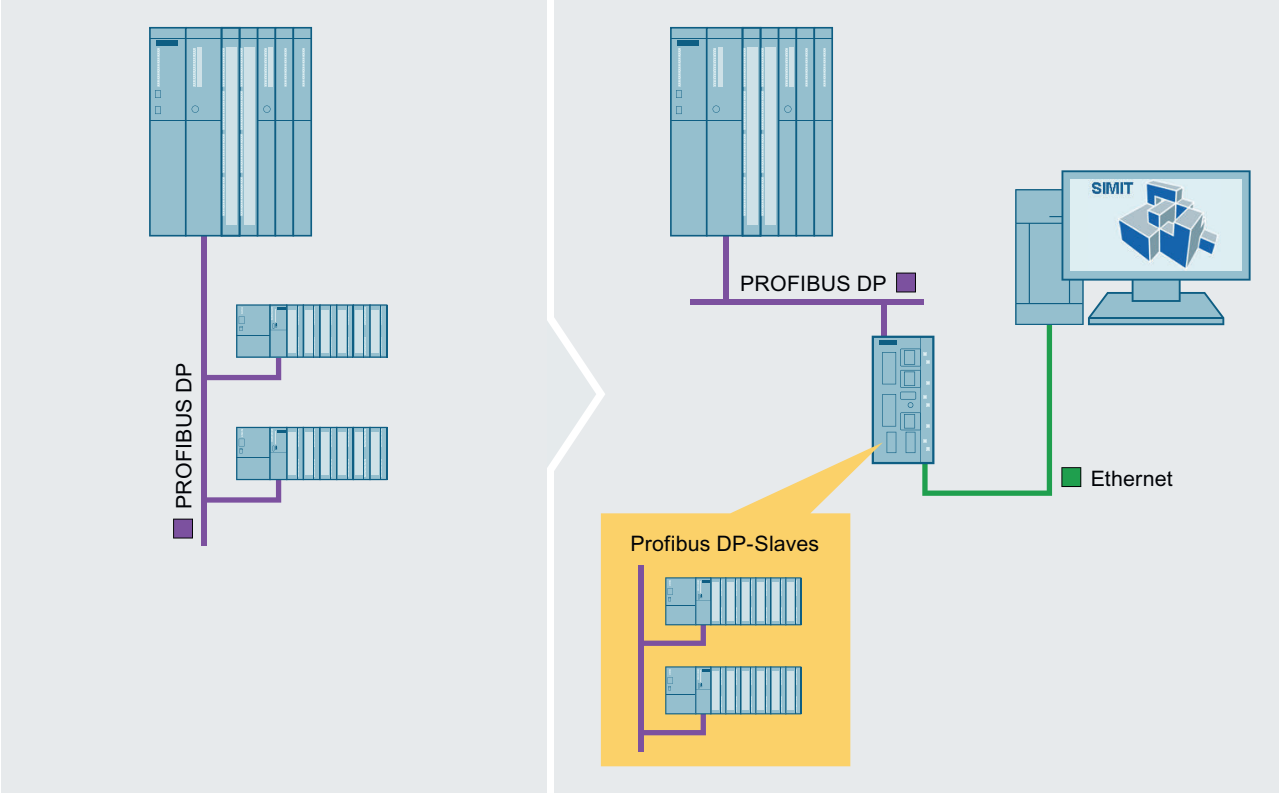
2.5.1 Funktionsweise der SIMIT Unit-Kopplung

Funktion

Mit der Kopplung "SIMIT Unit" kommuniziert SIMIT mit einem oder mehreren Bussystemen:

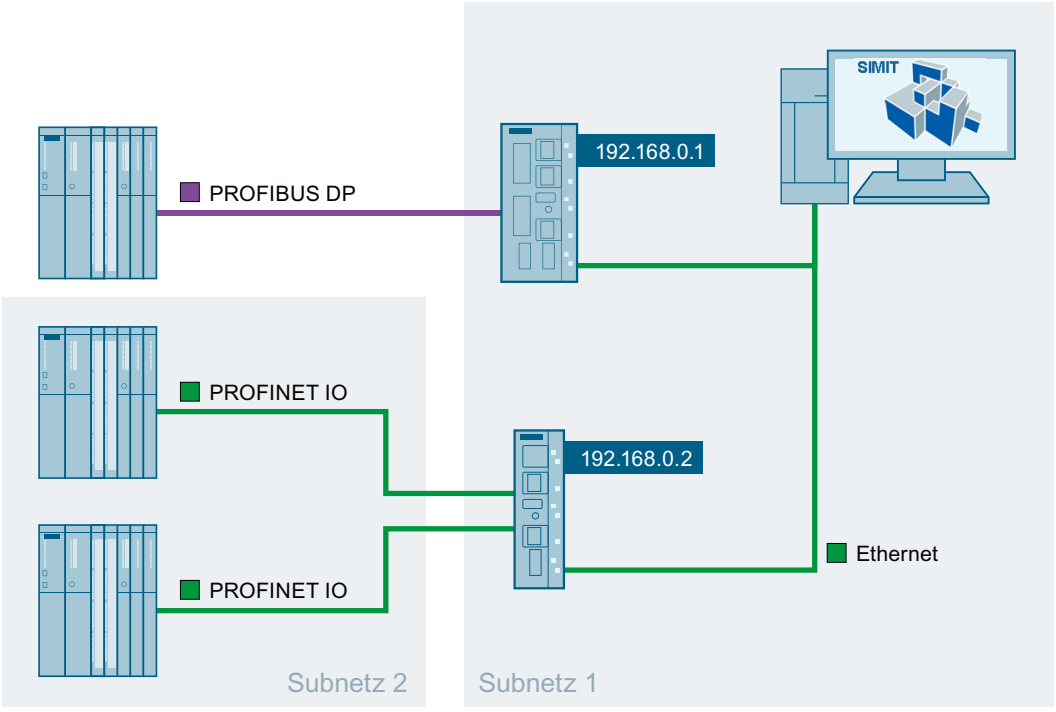
- PROFINET IO-Controller
- PROFIBUS DP-Master

Für die Kommunikation zwischen Controller und SIMIT wird die "SIMIT Unit" als Bindeglied eingesetzt. Die SIMIT Unit bildet dazu das Verhalten von Geräten am Bus nach und erlaubt den Datenaustausch zwischen Controller und SIMIT.



Anbindung der SIMIT Unit an SIMIT

Die SIMIT Unit kommuniziert mit einem SIMIT-PC über Ethernet. SIMIT-PC und alle verbundenen SIMIT Units müssen in einem gemeinsamen Subnetz liegen. Die am PROFINET simulierten Geräte müssen in einem anderen Subnetz liegen. Weitere Informationen dazu finden Sie im Kapitel: Unterstützte PROFINET IO-Konfigurationen (Seite 76)



Weiterführende Informationen

Weiterführende Informationen zur SIMIT Unit finden Sie im Internet (https://support.industry.siemens.com/cs/ww/de/view/109746192).

Siehe auch

SIMIT Unit konfigurieren (Seite 87)

Unterstützte PROFIBUS DP-Konfigurationen (Seite 69)

2.5.2 Unterstützte PROFIBUS DP-Konfigurationen

Einleitung

Im Folgenden werden typische PROFIBUS DP-Konfigurationen aufgeführt, die von SIMIT unterstützt werden. Links in der jeweiligen Abbildung (s. unten) sehen Sie die Konfiguration des Automatisierungssystems und rechts die Anordnung mit SIMIT. Sie finden folgende Beispiele für Konfigurationen mit nicht-redundanten Steuerungen:

- Eine Steuerung mit einem PROFIBUS DP-Strang
- Eine Steuerung mit zwei PROFIBUS DP-Strängen
- Zwei Steuerungen mit jeweils einem PROFIBUS DP-Strang

Für redundante Konfigurationen sind folgende Beispiele gegeben:

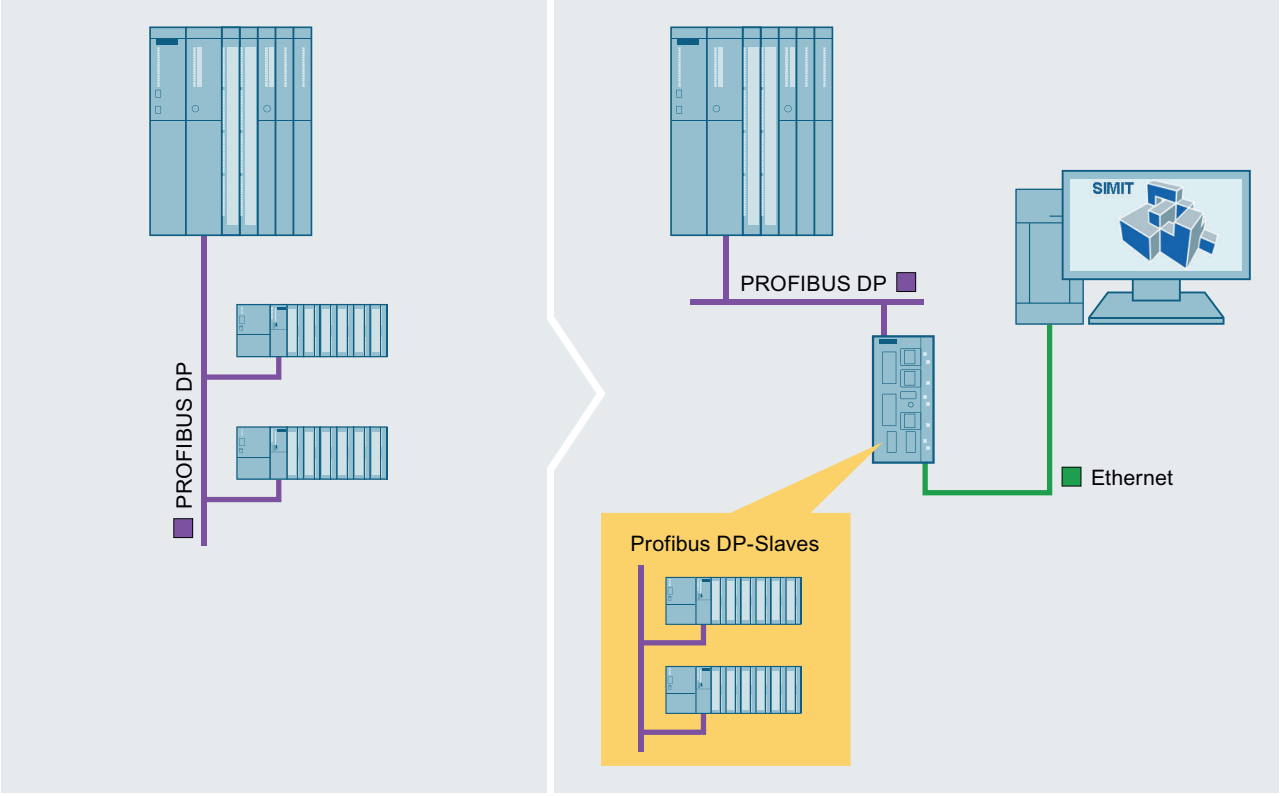
- Eine redundante Steuerung mit einem redundanten PROFIBUS DP-Strang
- Eine redundante Steuerung mit Y-Link
- Zwei redundante Steuerungen mit nur einseitig projektiertem PROFIBUS DP-Strang

Bei allen Konfigurationen ist auch eine Mischkonfiguration von realen und simulierten PROFIBUS DP-Slaves möglich.

Ebenso können Buskoppler eingesetzt werden, entweder als DP/DP-Koppler, als DP/PA-Koppler oder in redundanten Konfigurationen auch als Y-Link.

Fehlersichere Peripherie kann ebenfalls simuliert werden, auch in redundanten HF-Systemen.

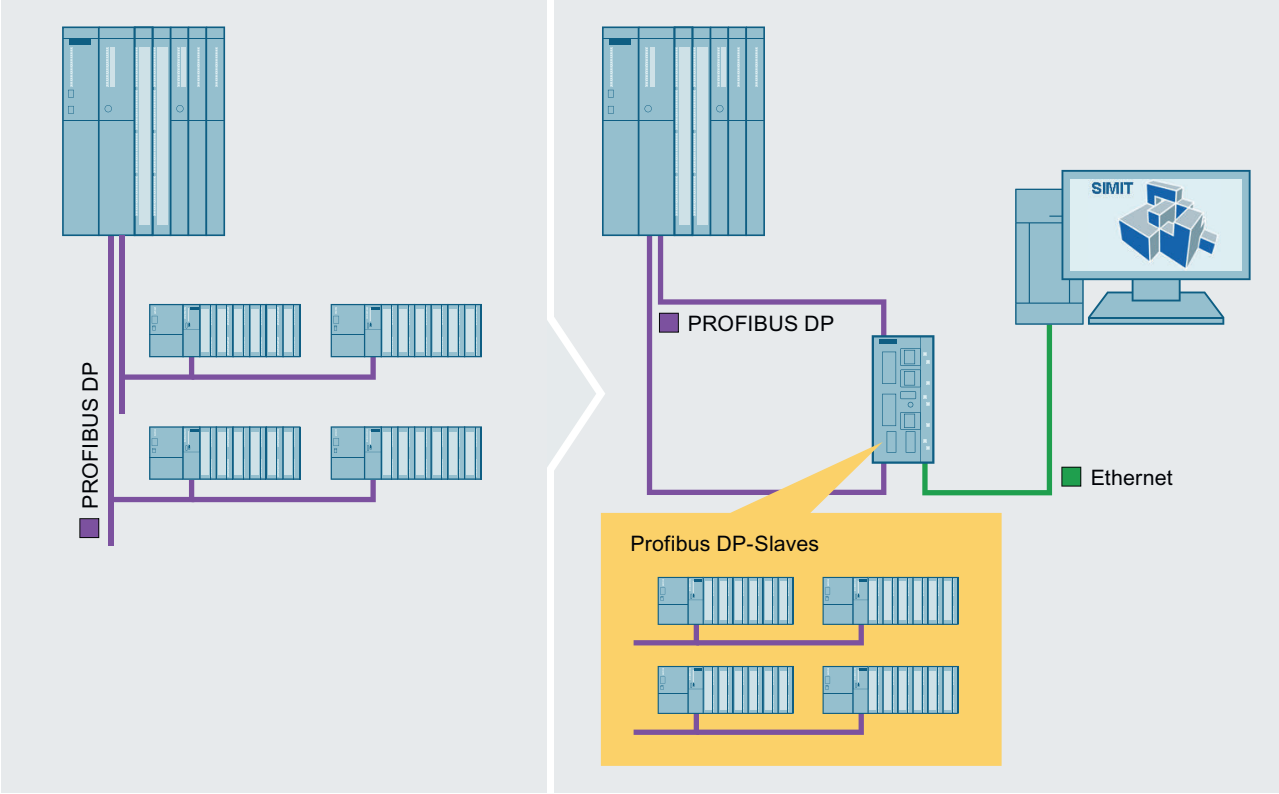
Eine Steuerung mit einem PROFIBUS DP-Strang



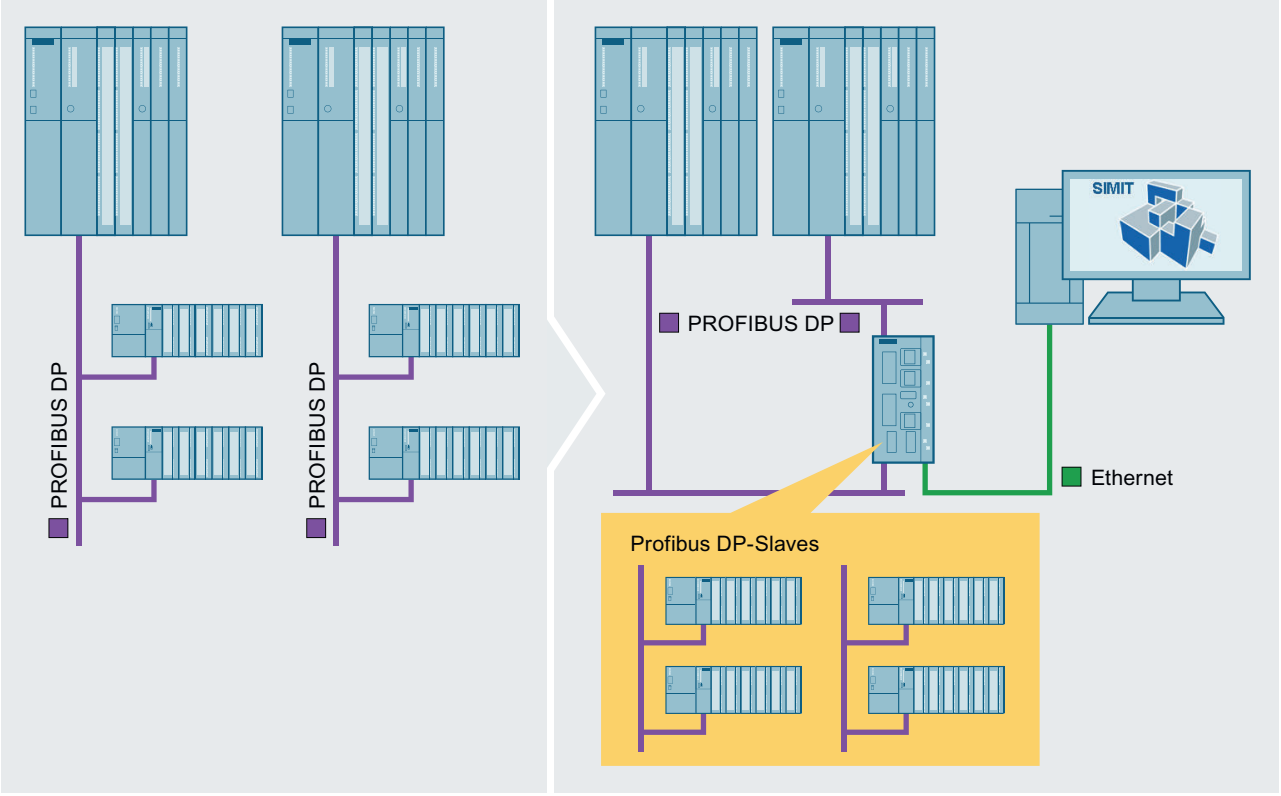
Hinweis

In dieser Konfiguration wird beim Einsatz einer zweikanaligen SIMIT Unit nur ein Kanal benutzt.

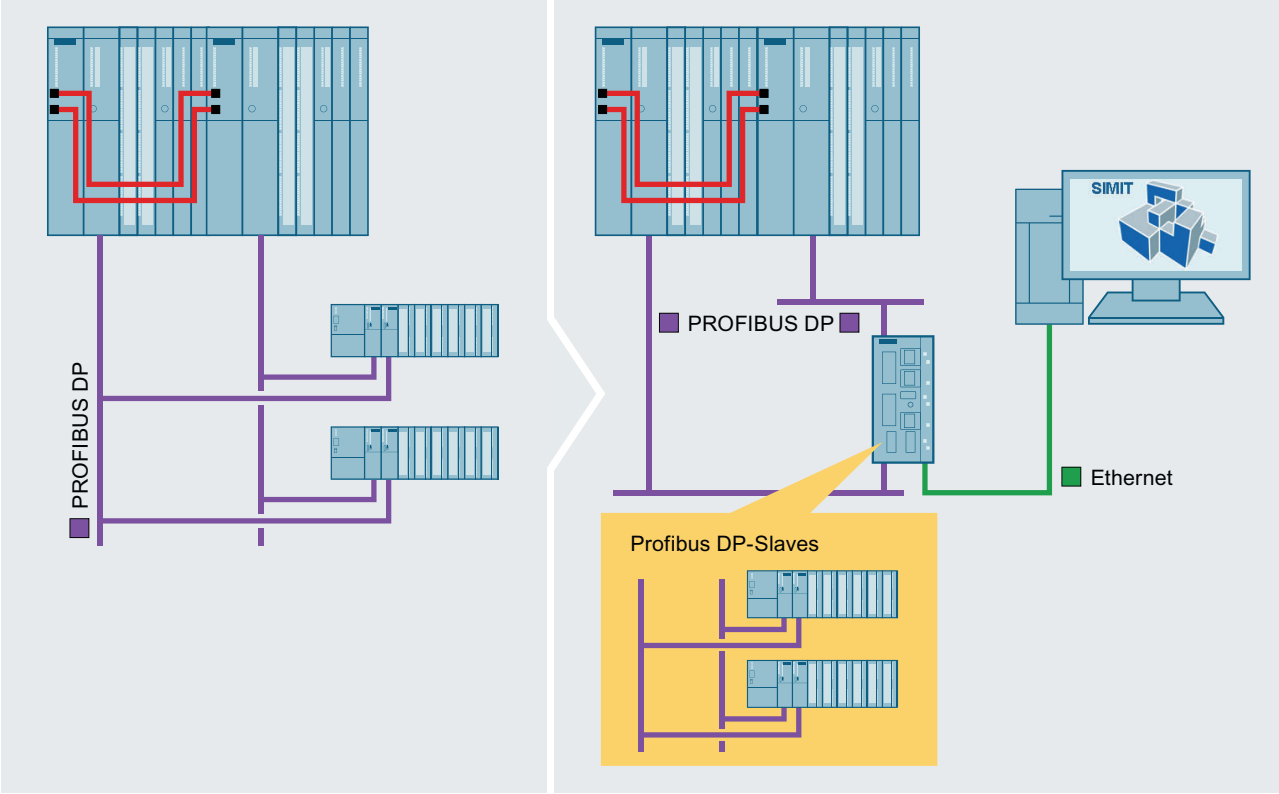
Eine Steuerung mit zwei PROFIBUS DP-Strängen



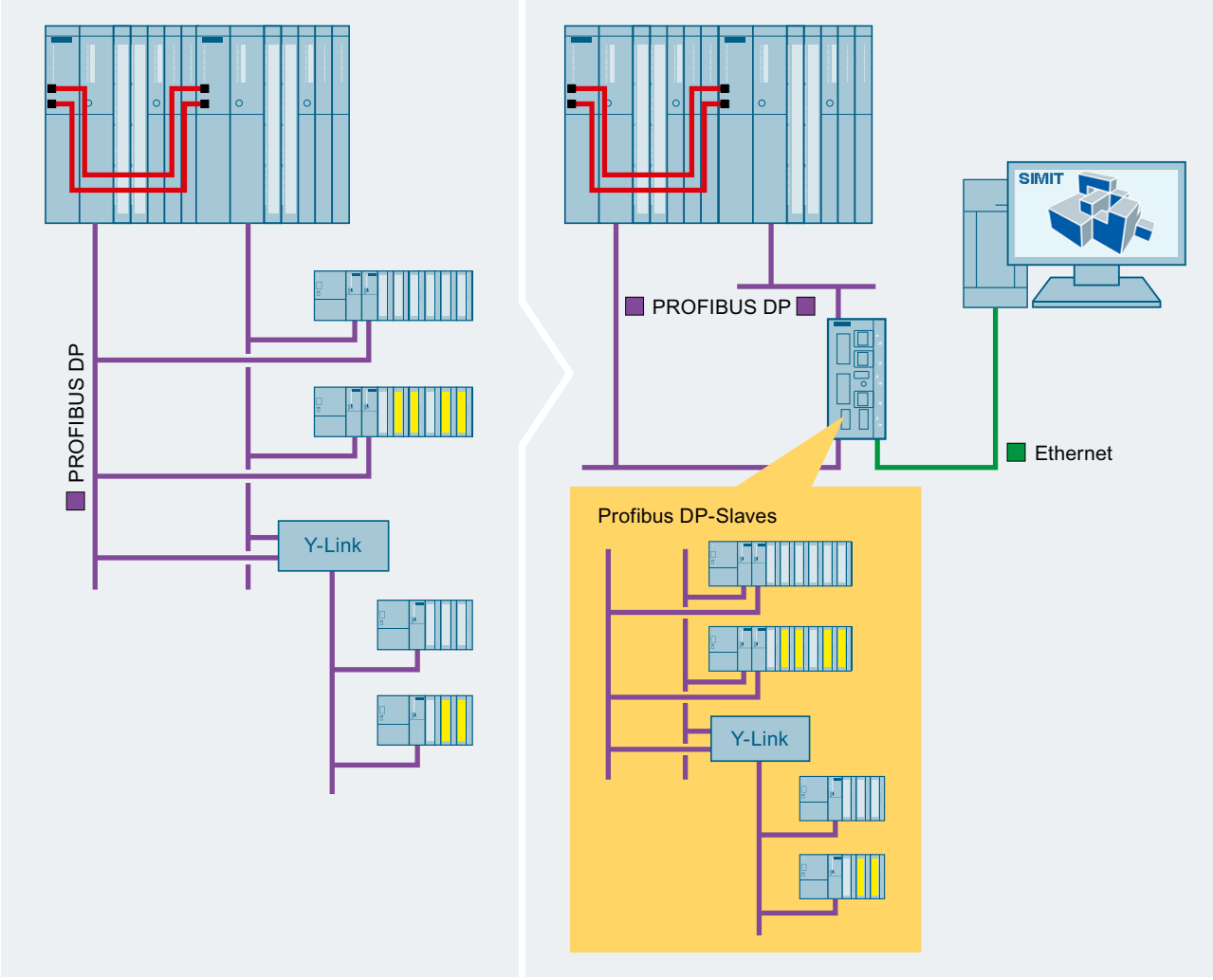
Zwei Steuerungen mit jeweils einem PROFIBUS DP-Strang



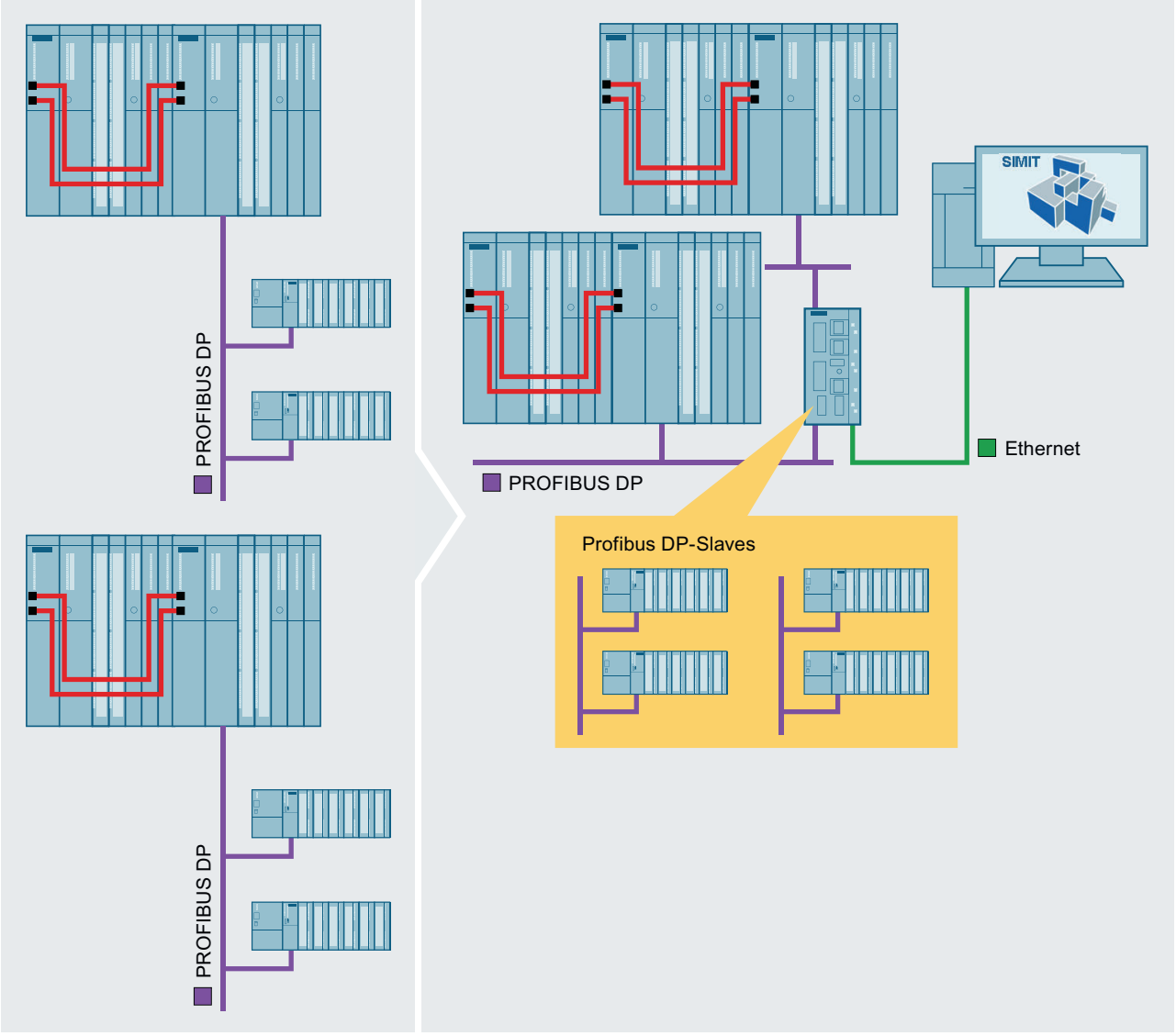
Eine redundante Steuerung mit einem redundanten PROFIBUS DP-Strang



Eine redundante Steuerung mit Y-Link



Zwei redundante Steuerungen mit nur einseitig projektiertem PROFIBUS DP-Strang



Hinweis

Beachten Sie, dass die Station in STEP 7 tatsächlich einseitig projektiert sein muss.

Siehe auch

Redundante Konfigurationen (H-Systeme) (Seite 81)

2.5.3 Unterstützte PROFINET IO-Konfigurationen

Einleitung

Im Folgenden werden typische PROFINET IO-Konfigurationen aufgeführt, die von SIMIT unterstützt werden. Links in der jeweiligen Abbildung (s. unten) sehen Sie die Konfiguration des Automatisierungssystems und rechts die Anordnung mit SIMIT.

- Eine Steuerung mit einem PROFINET IO-Strang
- Eine redundante Steuerung mit einem redundanten PROFINET IO-Strang
- Eine Steuerung mit Shared Device
- Ring-Topologie

Hinweis

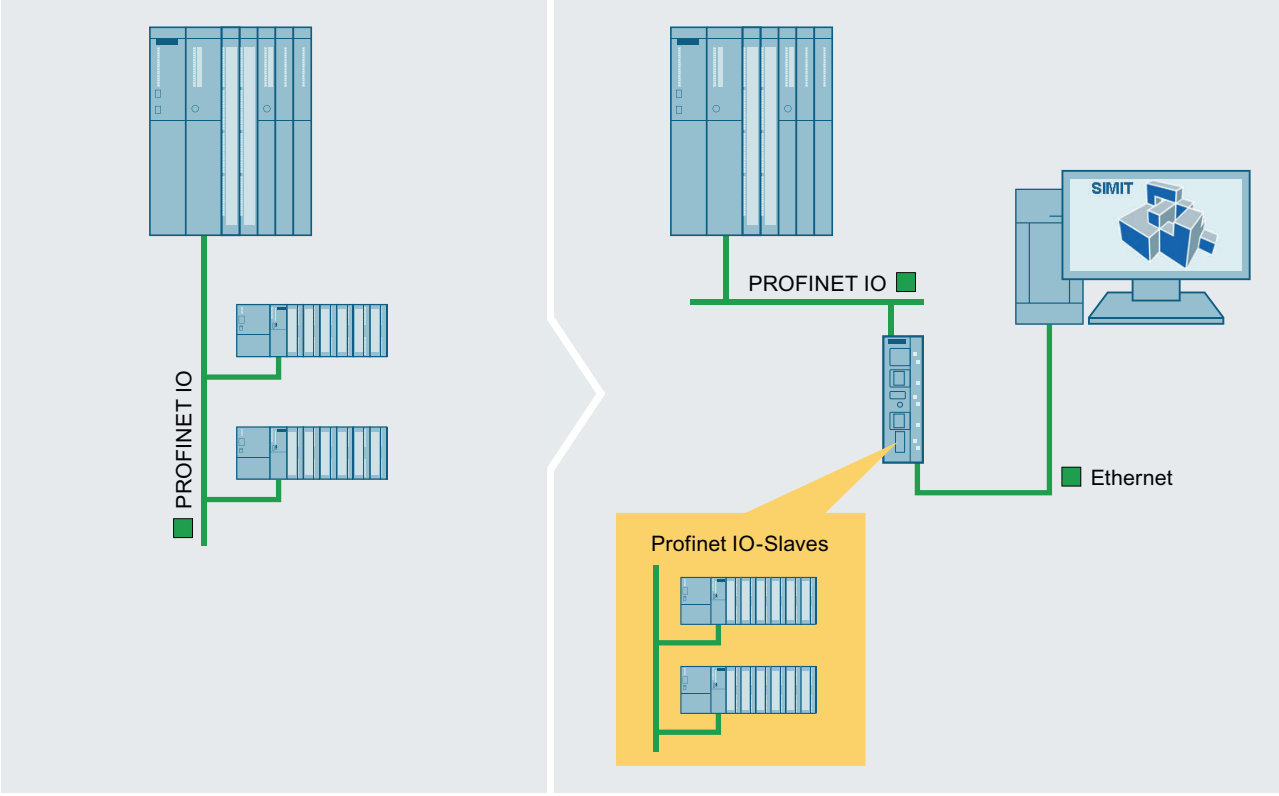
Die IP-Adresse der SIMIT Unit und die IP-Adressen der simulierten PROFINET IO-Devices müssen in unterschiedlichen Subnetzen liegen.

Beispiel:

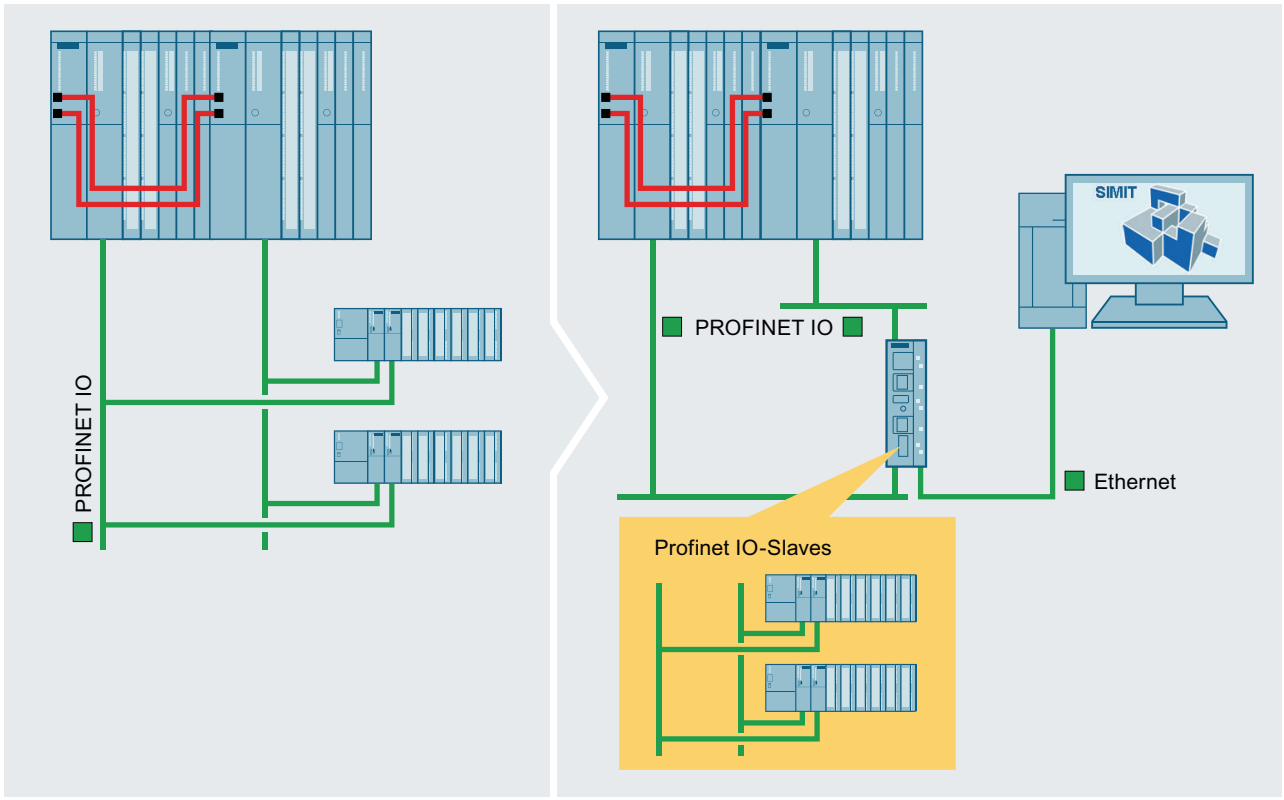
- IP-Adresse der SIMIT Unit: 192.16**9**.xxx.yyy, Subnetz-Maske: 255.255.0.0
- IP-Adresse eines simulierten PROFINET IO-Device: 192.16**8**.xxx.yyy, Subnetz-Maske: 255.255.0.0

Mindestens eine Ziffer in einem der ersten beiden Segmente der IP-Adresse muss unterschiedlich sein.

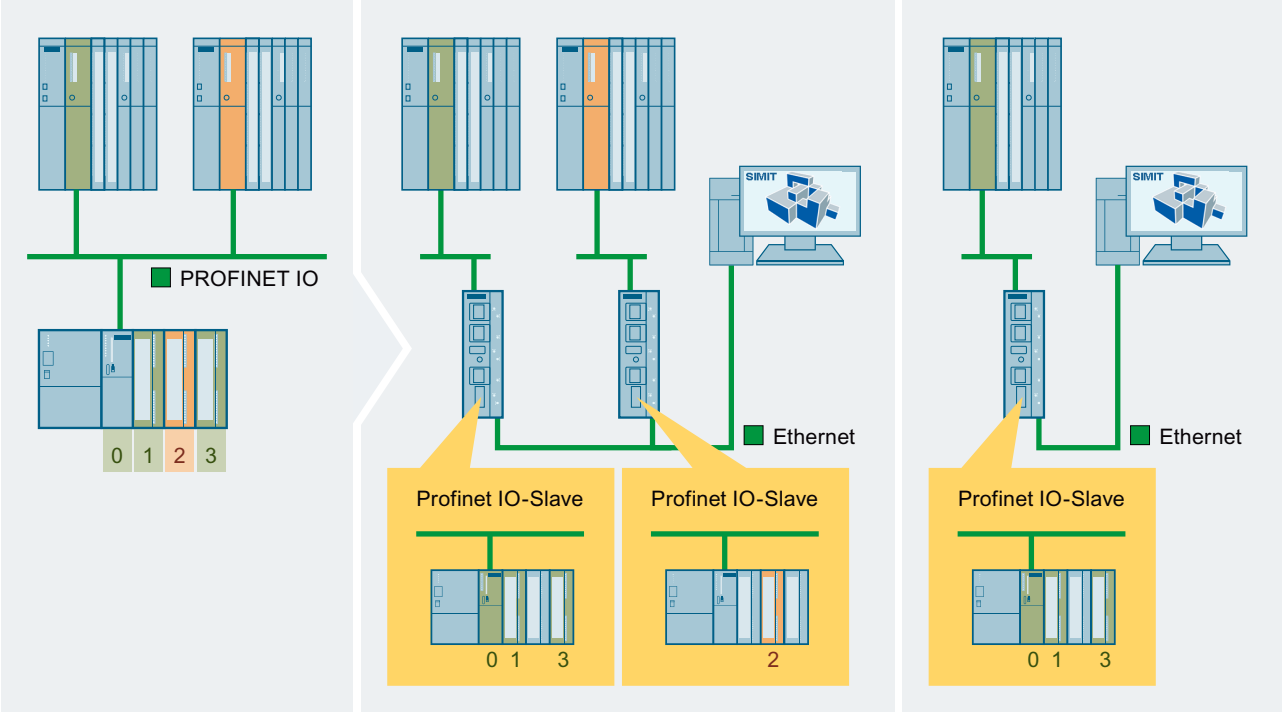
Eine Steuerung mit einem PROFINET IO-Strang



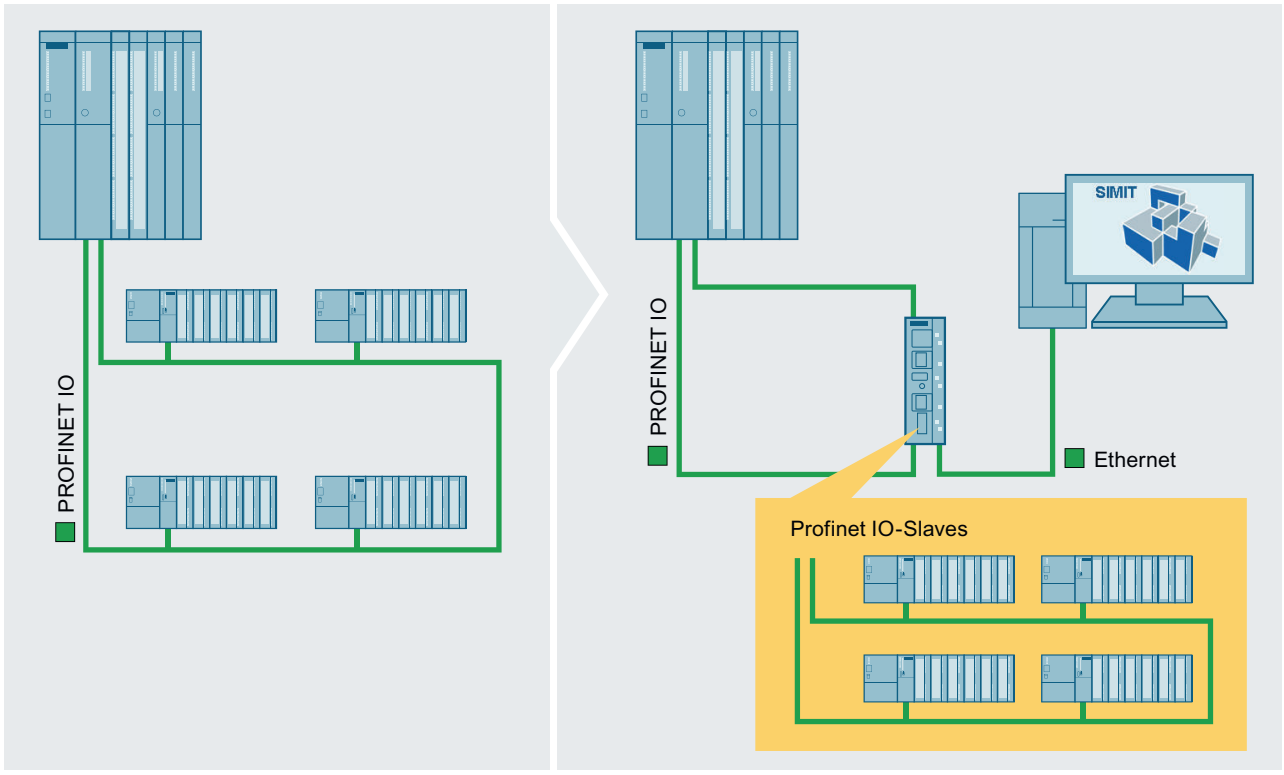
Eine redundante Steuerung mit einem redundanten PROFINET IO-Strang



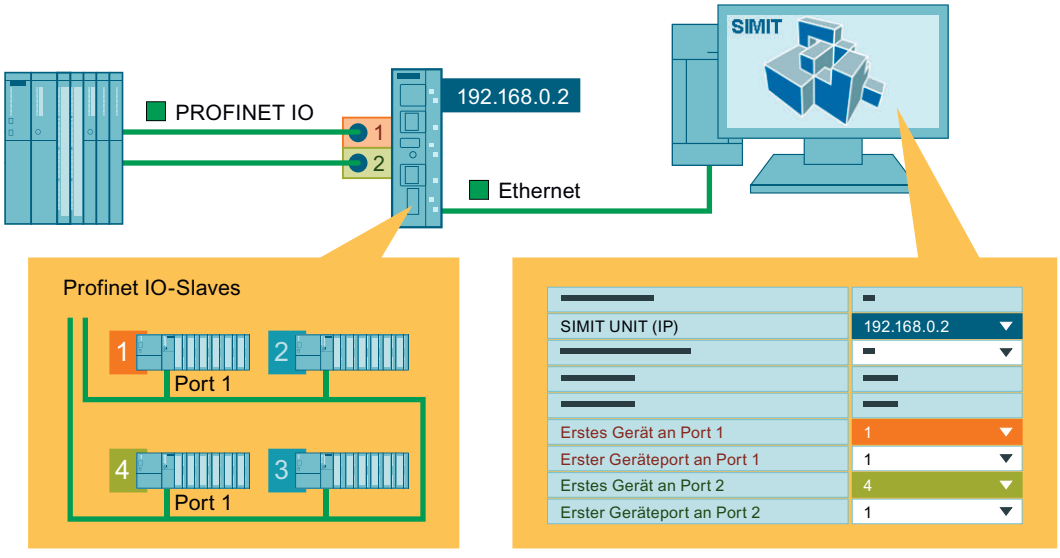
Steuerungen mit Shared Device



Ring-Topologie



Die folgende Abbildung zeigt schematisch die notwendigen Konfigurationseinstellungen in SIMIT:



Siehe auch

Eigenschaften der SIMIT Unit-Kopplung (Seite 100)

Redundante und fehlersichere Systeme (Seite 81)

2.5.4 Redundante und fehlersichere Systeme

2.5.4.1 Redundante Konfigurationen (H-Systeme)

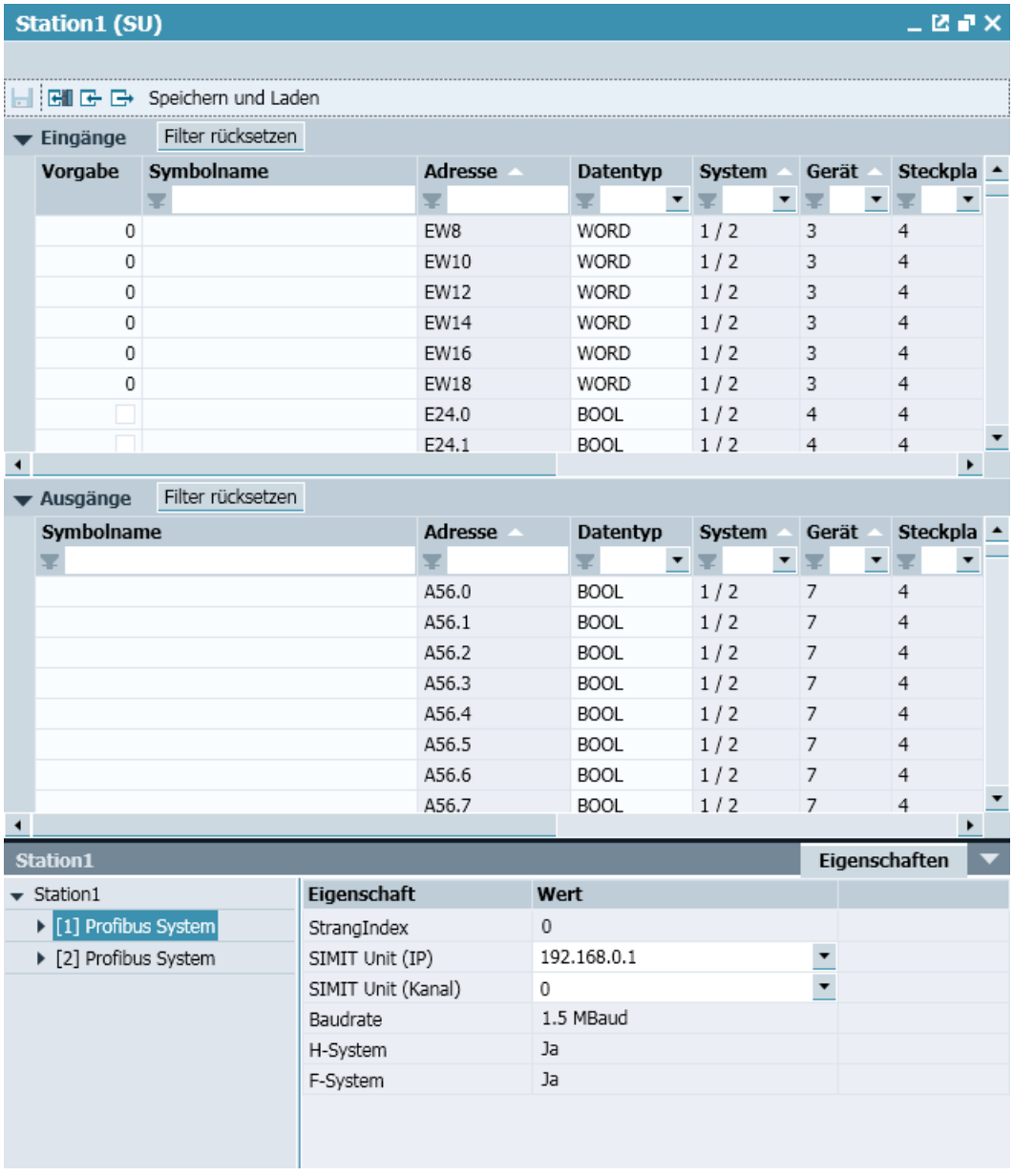
Einleitung

SIMIT unterstützt die Simulation von redundant ausgelegten Systemen (H-Systeme). In einem redundant angeschlossenen System werden alle erforderlichen redundanten Diagnosesignale von der SIMIT Unit erzeugt, sodass Sie auch Master/Standby-Umschaltungen an Ihrer Steuerung vornehmen können. Im Eigenschaftsfenster finden Sie unter "H-System" eines Slaves die Information, ob der Slave redundant oder nicht redundant ist.

In redundanten Systemen können auch DP/DP-Koppler als Y-Link enthalten sein. Die Kopplungssignale der Busteilnehmer, die hinter dem Y-Link angeschlossen sind, stehen in SIMIT dann genauso zur Verfügung wie die Kopplungssignale der direkt am redundanten Bus angeschlossenen Teilnehmer.

Redundantes PROFIBUS-System

In SIMIT werden nur die Signale des ersten redundanten Systems (Kanal 0) als Kopplungssignale verwendet, unabhängig davon welcher der beiden Busse aktiv ist. Busumschaltungen werden dazu von SIMIT verwaltet: Die Zuordnung der Kopplungssignale zum aktiven Kanal erfolgt automatisch. Die redundanten Signale des zweiten Kanals sind nicht aufgeführt, Sie erkennen aber in der Signalliste, dass diese Signale zwei Mastersystemen (1 und 2) zugeordnet sind.



Redundantes PROFINET-System

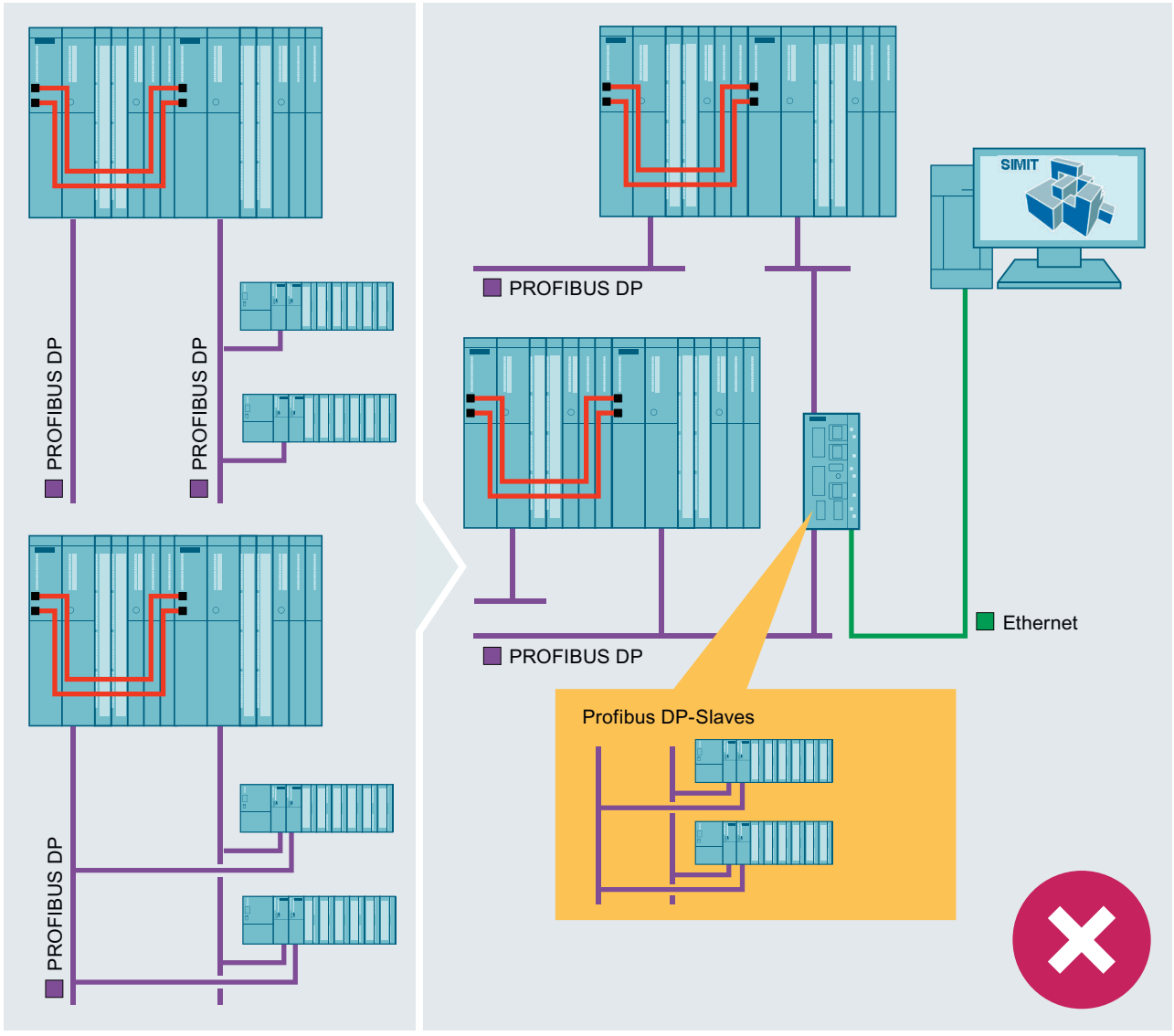
Hinweis

Die Simulation eines redundanten PROFINET-Systems wird nur mit SIMIT Unit PN 256 unterstützt.

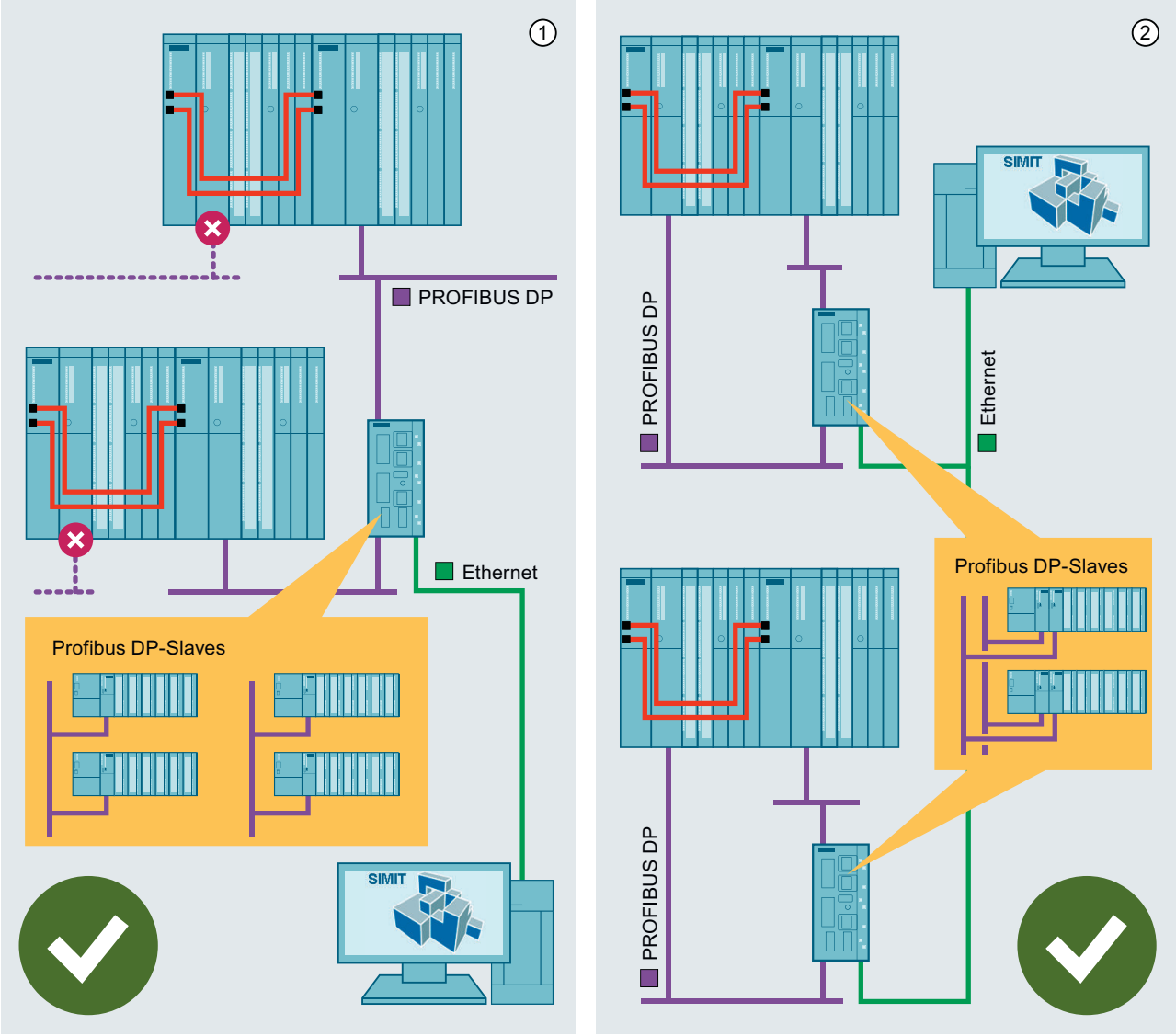
In SIMIT wird nur ein Strang des redundanten Systems angelegt.

Konfigurationsbeispiel

Wenn Sie eine Kopplung zu einem redundanten System projektieren, ist pro redundantem System eine SIMIT Unit notwendig. In folgendem Beispiel sind zwei redundante Systeme projektiert. An jedem redundanten System sind zwei Stränge projektiert, aber nur an je einem Strang ist dezentrale Peripherie angeschlossen. Die folgende Konfiguration mit nur einer SIMIT Unit ist nicht zulässig:



Die folgende Abbildung zeigt die beiden Lösungsmöglichkeiten:

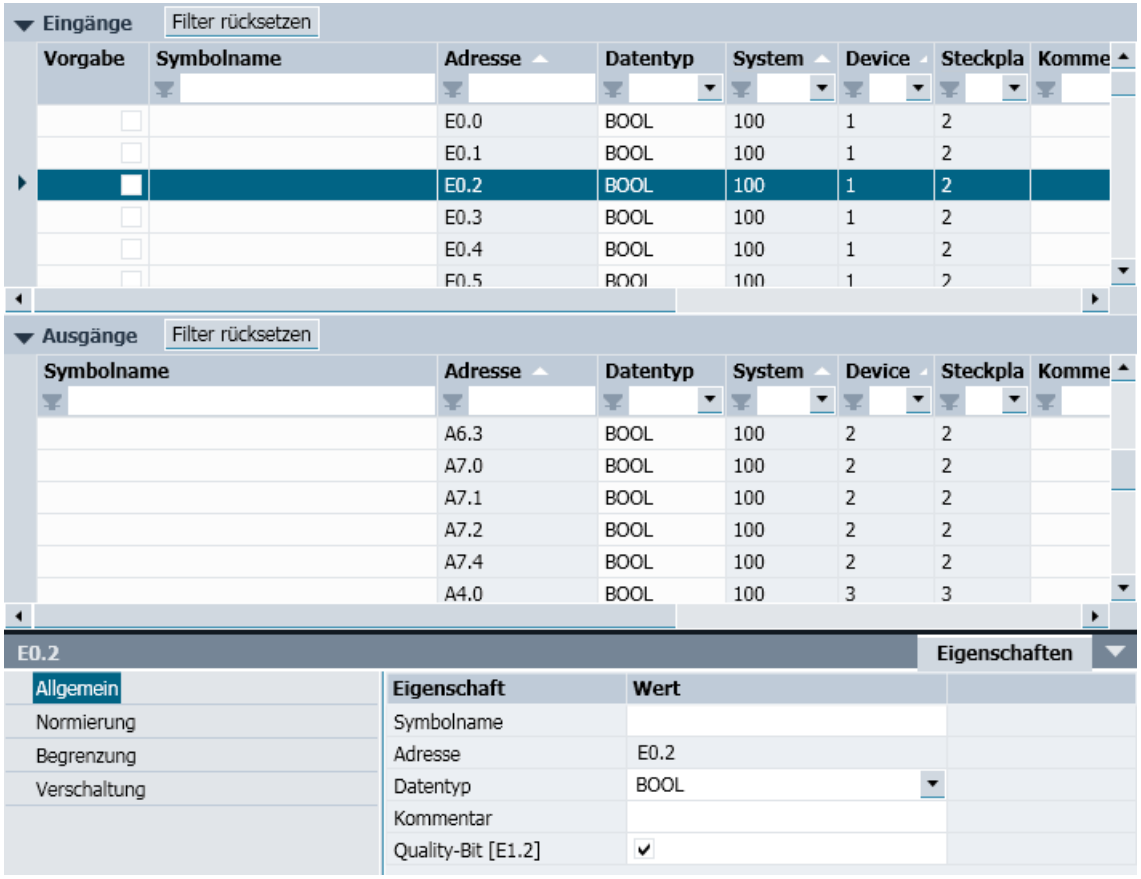


① Sie entfernen in der Hardware-Konfiguration den ungenutzten Strang.

② Sie setzen eine zusätzliche SIMIT Unit ein.

2.5.4.2 Fehlersichere Konfigurationen (F-Systeme)

Beim Import einer fehlersicheren Konfiguration werden automatisch für alle binären fehlersicheren Signale auch die zugehörigen Qualitätssignale angelegt und im Eigenschaftsfenster des fehlersicheren Signals entsprechend angezeigt.



Alle Qualitätssignale werden mit "1" als gültig vorbelegt. Um ein Signal auf "ungültig" zu setzen, setzen Sie das entsprechende Qualitätssignal auf "0" und deaktivieren Sie das Optionskästchen im Eigenschaftsfenster unter "Quality-Bit". Damit kann z. B. die Reaktion Ihres Steuerungsprogrammes auf eine Signalstörung getestet werden.

Hinweis

Nach dem Laden der Anschaltbaugruppe mit einer fehlersicheren Konfiguration muss an der angeschlossenen SIMATIC-CPU ein Neustart ausgeführt werden.

2.5.4.3 Redundante, fehlersichere Konfigurationen (HF-Systeme)

Redundante und fehlersichere Systeme (HF-Systeme) können ebenfalls simuliert werden. Die Signale und Qualitätssignale werden bearbeitet und angezeigt wie in den folgenden Abschnitten beschrieben:

- Redundante Konfigurationen (H-Systeme) (Seite 81)
- Fehlersichere Konfigurationen (F-Systeme) (Seite 86)

Folgende gemischte Konfigurationen können ebenfalls simuliert werden:

- Fehlersichere und nicht-fehlersichere Busteilnehmer
- Redundant und nicht-redundant angeschlossene Busteilnehmer

2.5.5 SIMIT Unit einrichten

2.5.5.1 SIMIT Unit konfigurieren

Voraussetzung

- SIMIT Unit ist im Netzwerk erreichbar.
- SIMIT ist geöffnet.

Vorgehen

Um eine SIMIT Unit zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie im Menü "Extras" den Befehl "SU-Verwaltung".
   Das Dialogfeld "SU-Verwaltung" wird geöffnet. Im Netzwerk wird automatisch nach erreichbaren SIMIT Units gesucht.
2. Geben Sie für die SIMIT Unit folgende Daten ein:
   - Name
   - IP-Adresse
   - Subnetz-Maske

Ergebnis

Die SIMIT Unit ist konfiguriert. Die Konfigurationsdaten werden in der SIMIT Unit gespeichert.

Siehe auch

Funktionsweise der SIMIT Unit-Kopplung (Seite 67)

Dialogfeld "SU-Verwaltung" (Seite 973)

Kopplung vom Typ "SIMIT Unit" anlegen (Seite 89)

2.5.5.2 Gerätebeschreibungsdatei in SIMIT importieren

Einleitung

Falls in der ausgelieferten SIMIT-Software bereits Daten zu einem Gerät existieren, werden diese in der "SU-Verwaltung" als "System" angezeigt. Falls Sie die GSD- bzw. GSDML-Dateien zu einem Gerät importieren, werden diese in der "SU-Verwaltung" als "User" angezeigt. Die "User"-Informationen haben gegenüber den "System"-Information generell den Vorrang.

Voraussetzung

- SIMIT ist geöffnet.
- Gerätebeschreibungsdatei liegt vor.

Vorgehen

Um eine Gerätebeschreibungsdatei zu importieren, gehen Sie folgendermaßen vor:

1. Wählen Sie im Menü "Extras" den Befehl "SU-Verwaltung".
   Das Dialogfeld "SU-Verwaltung" wird geöffnet. Im Netzwerk wird automatisch nach erreichbaren SIMIT Units gesucht.
2. Wählen Sie die entsprechende Registerkarte zum Importieren einer Gerätebeschreibungsdatei.
3. Um die Gerätebeschreibungsdatei zu öffnen, klicken Sie auf "Import".

Ergebnis

Die zusätzlichen Geräteinformationen werden importiert und aufgelistet. Die Geräteinformationen werden auf dem PC gespeichert, auf dem SIMIT installiert ist.

Siehe auch

Dialogfeld "SU-Verwaltung" (Seite 973)

SIMIT Unit konfigurieren (Seite 87)

2.5.5.3 Firmware der SIMIT Unit aktualisieren

Voraussetzung

- SIMIT Unit ist konfiguriert.
- SIMIT ist geöffnet.
- Firmware-Datei liegt vor.

Vorgehen

ACHTUNG
Beschädigung der SIMIT Unit während Firmware-Aktualisierung möglich Beachten Sie für die Dauer der Firmware-Aktualisierung folgende Regeln: • Schalten Sie die SIMIT Unit nicht aus. • Starten Sie die SIMIT Unit nicht neu. • Ziehen oder stecken Sie keine Kabel an der SIMIT Unit. Um eine unterbrechungsfreie Stromversorgung zu gewährleisten, schließen Sie die SIMIT Unit an eine USV an.

Um die Firmware der SIMIT Unit zu aktualisieren, gehen Sie folgendermaßen vor:

1. Wählen Sie im Menü "Extras" den Befehl "SU-Verwaltung".
   Das Dialogfeld "SU-Verwaltung" wird geöffnet. Im Netzwerk wird automatisch nach erreichbaren SIMIT Units gesucht.
2. Wählen Sie die gewünschte SIMIT Unit aus.
3. Klicken Sie auf "Firmware-Update".
4. Wählen Sie die Firmware-Datei aus.

Ergebnis

Die Firmware der SIMIT Unit wird aktualisiert. Nach erfolgreicher Aktualisierung der Firmware aktualisieren Sie die Übersicht der SIMIT Units unter "SU-Verwaltung".

Siehe auch

SIMIT Unit konfigurieren (Seite 87)

Dialogfeld "SU-Verwaltung" (Seite 973)

2.5.6 Kopplung vom Typ "SIMIT Unit" projektieren

2.5.6.1 Kopplung vom Typ "SIMIT Unit" anlegen

Einleitung

In einem SIMIT-Projekt können Sie eine Kopplung vom Typ "SIMIT Unit" anlegen. Unterhalb der SIMIT Unit legen Sie "Stationen" an, in die Sie die Hardware-Konfiguration einer Station aus STEP 7 importieren.

Voraussetzung

- Projekt ist in SIMIT geöffnet.

Vorgehen

Um Kopplung vom Typ "SIMIT Unit" anzulegen, gehen Sie folgendermaßen vor:

1. Fügen Sie in der Projektnavigation unter "Kopplungen" eine neue Kopplung hinzu.
   Das Dialogfeld "Auswahl" wird geöffnet.
2. Aktivieren Sie den Kopplungstyp "SIMIT Unit".

Ergebnis

Die Kopplung "SIMIT Unit" wird zusammen mit einer Station in der Projektnavigation angelegt.

Siehe auch

SIMIT Unit konfigurieren (Seite 87)

2.5.6.2 Station anlegen

Voraussetzung

- Projekt ist geöffnet.
- Kopplung vom Typ "SIMIT Unit" ist angelegt.
- Hardwarekonfiguration aus STEP 7 liegt vor.

Vorgehen

Um eine Station anzulegen, gehen Sie folgendermaßen vor:

1. Fügen Sie in der Projektnavigation unter "SIMIT Unit" eine neue Station hinzu.
2. Öffnen Sie die Station.
   Das Dialogfeld "SU Import" wird geöffnet.
3. Konfigurieren Sie den Import der Stationen aus der Hardwarekonfiguration.

Ergebnis

Das Ergebnis nach dem Import der Hardwarekonfiguration finden Sie
unter Hardwarekonfiguration in Station importieren (Seite 91).

Siehe auch

Dialogfeld "SU Import" (Seite 961)

2.5.6.3 Kopieren der Systemdatenbausteine

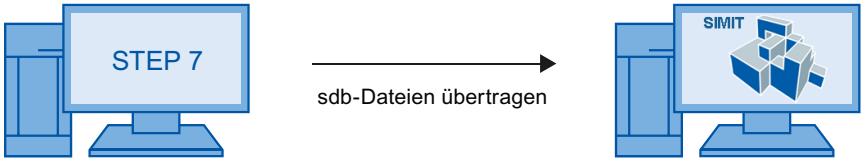
Einleitung

Wenn STEP 7 und SIMIT auf unterschiedlichen PCs installiert sind, übertragen Sie die Systemdatenbausteine mit einem von SIMIT mitgelieferten Softwarepaket:

- UnlockHWConfig.exe
- CopyHWConfig.exe

Beide Programme benötigen keine Installation und können von beliebiger Stelle aus aufgerufen werden. Die Programme sind in folgenden Ordnern abgelegt:

- SIMIT-CD: "Support\Tools"
- PC mit installiertem SIMIT: Windows-Startmenü unter "Siemens Automation > SIMIT > Tools"



UnlockHWConfig.exe

In STEP 7 werden die sdb-Dateien standardmäßig nach dem Übersetzen der Hardware-Konfiguration wieder gelöscht. Um dies dauerhaft zu verhindern, starten Sie das Programm "UnlockHWConfig.exe" einmalig auf dem STEP 7-PC. Beachten Sie, dass Sie dieses Programm mit Administratorrechten ausführen müssen!

CopyHWConfig.exe

Zum Kopieren der Systemdatenbausteine verwenden Sie das Programm "CopyHWConfig.exe". Als Quellordner sind standardmäßig folgende Ordner eingestellt:

- STEP 7 V5.5 und höher: Temporärer STEP 7-Ablageordner
- STEP 7 V13 und höher: "C:\TIAExports\"

2.5.6.4 Hardwarekonfiguration in Station importieren

Hinweis

Configuration in Run (CiR)

"Configuration in Run" wird von SIMIT nicht unterstützt. Wenn Sie die Hardware-Konfiguration in STEP 7 geändert haben, importieren Sie die Systemdatenbausteine erneut.

Einleitung

Die Hardware-Konfiguration importieren Sie direkt aus STEP 7 oder aus den kopierten Systemdatenbausteinen. Systemdatenbausteine werden beim jeden Übersetzen der Hardware-Konfiguration in STEP 7 erzeugt. Wenn Sie die Hardware-Konfiguration in STEP 7 ändern, importieren Sie die Hardware-Konfiguration wieder in die Station in SIMIT.

STEP 7-Version	Unterstützte SIMATIC-Controller	Format der Systemdatenbausteine
V5.5 und höher[1]	• S7-300 / S7-400	• SDB
ab V13 (TIA-Portal)[2]	• S7-300 / S7-400 • S7-1200 / S7-1500	• SDB • OMS

[1] Standard-Ablageordner für Systemdatenbausteine: "<STEP7 Installationsordner>\S7Tmp\SDBDATA \S7hwcnfx\DOWN\r00s0x"

[2] Standard-Ablageordner für Systemdatenbausteine: "C:\TIAExports"

Export der Hardware-Konfiguration aus STEP 7 V5.5 und höher

Exportieren Sie die Hardware-Konfiguration aus STEP 7 in folgenden Formaten:

- SDB
- CFG

Import der Hardware-Konfiguration von Drittsystemen

SIMIT unterstützt den Import der Hardware-Konfiguration von Drittsystemen im Format *.XML. Weiterführende Informationen finden Sie unter "XML-Importschnittstelle für Hardware-Konfiguration von Drittsystemen (Seite 103)".

Voraussetzung

- Kopplung vom Typ "SIMIT Unit" ist angelegt.
- Station ist unter der Kopplung angelegt.
- Hardware-Konfiguration der Station aus STEP 7 liegt vor.
- Hardware-Konfiguration in STEP 7 ist übersetzt und geladen.
- Station ist im Kopplungseditor geöffnet.

Vorgehen

Um die Hardware-Konfiguration in eine Station zu importieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie im Kopplungseditor das Dialogfeld zum Importieren der Hardware-Konfiguration.
2. Wählen Sie die Quelle, aus der Sie die Hardware-Konfiguration importieren wollen.
3. Wenn Sie die Hardware-Konfiguration aus STEP 7 V5.5 und höher importieren, wählen Sie zusätzlich die Konfigurationsdatei im Format *.CFG aus.

4. Wählen Sie die Geräte aus.
5. Klicken Sie auf "Importieren".

Ergebnis

Die Hardware-Konfiguration der Station wird importiert.

Siehe auch

Signale einer Kopplung bearbeiten (Seite 188)
Station anlegen (Seite 90)
Station konfigurieren (Seite 94)
Dialogfeld "SU Import" (Seite 961)

2.5.6.5 Hardwarekonfiguration für Shared Devices importieren

Einleitung

SIMIT unterstützt die Simulation von Shared Devices.

Voraussetzung

- Kopplung vom Typ "SIMIT Unit" ist angelegt.
- Station ist unter der Kopplung angelegt.
- Hardware-Konfiguration der Stationen aus STEP 7 liegen vor.
- Station mit den Kopfdaten für das Shared Device ist bekannt.
- Hardware-Konfiguration in STEP 7 ist übersetzt und geladen.
- Station ist im Kopplungseditor geöffnet.

Vorgehen

Um die Hardware-Konfiguration für Shared Devices in eine Station zu importieren, gehen Sie folgendermaßen vor:

1. Importieren Sie in die Station die Hardware-Konfiguration der CPU ohne Kopfdaten:
   - Öffnen Sie im Kopplungseditor das Dialogfeld zum Importieren der Hardware-Konfiguration.
   - Wählen Sie die Quelle aus.
   - Wenn Sie die Hardwarekonfiguration einer S7-300 oder S7-400 importieren, geben Sie zusätzlich die Konfigurationsdatei an.
   - Wählen Sie die Geräte aus.
   - Klicken Sie auf "Importieren".

   Der Hinweis "Unvollständiger Import von Shared Devices erkannt" wird angezeigt.
2. Importieren Sie in die Station die Hardware-Konfiguration der CPU, welche die Kopfdaten des Shared Device enthält:
   - Öffnen Sie im Kopplungseditor das Dialogfeld zum Importieren der Hardware-Konfiguration.
   - Wählen Sie die Quelle aus.
   - Wenn Sie die Hardwarekonfiguration einer S7-300 oder S7-400 importieren, geben Sie zusätzlich die Konfigurationsdatei an.
   - Aktivieren Sie die Option "Zusatzinformationen für Shared Devices".
   - Wählen Sie die Geräte aus.
   - Klicken Sie auf "Importieren".

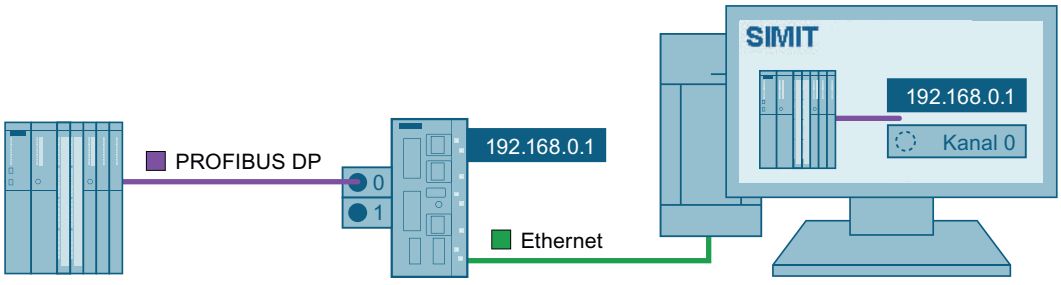
Ergebnis

Die Hardware-Konfiguration der Station aus STEP 7 wird importiert.

2.5.6.6 Station konfigurieren

Einleitung

Die folgende Abbildung zeigt schematisch, welche Informationen bei der Konfiguration einer SIMIT Unit in SIMIT notwendig sind:



Voraussetzung

- Station ist angelegt.
- IP-Adresse der SIMIT Unit ist bekannt.
- Hardwarekonfiguration ist importiert.

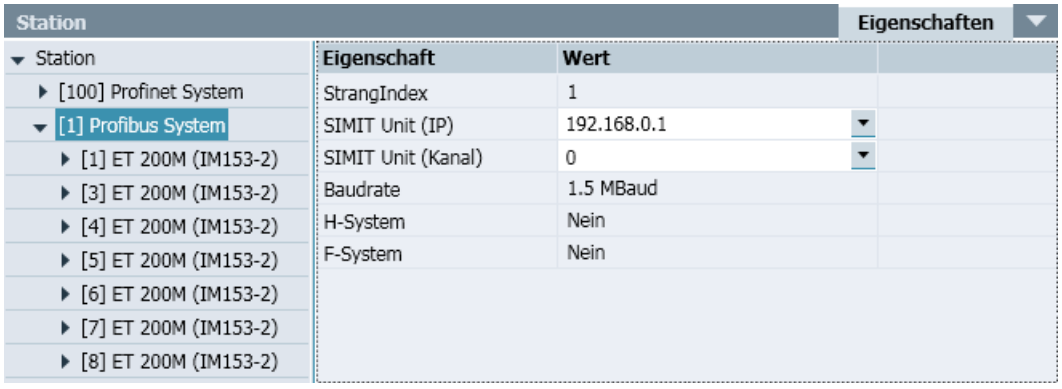
Vorgehen

Um eine Station zu konfigurieren, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in der Projektnavigation unter "Kopplungen > SIMIT Unit" auf die gewünschte Station.
2. Wählen Sie im Eigenschaftsfenster die "Zeitscheibe" der Station aus.
3. Ordnen Sie im Eigenschaftsfenster unter der "Station" jedes Mastersystems einer SIMIT Unit zu:
   - Wählen Sie die IP-Adresse der SIMIT Unit aus.
   - Wählen Sie den Kanal der SIMIT Unit aus, an der das Mastersystem angeschlossen ist.
4. Speichern Sie die Konfiguration.

Ergebnis

Die Station ist konfiguriert. Die folgende Abbildung zeigt die Zuordnung eines PROFIBUS-Mastersystems einer Station zu einer SIMIT Unit:



Siehe auch

Konfiguration in Station laden (Seite 99)

Hardwarekonfiguration in Station importieren (Seite 91)

Eigenschaften der SIMIT Unit-Kopplung (Seite 100)

2.5.6.7 Simulation von Alarmen

Alarme

In SIMIT können Sie Alarme für folgende Ebenen einer Station simulieren:

- PROFIBUS
  Strang, Slave, Modul, Kanal
- PROFINET IO
  Strang, Device, Modul, Submodul, Kanal

Alarme simulieren Sie entweder in den Eigenschaften einer Station oder über die Komponente "SetInterrupt".

Hinweis

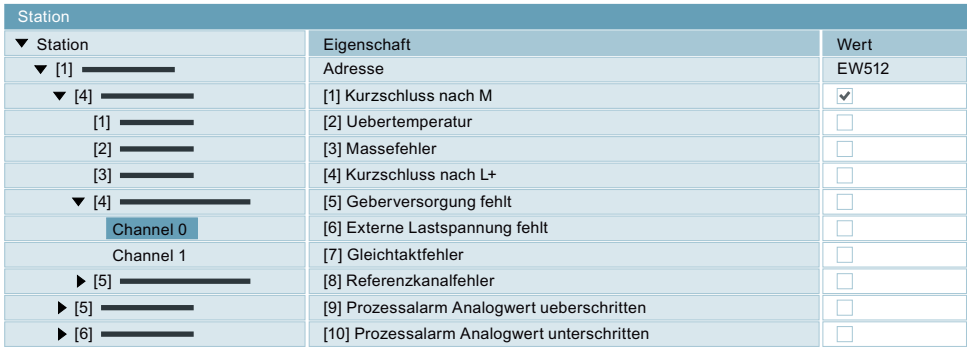
Verhalten bei migrierten Projekten aus SIMIT-Versionen kleiner V10

Um Alarme zu simulieren, importieren Sie einmal eine aktuelle Hardware-Konfiguration.

Alarm in den Eigenschaften einer Station simulieren

In Abhängigkeit von der Hardware-Konfiguration sind verschiedene Alarme simulierbar. In den Eigenschaften einer Station werden die simulierbaren Alarme für die ausgewählte Komponente angezeigt.

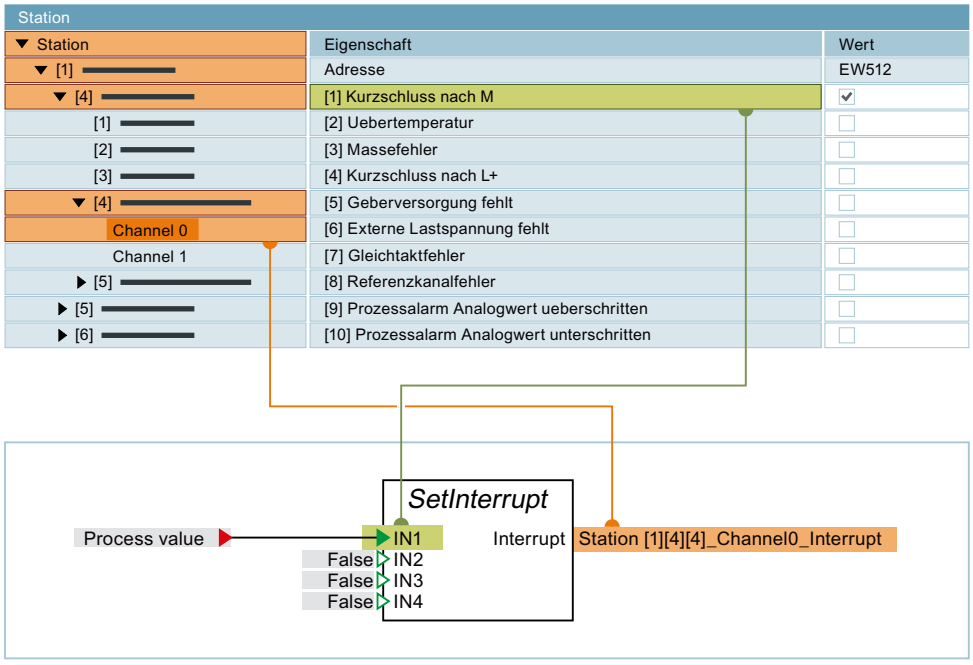
Im folgenden Beispiel aktiviert das Optionskästchen bei "[1] Kurzschluss nach M" im Channel 0 einen Kurzschluss nach Masse.



Alarm mit der Komponente "SetInterrupt" simulieren

Die Nummer des gewünschten Alarms in den Eigenschaften bestimmt die Nummer des Eingangs der Komponente "SetInterrupt".

Im abgebildeten Beispiel hat Channel 0 den Fehler "Kurzschluss nach M" unter Nummer 1. Deshalb verbinden Sie den Prozesswert mit dem Eingang "IN1" der Komponente "SetInterrupt". Channel 0 gehört zu Kopplung Station, Strang 1, Slave 4, Modul 4. Deshalb verbinden Sie das Signal Interrupt des Channel 0, also "Station [1][4][4]_Channel0_Interrupt", mit dem Ausgang der Komponente "SetInterrupt". Das Signal befindet sich in der Task-Card "Signale".

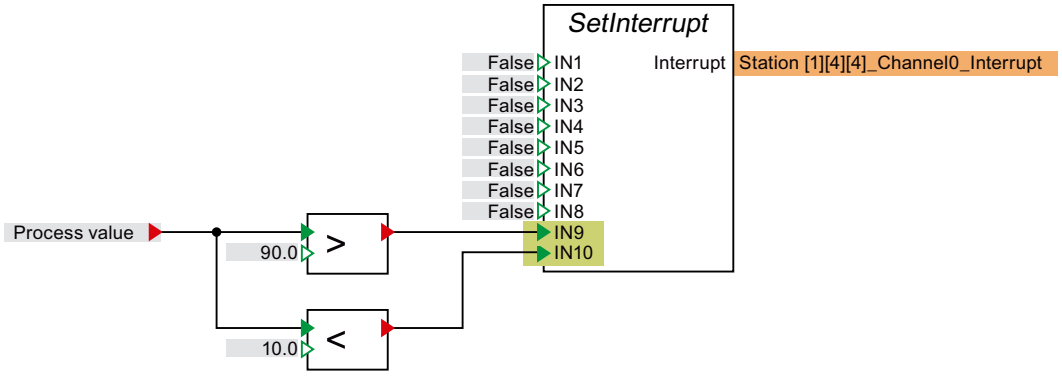


Prozessalarme

SIMATIC-Steuerungen benutzen Prozessalarme zur Überwachung binärer oder analoger Peripheriesignale: Beim Über- oder Unterschreiten eines Analogwerts oder beim Setzen oder Rücksetzen eines Binärwerts wird der zyklische Betrieb der Steuerung unterbrochen, um die Reaktion des Anwenderprogramms auf diesen Alarm auszuführen. Ob und unter welchen Bedingungen ein Gerät oder Modul Prozessalarme auslöst, ist im STEP 7-Projekt definiert.

Beispiel: Simulation eines Prozessalarms für einen Analogwert

Die folgende Abbildung zeigt ein Konfigurationsbeispiel für die Simulation eines Prozessalarms. Der Prozessalarm wird bei Über- oder Unterschreitung parametrierter Grenzwerte ausgelöst. Für die Simulation der Grenzwertüberwachung verwenden Sie die Komponente "Compare":



Siehe auch

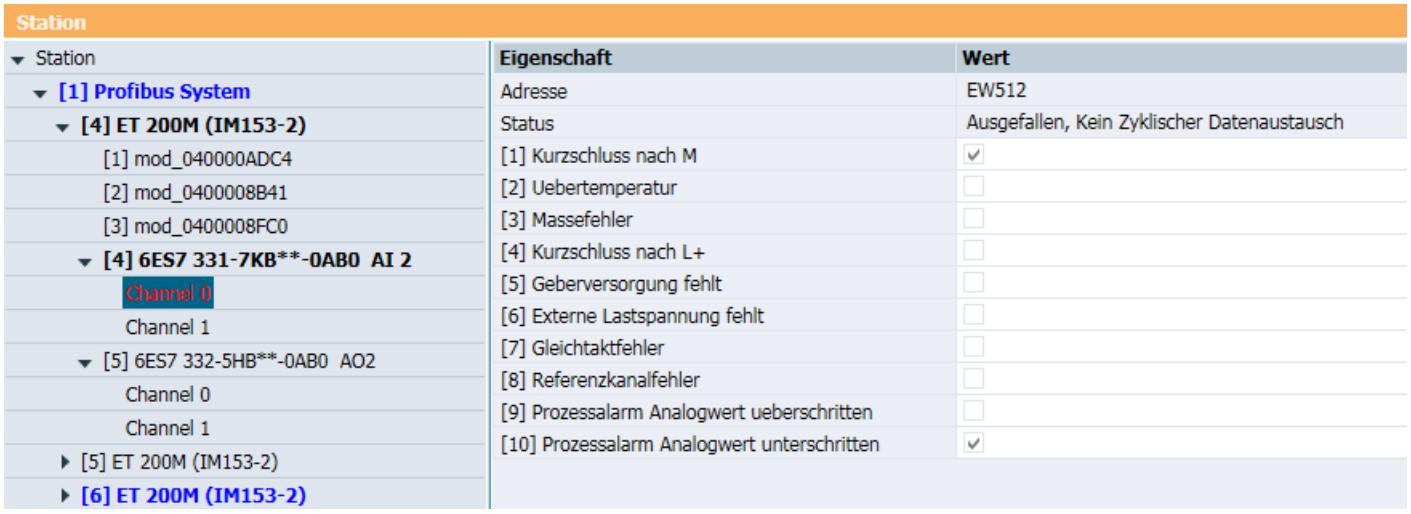
SetInterrupt (Seite 572)

2.5.6.8 Gerätestatus

Der Gerätestatus wird aus der SIMIT Unit-Hardware ermittelt und im Hardware-Baum dargestellt. Die Hervorhebungen haben folgende Bedeutung:

- Rot: Fehler
- Blau: Warnung
- Fett: Teilbaum mit Fehlern oder Warnungen

Im abgebildeten Beispiel ist ein Fehler im Channel 0. Channel 0 ist deshalb rot hervorgehoben. Das Modul [4], der Slave [4] und der Strang [1] sind fett hervorgehoben, weil ein untergeordnetes Objekt einen Fehler hat. Außerdem gibt es Warnungen für Slave [6] und Strang [1]. Slave und Strang werden deshalb blau hervorgehoben. In den Eigenschaften steht der Status der markierten Komponente, z. B. "Ausgefallen, Kein Zyklischer Datenaustausch" oder "Zyklischer Datenaustausch".



2.5.6.9 Strang, Gerät oder Modul deaktivieren

Einleitung

Um z. B. während einer Simulation den Ausfall eines Strangs, eines Geräts oder eines Moduls zu simulieren, deaktivieren Sie den Strang, das Gerät oder das Modul in den Eigenschaften der Kopplung. Auf diese Weise testen Sie z. B. die Reaktion des Controllers auf Ausfall und Wiederkehr eines Strangs, Geräts oder Moduls.

Das Deaktivieren funktioniert sowohl bei laufender Simulation als auch während der Konfiguration.

Voraussetzung

- Kopplung vom Typ "SIMIT Unit" ist angelegt.
- Station ist im Kopplungseditor geöffnet.
- Station ist konfiguriert.

Vorgehen

Um einen Strang, ein Gerät oder ein Modul zu deaktivieren, gehen Sie folgendermaßen vor:

1. Um einen Strang zu deaktivieren:
   - Wählen Sie in den Eigenschaften den Strang aus.
   - Aktivieren Sie die Option "Strang Ausfall".
2. Um ein Gerät zu deaktivieren:
   - Wählen Sie in den Eigenschaften das Gerät aus.
   - Aktivieren Sie die Option "Slave Ausfall".
3. Um ein Modul zu deaktivieren:
   - Wählen Sie in den Eigenschaften das Modul aus.
   - Aktivieren Sie die Option "Modul ziehen".

Ergebnis

Der Strang, das Gerät oder das Modul ist deaktiviert.

Siehe auch

Station konfigurieren (Seite 94)

2.5.6.10 Konfiguration in Station laden

Voraussetzung

- Mastersystem ist an Simulation angeschlossen.
- SIMIT Unit ist mit dem PC verbunden, auf dem SIMIT installiert ist.
- Station ist konfiguriert.

Vorgehen

Hinweis

Während des Ladens wird die Kommunikation zwischen Mastersystem und Controller kurzzeitig unterbrochen.

Um eine Konfiguration in eine Station zu laden, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Station.
2. Wenn Sie das SIMIT-Projekt zusätzlich in der SIMIT Unit speichern wollen, aktivieren Sie in den Eigenschaften der Station die Option "Projekt permanent laden".
3. Klicken Sie im Kopplungseditor auf "Speichern und Laden".

Ergebnis

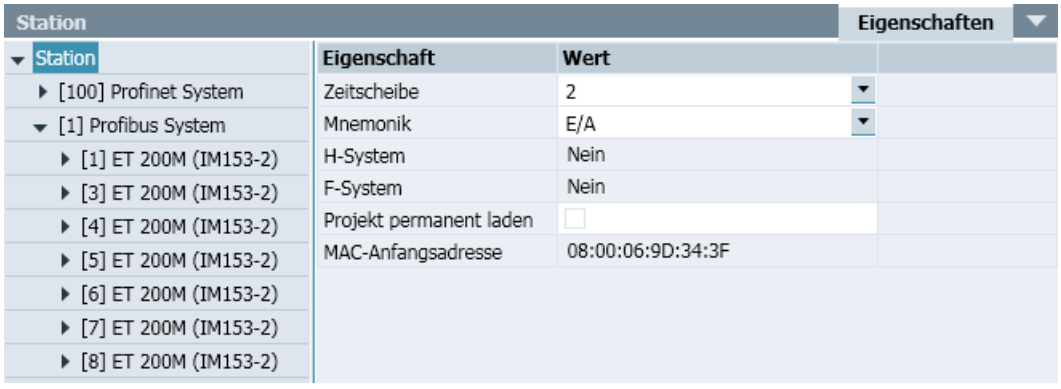
Die Konfiguration wird in die SIMIT Unit geladen. Nach dem Abschluss des Ladevorgangs können Sie die Simulation für diese Station starten.

Siehe auch

Station konfigurieren (Seite 94)

2.5.6.11 Eigenschaften der SIMIT Unit-Kopplung

Eigenschaften einer Station



- **Zeitscheibe**
  Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, das entspricht einem Zyklus von 100 ms.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- **Mnemonik**
  Legt fest, ob die deutsche ("E/A") oder englische ("I/Q") Bezeichnung der Absolutadressen verwendet wird.

  Hinweis

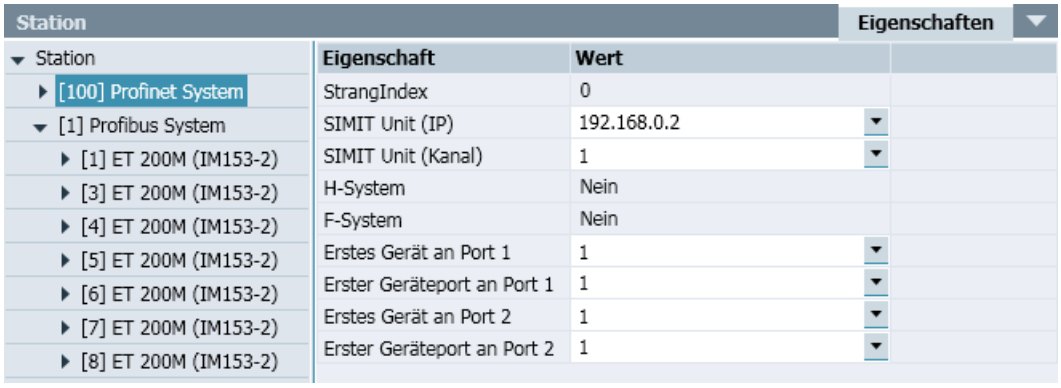
  Legen Sie die Mnemonik direkt nach Import fest. Wenn Sie die Mnemonik nachträglich umschalten, können bereits in einem Diagramm verschaltete Signale nicht mehr der ursprünglichen Adresse zugeordnet werden.

- **H-System**
  Zeigt an, ob das System ein H-System ist.
- **F-System**
  Zeigt an, ob das System ein F-System ist.
- **Projekt permanent laden**
  Legt fest, dass das Simulationsprojekt zusätzlich in der SIMIT Unit gespeichert wird. Bei einem Neustart der SIMIT Unit können Sie die Simulation fortsetzen, ohne dass Sie das Simulationsprojekt erneut in die SIMIT Unit laden müssen.
  Um das Simulationsprojekt aus der SIMIT Unit zu löschen, deaktivieren Sie die Option und laden Sie die Konfiguration in die SIMIT Unit.

Allgemeine Eigenschaften einer Anbindung über PROFIBUS/PROFINET

- **StrangIndex**
  Zeigt die Nummer des Mastersystems an.
- **SIMIT UNIT (IP)**
  Legt die IP-Adresse der Simulation Unit fest, mit welcher der SIMIT-PC verbunden ist.
- **SIMIT Unit (Kanal)**
  Legt den Kanal fest, an dem der Controller an der SIMIT Unit angeschlossen ist.
- **H-System**
  Zeigt an, ob das System ein H-System ist.
- **F-System**
  Zeigt an, ob das System ein F-System ist.

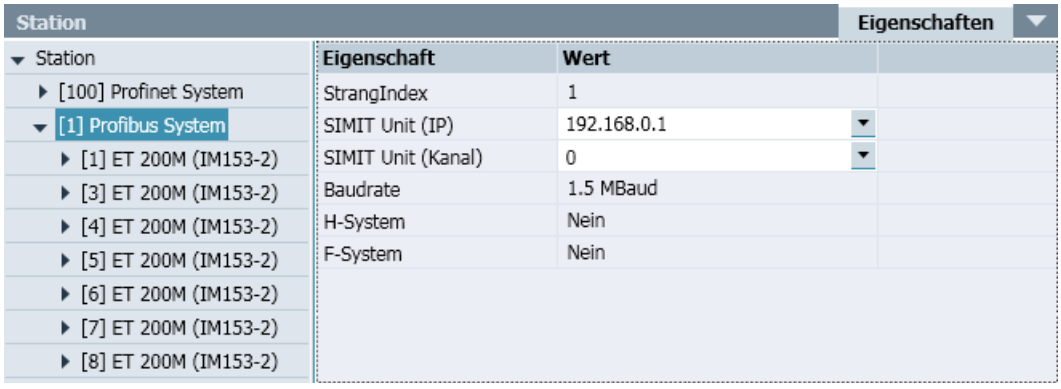
Spezifische Eigenschaften einer Anbindung über PROFINET



- **Erstes Gerät an Port 1**
  Legt das Gerät fest, das vom PROFINET-Controller aus gesehen in der Netzprojektierung als erstes Gerät projektiert ist.
- **Erster Geräteport an Port 1**
  Legt den Port fest, mit dem das Gerät mit dem PROFINET-Controller verbunden ist. Nur notwendig, wenn Topologie projektiert ist.
- **Erstes Gerät an Port 2**
  Legt das Gerät fest, das vom PROFINET-Controller aus gesehen in der Netzprojektierung als erstes Gerät projektiert ist. Nur notwendig, wenn Topologie projektiert ist.
- **Erster Geräteport an Port 2**
  Legt den Port fest, mit dem das Gerät mit dem PROFINET-Controller verbunden ist. Nur notwendig, wenn Topologie projektiert ist.

Ein Konfigurationsbeispiel finden Sie unter Unterstützte PROFINET IO-Konfigurationen (Seite 76).

Spezifische Eigenschaften einer Anbindung über PROFIBUS



- **Baudrate**
  Zeigt die Übertragungsgeschwindigkeit an.

Siehe auch

Station konfigurieren (Seite 94)

2.5.6.12 XML-Importschnittstelle für Hardware-Konfiguration von Drittsystemen

Grundlagen zum XML-Import

SIMIT unterstützt den Import von Hardware-Konfigurationsdaten von Steuerungen von Drittanbietern über eine XML-Hardwareimport-Schnittstelle.

Eine XML-Datei, die den im folgenden beschriebenen Formaten entspricht, kann in SIMIT importiert werden (SU Import/XML-Import). Damit wird ein lauffähiges und ladbares PROFIBUS- oder PROFINET-Projekt erzeugt.

Der Anwender kann so durch einen geeigneten Export der Hardwaredaten aus seinem Projektierungstool oder auch mittels eines XML-Editors relativ einfach eine Simulation aufbauen.

XML-Format für den Import eines PROFIBUS-Stranges

Im folgenden wird der Aufbau der XML-Datei beschrieben, die alle für SIMIT notwendigen Daten für die Simulation eines PROFIBUS-Stranges enthält.

`<?xml version="1.0" encoding="iso-8859-1"?>` `<HARDWARE>`	XML-Deklarationszeile Root-Tag, (immer "HARDWARE")
`<COMMENT/>`	Kommentar, optional
`<NAME>SimbaProfibus_0[0]</NAME>`	Name der Simulationsbox, Name ist frei vergebbar, die Zahl in den eckigen Klammern am Ende kennzeichnet den Kanal der Box und muss vorhanden sein.
`<HWTYPE>DPBOX</HWTYPE>`	Typ der Simulationsbox, für PROFIBUS "DPBOX"
`<BAUDRATE>6</BAUDRATE>`	PROFIBUS-Baudrate: 9 = 12 Mbd, 8 = 6 MBd, 7 = 3 Mbd, 6 = 1,5 Mbd ...
`<DPSUBSYSTEM>1</DPSUBSYSTEM>`	PROFIBUS-System-ID
`<CHILDTYPE>SLAVE</CHILDTYPE>`	Interner Schlüssel für die nächste Hierarchiestufe.
`<SLAVE>`	Tag für das PROFIBUS-Gerät
`<NAME>My ET200</NAME>`	Name (optional)
`<ORDERNUMBER>6ES7 326-1BK01-0AB0</ORDERNUMBER>`	Bestellnummer (optional)
`<INDEX>3</INDEX>`	PROFIBUS-Adresse
`<TYPE>801E</TYPE>`	Typkennung gemäß GSD-Datei
`<CHILDTYPE>MODULE</CHILDTYPE>`	Schlüssel für die nächste Hierarchiestufe
`<REDUNDANCY>1</REDUNDANCY>`	Redundanz, nur bei redundanten Bussystemen anzugeben
`<MODULE>`	Tag für Gerätemodule
`<NAME>My DI16</NAME>`	Name, optional

<INDEX>4</INDEX>	Steckplatz
<TYPE>4301009FC2</TYPE>	Modultyp gemäß GSD-Datei
<CHILDTYPE>CHANNEL</CHILDTYPE>	Nächste Hierarchiestufe
<INLEN>2</INLEN>	Eingangslänge des Moduls in Bytes
<OUTLEN>0</OUTLEN>	Ausgangslänge des Moduls in Bytes
<LOGIN>0</LOGIN>	Logische Eingangsadresse, muss AS-weit eindeutig sein. Datenaustausch von der Oberfläche erfolgt über diese Adresse.
</MODULE>	
...	Weitere Module können folgen.
<MODULE>	
<COMMENT/>	
<NAME>DI 24xFailsafe </NAME>	
<ORDERNUMBER>6ES7 326-1BK01-0AB0</ORDERNUMBER>	
<INDEX>8</INDEX>	
<TYPE>C2838908C1</TYPE>	
<CHILDTYPE>CHANNEL</CHILDTYPE>	
<INLEN>10</INLEN>	
<OUTLEN>4</OUTLEN>	
<LOGIN>4</LOGIN>	
<LOGOUT>4</LOGOUT>	Logische Ausgangsadresse, muss AS-weit eindeutig sein. Datenaustausch von der Oberfläche erfolgt über diese Adresse.
</MODULE>	
</SLAVE>	
<SLAVE>	Weitere Geräte können folgen
...	
</SLAVE>	
...	
</HARDWARE>	

XML-Format für den Import eines PROFINET-Stranges

Die XML-Datei für den Import eines PROFINET-Stranges unterscheidet sich von der eines Profibus-Stranges auf Grund einer anderen Busphysik und damit verbunden anderer notwendiger Parameter.

<HARDWARE>	
<COMMENT/>	
<NAME>SimbaPNIO_2[0]</NAME>	Name der Simulationsbox, Name ist frei vergebbar, die Zahl in den eckigen Klammern am Ende kennzeichnet den Kanal der Box und muss vorhanden sein.
<HWTYPE>PNIO</HWTYPE>	Typ der Simulationsbox. Für PROFINET "PNIO"
<IOSUBSYSTEM>100</IOSUBSYSTEM>	PROFINET-Sytem-ID
<FIRSTDEVICE>6</FIRSTDEVICE>	Topologieinformation, das Device, das direkt hinter dem Controller und hängt und damit an der externen Schnittstelle der Simulationsbox als erstes simuliert wird.

<FIRSTPORT>1</FIRSTPORT>	Der entsprechende Port dazu.
<SECONDDEVICE>6</SECONDDEVICE>	Topologieinformation für den zweiten Kanal bei redundanten Systemen.
<SECONDPORT>2</SECONDPORT>	Zweiter Port bei redundanten Systemen.
<CHILDTYPE>PNIODEV</CHILDTYPE>	Child-Typ der nächsten Hierarchiestufe.
<PNIODEV>	Tag für ein PROFINET-Gerät.
<NAME>My PROFINET IO</NAME>	Name des Gerätes (optional).
<INDEX>6</INDEX>	Geräteindex
<TYPE>002A0302</TYPE>	Gerätetyp gemäß GSDML
<IPADDR>192.168.0.4</IPADDR>	Simulierte IP-Adresse.
<SubnetMask>255.255.255.0</ SubnetMask >	Subnetmaske dazu
<PRJNAME>im153-4pn</PRJNAME>	Gerätename im Projekt, wird vom Controller zur Identifizierung des Gerätes per DCP-Telegramm benötigt.
<CHILDTYPE>PNIOSLOT</CHILDTYPE>	Typ der nächsten Hierarchiestufe.
<PNIOSLOT>	Tag für einen PNIO-Slot.
<NAME>! IM153-4 PN HF V4.0</NAME>	Name (optional)
<ORDERNUMBER>6ES7 153-4BA00- 0XB0</ ORDERNUMBER>	Bestellnummer (optional), die Bestellnummer kann nützlich sein, wenn eine Unterscheidung verschiedener Module mit gleicher Typkennung erfolgen soll.
<INDEX>0</INDEX>	Steckplatz, 0 ist bei PROFINET immer das Kopfmodul. Das muss im Gegensatz zu Profibus-Geräten hier zwingend mit angegeben werden.
<TYPE>00000423</TYPE>	Slot-Typ gemäß GSDML
<API>0</API>	API-Nr.
<HEAD>8</HEAD>	Kennzeichnung des Slot als Kopfmodul, darf nur einmal im Device vorhanden sein.
<CHILDTYPE>SUBSLOT</CHILDTYPE>	Typ der nächsten Hierarchieebene
<SUBSLOT>	Tag für einen Pnio-Subslot
<NAME>DIM 9 HF V4.0</NAME>	Name (optional)
<INDEX>1</INDEX>	Fortlaufender Index innerhalb des Slot
<TYPE>00000000</TYPE>	Typ gemäß GSDML
<INLEN>0</INLEN>	IO-Daten-Inputlänge in Bytes
<OUTLEN>0</OUTLEN>	IO-Daten-Outputlänge in Bytes
<IOPS_Length>1</IOPS_Length>	IO-Provider-Modulstatuslänge in Bytes
<IOCS_Length>1</IOCS_Length>	IO-Consumer-Modulstatuslänge in Bytes
<CHILDTYPE>CHANNEL</CHILDTYPE>	Typ der nächsten Hierarchieebene
</SUBSLOT>	
<SUBSLOT>	
<NAME>PN-IO</NAME>	Subslot für das Pnio-Interface
<INDEX>32768</INDEX>	
<TYPE>00008002</TYPE>	
<INLEN>0</INLEN>	
<OUTLEN>0</OUTLEN>	
<IOPS_Length>1</IOPS_Length>	
<IOCS_Length>1</IOCS_Length>	

</SUBSLOT>	
<SUBSLOT>	
<NAME>Port 1</NAME>	Subslot für einen Port des Pnio-Interface
<INDEX>32769</INDEX>	
<TYPE>0000C000</TYPE>	
<INLEN>0</INLEN>	
<OUTLEN>0</OUTLEN>	
<IOPS_Length>1</IOPS_Length>	
<IOCS_Length>1</IOCS_Length>	
</SUBSLOT>	
<SUBSLOT>	
<NAME>Port 2</NAME>	
<INDEX>32770</INDEX>	
<TYPE>0000C000</TYPE>	
<INLEN>0</INLEN>	
<OUTLEN>0</OUTLEN>	
<IOPS_Length>1</IOPS_Length>	
<IOCS_Length>1</IOCS_Length>	
</SUBSLOT>	
</PNIOSLOT>	
<PNIOSLOT>	Slot-Beispiel für eine Input-Baugruppe
<NAME>SM 321 DI16xDC24V</NAME>	
<ORDERNUMBER>6ES7 321-1BH50-0AA0</ ORDERNUMBER>	
<INDEX>1</INDEX>	
<TYPE>00009FC2</TYPE>	
<API>0</API>	
<CHILDTYPE>SUBSLOT</CHILDTYPE>	
<SUBSLOT>	
<TYPE>00000000</TYPE>	
<INDEX>1</INDEX>	
<NAME>14</NAME>	
<INLEN>2</INLEN>	
<OUTLEN>0</OUTLEN>	
<IOPS_Length>1</IOPS_Length>	
<IOCS_Length>1</IOCS_Length>	
<LOGIN>6</LOGIN>	Logische Eingangsadresse, muss AS-weit eindeutig sein. Datenaustausch von der Oberfläche erfolgt über diese Adresse.
<CHILDTYPE>CHANNEL</CHILDTYPE>	
</SUBSLOT>	
</PNIOSLOT>	
...	Weitere belegte Slots
<PNIOSLOT>	

<NAME>FDO10xDC24V/2A</NAME>	
<INDEX>4</INDEX>	
<TYPE>000008C1</TYPE>	
<API>0</API>	
<CHILDTYPE>SUBSLOT</CHILDTYPE>	
<SUBSLOT>	
<TYPE>00000000</TYPE>	
<INDEX>1</INDEX>	
<INLEN>6</INLEN>	
OUTLEN>6</OUTLEN>	
<IOPS_Length>1</IOPS_Length>	
<IOCS_Length>1</IOCS_Length>	
<LOGIN>18</LOGIN>	
<LOGOUT>18</LOGOUT>	Logische Ausgangsadresse, muss AS-weit eindeutig sein. Datenaustausch von der Oberfläche erfolgt über diese Adresse.
</SUBSLOT>	
</PNIOSLOT>	
</PNIODEV>	
...	Weitere Pnio-Geräte
<PNIODEV>	
...	
</PNIODEV>	
</HARDWARE>	

Gleichzeitiger Import mehrerer Stränge

Für den gleichzeitigen Import mehrerer Bus-Stränge können Sie die einzelnen "HARDWARE"-Tags innerhalb eines "PROJECT"-Tag anordnen. Das kan z. B. sinnvoll sein, um die beiden Bus-Stränge eines H-Systems gleichzeitig zu erzeugen.

<?xml version="1.0" encoding="iso-8859-1"?> <PROJECT>	XML-Deklarationszeile Root-Tag, (jetzt "PROJECT")
<HARDWARE>	erste "HARDWARE"
...	
</HARDWARE>	
<HARDWARE>	zweite "HARDWARE"
...	
</HARDWARE>	
</PROJECT>	

Syntaktische Validierung der erzeugten XML-Dateien

Die für den Import in SIMIT erzeugten XML-Dateien können vor dem Import auf syntaktische Fehler oder fehlende Schlüssel überprüft werden. Zu diesem Zweck steht eine Dokument-Typ-Definition (DTD-Datei) zur Verfügung. Hierbei wird jedoch nicht der Inhalt der Tags auf Sinnfälligkeit überprüft.

Zur Überprüfung Ihrer XML_Datei kopieren Sie folgenden Textblock ab der zweiten Zeile in Ihre XML-Datei und laden dann das ganze in einen DTD-Validator.

```
<!ELEMENT HARDWARE (COMMENT?, NAME?, HWTYPE, ((DPSUBSYSTEM, BAUDRATE,
CHILDTYPE, SLAVE*) | (IOSUBSYSTEM, FIRSTDEVICE, FIRSTPORT, SECONDDEVICE?,
SECONDPORT?, CHILDTYPE, PNIODEV*)) )>
<!ELEMENT COMMENT (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT HWTYPE (#PCDATA)>
<!ELEMENT CHILDTYPE (#PCDATA)>
<!ELEMENT BAUDRATE (#PCDATA)>
<!ELEMENT DPSUBSYSTEM (#PCDATA)>
<!ELEMENT IOSUBSYSTEM (#PCDATA)>
<!ELEMENT FIRSTDEVICE (#PCDATA)>
<!ELEMENT FIRSTPORT (#PCDATA)>
<!ELEMENT SECONDDEVICE (#PCDATA)>
<!ELEMENT SECONDPORT (#PCDATA)>
<!ELEMENT SLAVE (COMMENT?, NAME?, ORDERNUMBER?, INDEX, TYPE, CHILDTYPE,
REDUNDANCY?, MODULE*)>
<!ELEMENT INDEX (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT ORDERNUMBER (#PCDATA)>
<!ELEMENT REDUNDANCY (#PCDATA)>
<!ELEMENT MODULE (COMMENT?, NAME?, ORDERNUMBER?, INDEX, TYPE, CHILDTYPE,
INLEN, OUTLEN, LOGIN?, LOGOUT?)>
<!ELEMENT INLEN (#PCDATA)>
<!ELEMENT OUTLEN (#PCDATA)>
<!ELEMENT LOGIN (#PCDATA)>
<!ELEMENT LOGOUT (#PCDATA)>
<!ELEMENT FAILSAFE (#PCDATA)>
<!ELEMENT PNIODEV (COMMENT?, NAME?, ORDERNUMBER?, INDEX, TYPE, IPADDR,
PRJNAME, CHILDTYPE, PNIOSLOT*) >
<!ELEMENT IPADDR (#PCDATA)>
<!ELEMENT PRJNAME (#PCDATA)>
<!ELEMENT PNIOSLOT (COMMENT?, NAME?, ORDERNUMBER?, INDEX, TYPE, API, HEAD?,
CHILDTYPE, SUBSLOT*) >
<!ELEMENT API (#PCDATA)>
<!ELEMENT HEAD (#PCDATA)>
<!ELEMENT SUBSLOT (NAME?, INDEX, TYPE, INLEN, OUTLEN, IOPS_Length,
IOCS_Length, LOGIN?, LOGOUT?, CHILDTYPE?) >
<!ELEMENT IOPS_Length (#PCDATA)>
<!ELEMENT IOCS_Length (#PCDATA)>
```

Wenn Sie den Text nicht jedes Mal kopieren wollen, kopieren Sie den Textblock in eine Textdatei (z. B. "SIMITXMLImport.dtd") und für die Validierung in Ihre XML-Dateien in die zweite Zeile nur einen Link auf diese Textdatei einbauen.

```
<!DOCTYPE HARDWARE SYSTEM "SIMITXMLImport.dtd">
```

Für die Verwendung in einer externen DTD-Datei sind die erste und letzte Zeile des Textblocks wegzulassen.

XML-Validierer im Internet sind z. B.:

- https://www.xmlvalidation.com (https://www.xmlvalidation.com)
- https://www.w3schools.com/xml/xml_validator.asp (https://www.w3schools.com/xml/xml_validator.asp)

Diese arbeiten jedoch nur, wenn die DTD direkt in die XML-Datei integriert ist.

Für eine Offline-XML-Validierung eignet sich das "XMLNotepad2007" von Microsoft.

2.6 Virtual Controller

2.6.1 Funktionsweise des Virtual Controller

2.6.1.1 Einleitung

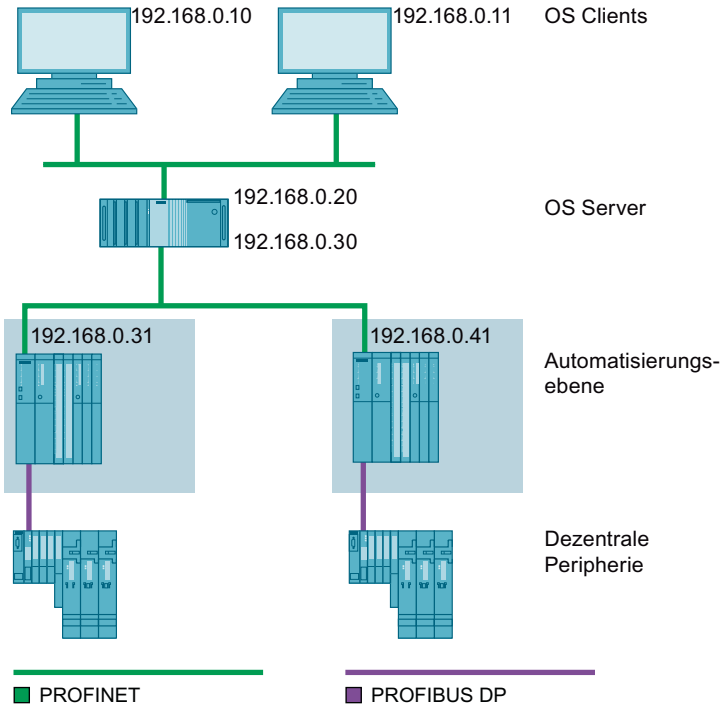
Funktion

Der Virtual Controller bildet das Verhalten einer SIMATIC-Steuerung vom Typ S7-300 oder S7-400 nach. Der Virtual Controller wird mit dem originalen SPS-Anwenderprogramm geladen. Die Feldgeräteebene mit ihrer Anbindung über dezentrale Bussysteme wird nicht berücksichtigt. Die Prozess-Simulation wird stattdessen direkt über das Prozessabbild der Ein- und Ausgänge angebunden.

Im Simulationsprojekt können Sie bis zu 32 Virtual Controller einsetzen, die Sie auch auf mehrere PCs verteilen können.

Virtual Controller können untereinander kommunizieren, auch Verbindungen zu externen Partnern und dem Bedien- und Beobachtungssystem sind möglich. Allerdings werden ausschließlich IP-basierte Verbindungen im Adressraum IPv4 unterstützt.

Die folgende Abbildung zeigt, welche Komponenten der Automatisierungsebene als Virtual Controller abgebildet werden:



In SIMIT werden alle Virtual Controller eines Simulationsprojekts in der Projektnavigation unter "Kopplungen > Virtual Controller" als eigenständige Kopplungen dargestellt und konfiguriert. Der Datenaustausch zwischen SIMIT und dem Virtual Controller findet über diese Kopplungen statt.

Bestimmungsgemäßer Einsatz eines Virtual Controller

ACHTUNG

Wichtige Hinweise:

- Mit dem Virtual Controller sind Automatisierungsprogramme unter MS Windows ablauffähig. Da MS Windows kein Echtzeit-Betriebssystem ist, müssen Sie folgende Einschränkungen in Kauf nehmen:
  - Verfügbarkeit des Systems
  - Genauigkeit der Nachbildung

  Zusätzlich können Sie reale E/A- und Bussysteme nicht ankoppeln.
- Auch wenn die Funktionen des Virtual Controller für die vorgesehenen Einsatzzwecke (Test- und Trainingssysteme) geprüft wurden, kann keine Gewähr dafür abgegeben werden, dass das System in jedem Fall das zeitliche Verhalten exakt abbilden kann. Die zur Verfügung gestellten Ersatzfunktionen können nicht in jedem Fall den Funktionsumfang der realen Steuerung vollständig abbilden.

Das System ist deshalb nicht für die Steuerung realer Anlagen vorgesehen.

Die Ergebnisse, die für eine virtuelle Inbetriebnahme ermittelt wurden, müssen vom Anwender entsprechend überprüft und bewertet werden.

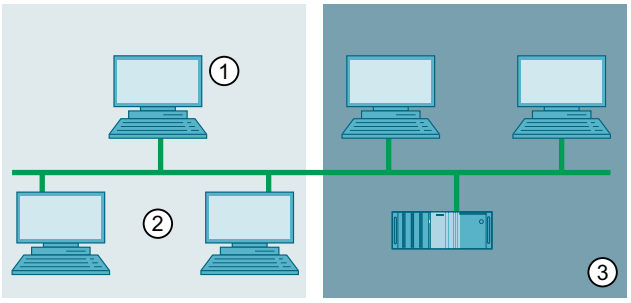
Benutzen Sie zur Wahrung der Vertraulichkeit Ihrer Daten den Virtual Controller nur im Rahmen eines ganzheitlichen Industrial Security-Konzepts, das dem aktuellen Stand der Technik entspricht. Berücksichtigen Sie dabei auch eingesetzte Produkte von anderen Herstellern.

Weitere Informationen finden Sie im Internet (http://www.siemens.com/industrialsecurity).

Netzwerktopologie einer Simulation mit Virtual Controller

Das gesamte Simulationsprojekt kann inklusive der Bedienung und Beobachtung auch auf einem einzigen PC ablaufen. Wenn die Rechenleistung eines PC nicht ausreicht, können Sie Virtual Controller auf mehrere PCs verteilen.

Die folgende Abbildung zeigt die Netzwerktopologie mit auf mehreren PCs verteilten Virtual Controllern:



TCP/IP

① PC mit installierter SIMIT Simulation Platform
- Verwaltet das SIMIT-Projekt
- Startet/stoppt die Simulation
- Simuliert die Feld- und Prozessebene
- Kann zusätzlich Virtual Controller beinhalten.

② PC mit installiertem SIMIT VC (Virtual Controller)
- Hostet 1 bis 8 Virtual Controller
- Muss vom SIMIT-PC aus im Netzwerk erreichbar sein
- Muss vom OS Server aus erreichbar sein

③ PCs mit installiertem PCS 7 (ES, OS Server und Client)

Hinweis

Unterstützte STEP 7-Version

Die Virtual Controller-Kopplung wird bis einschließlich STEP 7 V5.5 unterstützt.

Hinweis

Einsatz von CFC-Plänen auf einer im Virtual Controller nachgebildeten S7-410

Wenn Sie mit dem Virtual Controller eine S7-410 nachbilden, beachten Sie bei Verwendung von CFC-Plänen Folgendes:

Sie können ein Anwenderprogramm ab CFC V8.2 Update 3 uneingeschränkt laden. Wenn Sie eine ältere Version von CFC einsetzen, gehen Sie auf jedem als Virtual Controller konfigurierten PC folgendermaßen vor:

1. Wechseln Sie in Windows in folgendes Verzeichnis:
   "Programs (x86)\Siemens\Automation\SIMIT\SIMIT VC\data\CPU"
2. Benennen Sie folgende Dateien um:
   – "6ES7 410 5HN08-0AB0" → "6ES7 410 5HN08-0AB0_CPU410"
   – "6ES7 410 5HX08-0AB0" → "6ES7 410 5HX08-0AB0_CPU410"
3. Benennen Sie folgende Dateien um:
   – "6ES7 410 5HN08-0AB0_LEGACY" → "6ES7 410 5HN08-0AB0"
   – "6ES7 410 5HX08-0AB0_LEGACY" → "6ES7 410 5HX08-0AB0"

Mit dieser Maßnahme meldet sich der Virtual Controller als S7-417 zurück. Sie können das Projekt mit CFC-Plänen von PCS 7 aus in die PLC laden. Wenn Sie Ihre CFC-Installation auf die neueste Version aktualisieren, machen Sie die Umbenennung wieder rückgängig.

Hinweis

Delta-Download

Mit jedem Delta-Download legt SIMIT einen Zeitstempel im Controller ab. Anhand dieses Zeitstempels wird entschieden, ob das Programm im Controller ("Online") und in SIMIT ("Offline") identisch ist. Der Virtual Controller unterstützt das Speichern dieses Zeitstempels nicht über einen Neustart der Simulation hinweg, legt das Programm aber korrekt ab. SIMIT wird daher nach Deltadownload und Neustart der Simulation beim Onlinegehen fälschlicherweise einen inkonsistenten Stand des Programms melden. Sie können diese Meldung vermeiden, indem Sie einen vollständigen Download durchführen.

Hinweis

BlockPrivacy

Der Virtual Controller kann keine Blöcke ausführen, die mit "BlockPrivacy" geschützt sind. Wenn Ihr Projekt solche geschützten Blöcke enthält, werden diese im Virtual Controller durch ein NOP (No Operation) ersetzt.

2.6.1.2 Anforderungen an ein Simulationsnetzwerk

Vermeiden von IP-Adresskonflikten

Beim Einsatz eines Virtual Controller werden alle seine projektierten IP-Adressen und die darauf aufsetzenden Verbindungen übernommen. In den Netzwerkeinstellungen des PCs müssen Sie deshalb diese zusätzlichen IP-Adressen eintragen.

ACHTUNG
Hinzufügen und Ändern von IP-Adressen Das Ändern einer IP-Adresse kann zu Kommunikationsproblemen im Netzwerk führen, z. B. kann ein PC nicht mehr erreichbar sein. Ändern Sie IP-Adressen nie ohne Rücksprache mit Ihrem IT-Administrator.

Weiterführende Informationen finden Sie unter TCP/IP-Adressierung und Subnetze (https://support.microsoft.com/de-de/kb/164015).

Regeln zur Simulation mit einem Virtual Controller

Beachten Sie folgende Regeln:

- Wenn ein Virtual Controller eine Verbindung zum Bedienen und Beobachten herstellen soll, muss der Virtual Controller über den Anlagenbus erreichbar sein.
- Der PC, auf dem SIMIT Simulation Platform installiert ist, muss mit seiner IP-Adresse im gleichen Subnetz liegen wie die PCs, auf denen ein Virtual Controller läuft.
- Jede IP-Adresse ist eindeutig.

- Jeder Computername ist eindeutig.
- Die IP-Adressen der simulierten Stationen sind in den Netzwerkeinstellungen der jeweiligen PCs eingetragen.

Siehe auch

Virtual Controller projektieren (Seite 128)

2.6.1.3 Funktionen

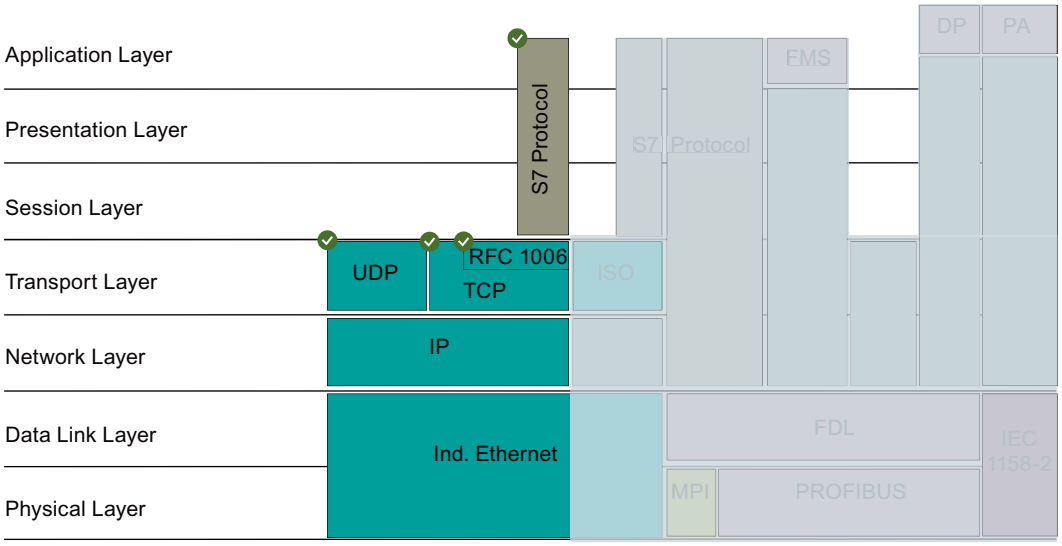
Die Virtual Controller unterstützen folgende Grundfunktionen einer Steuerung, unabhängig vom individuellen Typ:

- Download der Hardwarekonfiguration und Applikationssoftware in den virtuellen Speicher der Virtual Controller
- Interpretation von MC7-Code
- Bereitstellung von Datenbereichen, Rechenwerken, Akkus und Verknüpfungsergebnissen, Zählern, Merkern, Timern, Speicherbereichen für die Prozessabbilder usw.
- OB-Verwaltung und Scheduling
- Kommunikationsfunktionen
- Eine Auswahl an Systemfunktionen (SFB/SFC), soweit sie im Rahmen der Virtualisierung möglich und sinnvoll sind.

2.6.1.4 Grundlagen zur Virtual Controller-Kommunikation

Einleitung

Der Virtual Controller unterstützt die IP-basierte Kommunikation über Ethernet / PROFINET und folgende Protokolle:



Wird unterstützt:

- S7
- S7H
- TCP
- UDP
- RFC 1006

Unterschieden wird die Kommunikation zwischen folgenden Teilnehmern:

- Virtual Controller ⇔ Externer Partner
- Virtual Controller ⇔ Virtual Controller innerhalb einer SIMIT-Instanz

Verbindungskonfiguration in SIMIT

Verbindungskonfigurationen werden aus dem importierten STEP 7-/PCS 7-Projekt übernommen. Konfigurationsänderungen einer Verbindung sind ausschließlich im STEP 7-/PCS 7-Projekt möglich. In SIMIT können Sie im Verteilungseditor die Verwendung von IP-Adressen eines Virtual Controller einzeln aktivieren und deaktivieren.

Kommunikation mit einem externen Partner

Als "externer Partner" gelten folgende Teilnehmer:

- Beliebiges anderes Gerät, z. B. reale PLC
- OS-Server, z. B. WinCC
- Virtual Controller, die im Kontext einer anderen SIMIT-Instanz laufen (z. B. bei verteilter Simulation)

Hinweis

Bitte beachten Sie die Einschränkungen, die sich beim Einsatz von Ersatzfunktionen in Virtual Controllern ergeben können.

Kommunikation zwischen zwei Virtual Controllern innerhalb einer SIMIT-Instanz

Alle IP-basierten Verbindungen zwischen zwei Virtual Controllern werden unterstützt.

Open User Communication (TSEND / TRECV)

Im Rahmen der Open User Communication (TSEND / TRECV) werden standardmäßig die IP-Adressen und Ports verwendet, die in den Datenbausteinen des S7-Programms hinterlegt sind. Diese IP-Adressen müssen auf den PCs in den Netzwerkeinstellungen eingetragen und die Portnummern frei sein.

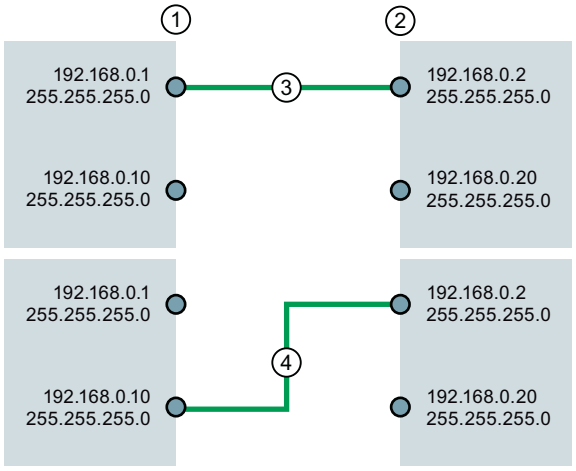
Wenn Sie die Open User Communication ausschließlich zwischen Virtual Controllern nutzen, können Sie simulationsinterne Adressierung verwenden.

Die im S7-Programm hinterlegten IP-Adressen müssen Sie in diesem Fall nicht in den Netzwerkeinstellungen der PCs eintragen. Konflikte mit belegten Portnummern werden vermieden. Diese Art der Adressierung definieren Sie für das gesamte Projekt einmalig und für jedes Protokoll getrennt im Verteilungseditor (Seite 133).

Virtual Controller mit mehreren IP-Schnittstellen im selben Subnetz

Eine projektierte Verbindung besteht immer aus einer lokalen IP-Adresse und der IP-Adresse des Kommunikationspartners. Die lokale IP-Adresse lässt sich nicht immer eindeutig aus der Verbindungsprojektierung bestimmen. SIMIT prüft daher, ob zur IP-Adresse des Kommunikationspartners eine lokale IP-Adresse im selben Subnetz existiert, und ordnet diese der Verbindung zu. Falls es mehrere lokale IP-Adressen in diesem Subnetz gibt, verwendet SIMIT die numerisch kleinste lokale IP-Adresse. Bei UDP-Verbindungen kann die lokale IP-Adresse unter Umständen auch gar nicht ermittelt werden (Multicast, freie Verbindung) und bleibt dann leer. Dadurch werden Verbindungen möglicherweise nicht oder falsch zugeordnet.

Die folgende Abbildung zeigt schematisch, wie SIMIT beim Import mit mehreren IP-Schnittstellen im selben Subnetz umgeht:



① Die lokalen IP-Adressen können systembedingt nicht ausgelesen werden. Diese Adressen müssen Sie in den Verbindungskonfigurationen möglicherweise manuell korrigieren.

② IP-Adressen des Verbindungspartners können ausgelesen werden.

③ Ergebnis nach Import: Verbindung wird korrekt zugeordnet.
Annahme: Alle Verbindungen zu externen Partnern gehen von der niedrigsten IP-Adresse aus.

④ Ergebnis nach Import: Verbindung wird nicht korrekt umgesetzt.
Auf Grundlage der Annahme aus ③ trägt SIMIT beim Import als lokale Adresse "192.168.0.1" ein.

Um falsch oder gar nicht zugeordnete Verbindungen zu korrigieren, gehen Sie folgendermaßen vor:

1. Benennen Sie im SIMIT-Projektordner unter "\io\VCROOT\VC<Instanznummer>" die Datei "HWConnection_Override.xml.default" in "HWConnection_Override.xml" um.
   In der Datei "HWConnection_Override.xml.default" werden die Verbindungsinformationen gespeichert. Diese Datei wird bei jedem Import neu erstellt.
2. Korrigieren oder ergänzen Sie in der Datei "HWConnection_Override.xml" die lokalen IP-Adressen der Verbindungen.
3. Ändern Sie bei Bedarf für jede Verbindung zwischen zwei Virtual Controllern den Parameter "Fast Connection" von "slow" auf "fast".

Siehe auch

Virtual Controller konfigurieren (Seite 133)

Unterstützte Systemfunktionen (Seite 117)

Unterstützte S7-Bausteine (Seite 119)

Unterstützte Dienste (Seite 120)

2.6.1.5 Unterstützte Systemfunktionen

Der Virtual Controller läuft im Unterschied zu einer realen Steuerung auf einer PC-Hardware und unter dem Betriebssystem Windows. Daher kann nicht erwartet werden, dass alle Eigenschaften, Funktionen und Verhaltensweisen einer realen Steuerung vom Virtual Controller nachgebildet werden können.

Es werden nur die Standardfunktionalitäten, z. B. einer S7-417-5H, unterstützt. Für Erweiterungen und Zusatzfunktionen einer AS 410-5H bilden Virtual Controller die Funktionalität einer S7-400 CPU 417-5H nach.

Ersatzfunktionen in Virtual Controllern

Automatisierungsprogramme bestehen aus MC7-Code, der vom Virtual Controller interpretiert werden kann und Systemfunktionen (SFC/SFB), die Teil der Firmware einer Steuerung sind und auf dem PC nicht im Original abgearbeitet werden können. Für die wichtigsten Systemfunktionen verfügt der Virtual Controller über eine angepasste Ersatzimplementierung:

Typ	Blocknummer	Kommentar
SFB	0–5, 8, 9, 12–15, 22, 23, 31, 33–36	
SFB	54	
SFC	0	Die Virtual Controller werden intern immer auf die Rechnerzeit synchronisiert, SET_CLK hat daher nur einen temporären Effekt
SFC	1–6, 13–15, 17–34, 36–44, 46–50	
SFC	51	SZLs werden vom Virtual Controller eingeschränkt unterstützt, da keine Simulation der dezentralen Peripherie existiert
SFC	60, 62, 64, 79, 80, 81, 85[1], 87, 90, 107	
SFC	131, 132, 133, 134, 135, 136	T-Kommunikation

[1] Nur für S7-300

Hinweis

- Die Verwendung von SFCs ist prinzipbedingt eingeschränkt, da z. B. die Peripherie nicht simuliert wird.
- Überprüfen Sie vor dem Einsatz von Virtual Controllern, ob die von Ihrem Automatisierungsprogramm benötigten SFBs und SFCs unterstützt werden. Vergleichen Sie hierzu z. B. im SIMATIC Manager die Referenzdaten der Bausteine. Weitere Informationen zu den Referenzdaten finden Sie in der Hilfe zu STEP 7.
- Die Ersatzfunktionen liefern verwertbare Rückmeldungen für den Standardbetrieb ohne Störung der Hardware, soweit es mit der Implementierungstiefe und den Informationen aus dem SIMATIC-Projekt möglich ist. Aussagen zum Verhalten im Störfall sind nur sehr bedingt möglich und müssen vom Anwender unabhängig überprüft werden.
- Anwenderbausteine und Bibliothekskomponenten (FB, FC) können unter Umständen auf Systemfunktionen zugreifen, die vom Virtual Controller nicht unterstützt werden. In diesem Fall wird der gesamte Baustein ignoriert (NoOperation).
- Wenn das Automatisierungsprogramm Signale von diesen Systemfunktionen erwartet, können Sie den entsprechenden (Instanz-)DB in SIMIT adressierbar machen und die erwarteten Informationen für SIMIT direkt in den entsprechenden DB schreiben.
- Der Virtual Controller unterstützt die Bibliothek "S7 F Systems Lib V1_3" und "Safety Matrix" im fehlerfreien Betrieb. Fehlerzustände sind nicht abbildbar. F-Kommunikation ist nur zwischen Virtual Controllern innerhalb einer SIMIT-Instanz möglich.
- "Distributed Safety" wird nicht unterstützt.

2.6.1.6 Unterstützte S7-Bausteine

Die folgende Tabelle zeigt die unterstützten S7-Bausteine abhängig vom verwendeten Protokoll und den Kommunikationspartnern:

Kommunikation zwischen:	Virtual Controller-Virtual Controller				Virtual Controller-Externer Partner			
S7-Baustein	S7	TCP	UDP	ISO-on-TCP	S7	TCP	UDP	ISO-on-TCP
FB12 BSEND	x	–	–	–	x	–	–	–
FB13 BRCV	x	–	–	–	x	–	–	–
FB14 GET	x	–	–	–	x	–	–	–
FB15 PUT	x	–	–	–	x	–	–	–
FB8 USEND	x	–	–	–	x	–	–	–
FB9 URCV	x	–	–	–	x	–	–	–
FC5 AG_SEND	–	x	x	x	–	x	x	x
FC50 AG_LSEND	–	x	x	x	–	x	x	x
FC53 AG_SSEND	–	x	x	x	–	x	x	x
FC6 AG_RECV	–	x	x	x	–	x	x	x
FC60 AG_LRECV	–	x	x	x	–	x	x	x
FC63 AG_SRECV	–	x	x	x	–	x	x	x
SFB12 BSEND	x	–	–	–	x	–	–	–
SFB13 BRCV	x	–	–	–	x	–	–	–

Kommunikation zwischen:	Virtual Controller-Virtual Controller				Virtual Controller-Externer Partner			
S7-Baustein	S7	TCP	UDP	ISO-on-TCP	S7	TCP	UDP	ISO-on-TCP
SFB14 GET	x	–	–	–	x	–	–	–
SFB15 PUT	x	–	–	–	x	–	–	–
SFB22 STATUS	x	–	–	–	–	–	–	–
SFB23 USTATUS	x	–	–	–	–	–	–	–
SFB31 NOTI-FY_8P	–	–	–	–	x	–	–	–
SFB33 ALARM	–	–	–	–	x	–	–	–
SFB34 ALARM_8	–	–	–	–	x	–	–	–
SFB35 ALARM_8P	–	–	–	–	x	–	–	–
SFB36 NOTIFY	–	–	–	–	x	–	–	–
SFB8 USEND	x	–	–	–	x	–	–	–
SFB9 URCV	x	–	–	–	x	–	–	–
SFC107 ALARM_DQ	–	–	–	–	x	–	–	–
SFC108 ALARM_D	–	–	–	–	x	–	–	–
SFC17 ALARM_SQ	–	–	–	–	x	–	–	–
SFC18 ALARM_S	–	–	–	–	x	–	–	–
SFC19 ALARM_SC	–	–	–	–	x	–	–	–
SFC131 T_SEND	–	x	–	x	–	x	–	x
SFC132 T_RCV	–	x	–	x	–	x	–	x
SFC133 T_CON	–	x	x	x	–	x	x	x
SFC134 T_DIS-CON	–	x	x	x	–	x	x	x
SFC135 T_USEND	–	–	x	–	–	–	x	–
SFC136 T_URCV	–	–	x	–	–	–	x	–

2.6.1.7 Unterstützte Dienste

Einleitung

Reale Steuerungen stellen umfassende Dienste zur Verfügung, die den Zugriff auf die Steuerung erlauben.

SIMIT VC unterstützt die nachfolgenden Kommunikationsdienste, die für eine Kommunikation zwischen virtuellen Steuerungen und PCS 7 OS/WinCC-Servern oder PCS 7 OS/WinCC-

Clients benötigt werden. Diese Dienste stellen Antworten auf folgende Anfragen zur Verfügung, die über die S7_DOS-Schnittstelle von PCS 7 OS/WinCC gestellt werden können.

Hinweis

Andere Anfragen werden von der virtuellen Steuerung nicht beantwortet. Deshalb können Systeme, die auf weitergehende Dienste angewiesen sind, nicht korrekt mit den emulierten Steuerungen kommunizieren. Dazu gehören z. B.:

- BRAUMAT Classic
- AS based Batch

Unterstützte Dienste von PCS 7 OS / WinCC

- VFD / Virtual Device Services
  - Read SZL (wobei nur der Umfang von SZL IDs implementiert ist, der zur Unterstützung der PCS 7 OS/WinCC-Kommunikation erforderlich ist)
- BuB Services
  - Cyclic Read Variables (Start, Stop, Change, Abort, Delete)
  - Read Variables
  - Write Variables
- Message Services
  - Acknowledge
  - Announcement for Messages (nicht SCAN, LT-Sammelmeldung und Archive)
  - Lock / Unlock Messages (nicht SCAN, LT-Sammelmeldung und Archive)
  - Message Update (nicht SCAN, LT-Sammelmeldung und Archive)
- PBK Services
  - USEND / URCV
  - BSEND / BRCV
  - PUT (Write Variables)
  - GET (Read Variables)

Unterstützte S7-Dienste

- SPS7
  - Einrichten Applikationsbeziehung
  - Abbruch Applikationsbeziehung
  - Status_Virtuelles_Gerät
  - DataExchange2 Cancel
- Bedienen und Beobachten
  - Lesen (Variable / segmentiert)
  - Schreiben (Variable / segmentiert)
  - Zyklisches_Lesen_Einrichten
  - Änderungsgesteuertes_Zyklisches_Lesen_Einrichten
  - Auftrag_starten
  - Auftrag_stoppen
  - Auftrag_löschen
  - Ändern_Zyklisches_Lesen ohne Delta-Auswertung
  - Ändern_Zyklisches_Lesen mit Delta-Auswertung
- Diagnose
  - An_Abmelden
  - Quittierung
  - Quittierung_an_alle
  - Sperren_Meldungen
  - Freigeben_Meldungen
  - Sperren_an_alle
  - Freigeben_an_alle
  - Update
  - Update_anfordern_an_alle
  - SZL_Lesen (weiterleiten auf SoftPLC-Onlineschnittstelle)

- Programmierbare Bausteinkommunikation
  - ALARM_MELD
  - ALARM_8_MELD
  - ALARM_SQ / ALARM_DQ
  - ALARM_S / ALARM_D
  - ALARM_MELD_QTM
  - ALARM_MELD_8_QTM
  - NOTIFY_8_ALARM_MELD
  - USEND / URCV
  - BSEND / BRCV
  - GET
  - PUT
- Test und Inbetriebsetzung / Objektverwaltung
  - Weiterleitung auf Online-Schnittstelle

Unterstützte Dienste für ein Programmiergerät (PG)

- Laden
- Delta-Laden
- Online-Monitoring (Werte)
- Diagnosepuffer lesen

Hinweis

Forcen wird nicht unterstützt.

2.6.1.8 Behandlung von Synchronfehlern

Nicht alle Synchronfehler werden erkannt bzw. richtig behandelt (Aufrufen eines Fehler OBs). Eventuell wird der Fehler nicht erkannt, erzeugt aber einen Runtime Error (RTE).

Die Fehlerbehandlung des Virtual Controller im Überblick:

Fehlerbeschreibung	unterstützt
BCD Wandlungsfehler	Ja
Bereichslängenfehler beim Lesen	Nein, erzeugt RTE
Bereichslängenfehler beim Schreiben	Nein, erzeugt RTE
Bereichsfehler beim Lesen	Ja
Bereichsfehler beim Schreiben	ja
Timernummernfehler	Nein, erzeugt RTE
Zählernummernfehler	Nein, erzeugt RTE

Fehlerbeschreibung	unterstützt
Ausrichtungsfehler beim Lesen	Ja
Ausrichtungsfehler beim Schreiben	Ja
Schreibfehler Datenbaustein	Nein, es wird in den schreibgeschützten Baustein geschrieben
Schreibfehler Instanzdatenbaustein	Nein, es wird in den schreibgeschützten Baustein geschrieben
Bausteinnummernfehler DB	Nein, erzeugt RTE
Bausteinnummernfehler DI	Nein, erzeugt RTE
Bausteinnummernfehler FC	Nein, erzeugt RTE
Bausteinnummernfehler FB	Nein, erzeugt RTE
DB nicht geladen	Ja
FC nicht geladen	Ja
FB nicht geladen	Nein, erzeugt RTE

2.6.1.9 Ablaufsteuerung

Die folgenden Organisationsbausteine einer realen Steuerung werden unterstützt:

Von SIMIT VC unterstützt	Bausteintyp	Baustein	Beschreibung und Ereignis	Priorität
1	Anwenderprogramm	OB1	Bearbeitung des Anwenderprogramms nach Wiederanlauf (Ende von OB100) und am Zyklusende.	1
10–17	Uhrzeitalarm	OB10 OB11 OB12 OB13 OB14 OB15 OB16 OB17	Uhrzeit und Datum. Zu einem bestimmten Zeitpunkt kann angetriggert werden, dass ein Uhrzeit OB (10–17) aufgerufen wird, der dann ein Programm abarbeitet. Wird auch Uhrzeitalarm OB genannt.	2
20–23	Verzögerungsalarm	OB20 OB21 OB22 OB23	Verzögerungsalarme. Nach einer Verzögerungszeit wird der OB (20–23) aufgerufen und das Programm wird abgearbeitet.	3 4 5 6
30–38	Weckalarm	OB30 OB31 OB32 OB33 OB34 OB35 OB36 OB37 OB38	Weckalarme, starten periodisch nach einer definierten Zeit. Ähnlich wie Taktsignale, jedoch wesentlich genauer. Die Bearbeitung des OB1 wird unterbrochen, da sie eine höhere Priorität haben.	7 8 9 10 11 12 13 14 15

–	Prozessalarm	OB40 OB41 OB42 OB43 OB44 OB45 OB46 OB47	Prozessalarme. Reagieren an alarmfähigen Eingangs-, Ausgangs- oder Funktionsbaugruppen auf parametrierte Ereignisse, z. B. positive Flanke, Grenzwertüberschreitung. Werden z. B. eingesetzt, wenn die Reaktionszeit im Programm zu lang ist.	16 17 18 19 20 21 22 23
–	DPV1-Alarm	OB55 OB56 OB57	DPV1-Alarme. In Verbindung mit DPV1-Slaves werden Status-, Update- oder herstellerspezifische Alarme ausgelöst.	2
–	Multicomputingalarm	OB60	Synchroner Betrieb mehrerer CPUs	25
–	Taktsynchronalarm	OB61 OB62 OB63 OB64	Kurze und gleichlange Prozessreaktionszeiten am PROFIBUS DP projektieren	25
–	Hintergrundzyklus	OB90	Für die Programmbearbeitung im Hintergrund	29
100–102	Anlauf	OB100 OB101 OB102	Nach Neustart (Warmstart) der CPU Nach Wiederanlauf der CPU Nach Kaltstart der CPU	27
121, nicht 122	Synchrone Fehler	OB121 OB122	Wenn ein Baugruppenfehler auftritt	29

Die Ablaufsteuerung von SIMIT unterbricht bereits laufende OBs nicht. Sie werden nach ihrer Priorität in den Grundzyklus von SIMIT eingereiht.
Die folgende Abbildung hat einen Grundzyklus von 100 ms:

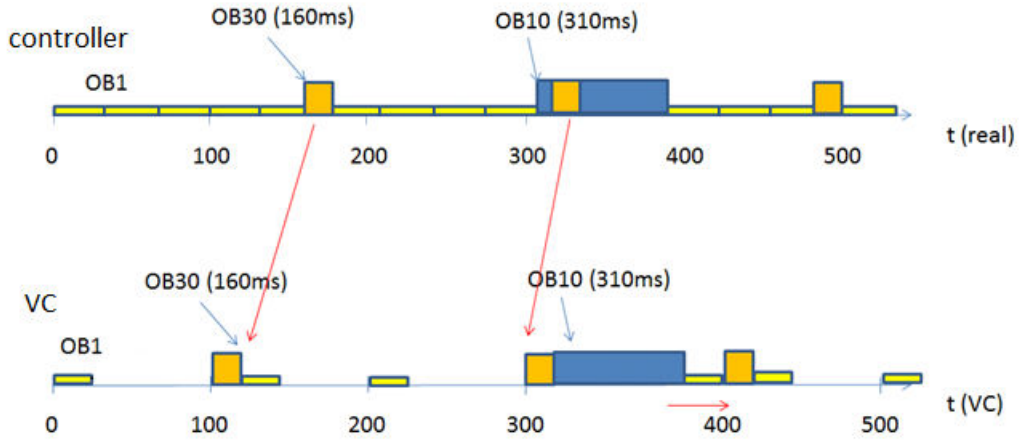


Bild 2-1 Ablaufsteuerung

Somit ergibt sich folgendes Zeitverhalten:

In jedem Zyklus werden die anstehenden OBs gemäß ihrer Priorität nacheinander und ohne Verzögerungen ausgeführt. Eine Zyklusüberschreitung findet in der Abbildung oben beim Wert "400" statt. Daraufhin wird sich der Gesamtzyklus vergrößern und der nächste Grundzyklus

damit verzögert gestartet. Im darauf folgenden Zyklus korrigiert SIMIT intern alle Zeiten (wie in der Abbildung oben dargestellt) von "400+x" zurück auf "400" und rechnet damit weiter, auch wenn die reale Zeit schon bei "400+x" steht. So wird die virtuelle Simulationszeit konsistent gehalten.

Hinweis

SIMIT unterscheidet zwischen Timerzeiten und der Systemzeit, die z. B. für das Versenden von Meldungen verwendet wird. Die Systemzeit wird unabhängig vom Fortschritt der Timerzeiten regelmäßig auf die Systemzeit des SIMIT-Rechners synchronisiert.

2.6.1.10 Datensatzkommunikation

Der Virtual Controller unterstützt Datensatzkommunikation über folgende SFCs:

- WR_REC (SFC 58)
- RD_REC (SFC 59)
- WRREC (SFB 53)
- RDREC (SFB 52)
- DPNRM_DG (SFC13)

Diese Bausteine benötigen ein entsprechendes Gerätemodell in SIMIT, welches die Datensätze bereitstellen kann.

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

2.6.1.11 Ersatzmechanismus für Datenaustausch

Über PROFIBUS oder PROFINET angebundene Signale sind über das Prozessabbild direkt verfügbar. Alle anderen Signalanbindungen wie Modbus oder FF benötigen einen Ersatzmechanismus.

Hinweis

Ersatzmechanismen können Sie im Kopplungseditor der Virtual Controller-Kopplung in der Registerkarte "DB" eintragen. Weitere Informationen hierzu finden Sie im Abschnitt: Eigenschaften der Virtual Controller-Kopplung (Seite 140).

Siehe auch

Unterstützte Systemfunktionen (Seite 117)

2.6.1.12 Hinweise zum Einsatz eines Virtual Controller

Erstellen von Schnappschüssen

Bei laufender Simulation können von den Virtual Controllern Schnappschüsse angelegt werden. Die Schnappschüsse werden im SIMIT-Projektverzeichnis abgelegt und damit Bestandteil eines SIMIT-Projektarchivs. Schnappschüsse bleiben auch bei Änderungen am SIMIT-Projekt unverändert erhalten.

Folgende Informationen werden in Schnappschüssen nicht berücksichtigt:

- Die Zustände angeschlossener Systeme
- Die Kommunikation zwischen Virtual Controllern
- Die Kommunikation zwischen Virtual Controllern und angeschlossenen Systemen

---

Hinweis

Schnappschüsse werden einem Virtual Controller über dessen Namen zugeordnet.

Wenn Sie den Namen eines Virtual Controller nach dem Erstellen eines Schnappschusses in der Projektnavigation ändern, kann der Schnappschuss nicht mehr zu diesem Virtual Controller zugeordnet werden. Benennen Sie nach dem Erstellen eines Schnappschusses den Virtual Controller nicht in der Projektnavigation um.

Hinweis

Schnappschüsse bei Änderungen der Simulation

Schnappschüsse speichern die Zustände von Modell und einzelnen Virtual Controllern.

- Beachten Sie, dass das Mapping der Daten im Schnappschuss über Namen (Signalnamen, DB-Nummern, etc.) erfolgt. Deshalb kann nicht garantiert werden, dass ein Schnappschuss nach Änderungen an der Simulation (Modell, Automatisierungsprogramm) zu einem sinnvollen Zustand der Simulation bzw. des Virtual Controller führt. Ob die Änderung durch den Anwender oder das Anwenderprogramm (z. B. durch den Einsatz von Systemfunktionen wie CREAT_DB) verursacht wird, ist dabei unerheblich.
- Beachten Sie insbesondere, dass nach dem Umbenennen eines Virtual Controller eventuell bereits vorhandene Schnappschüsse nicht mehr korrekt zugeordnet werden können.

2.6.2 Konfigurieren des Virtual Controller

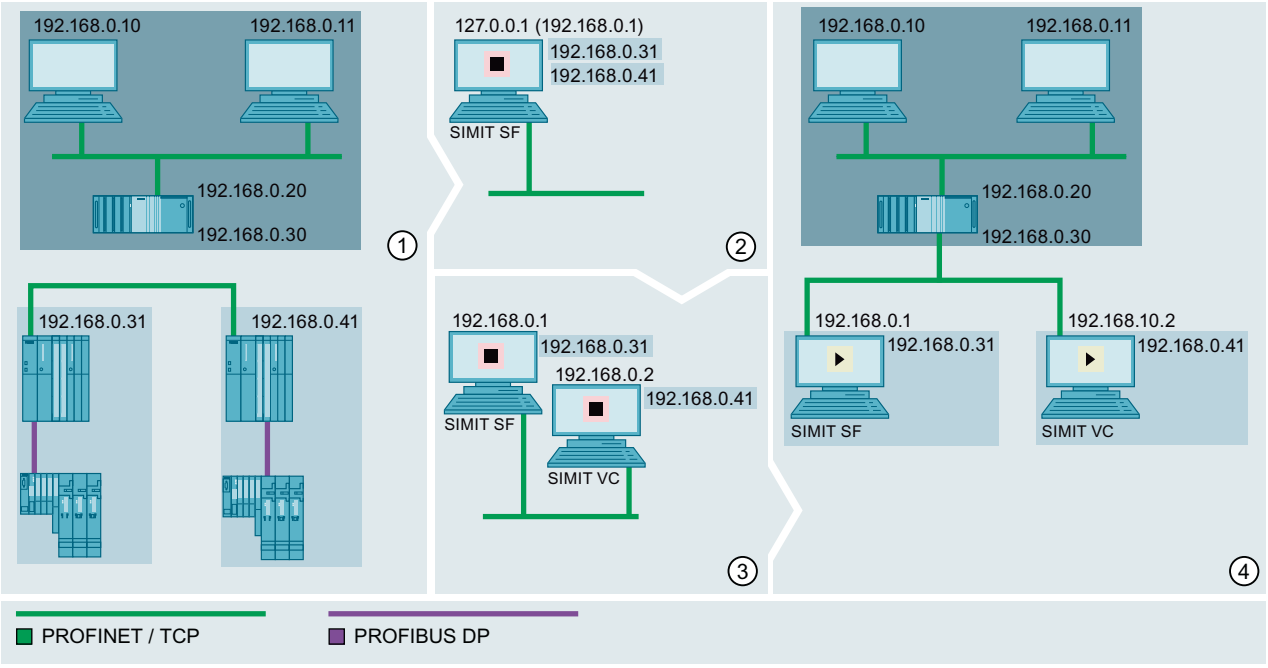
2.6.2.1 Virtual Controller projektieren

Voraussetzung

- IP-Adressen der Teilnehmer am Anlagenbus und Terminalbus sind bekannt.
- IP-Adressen im Firmen-/Simulationsnetzwerk sind bekannt.
- IP-Adressen sind alle eindeutig.

Handlungsübersicht

Die folgende Abbildung zeigt das Vorgehen beim Erstellen einer Virtual Controller-Kopplung:



Automatisierungsebene mit ES, OS-Server und zwei OS-Clients

Steuerungsebene mit PLC und dezentraler Peripherie.

Simulation im Zustand "Stopp"

Simulation im Zustand "Gestartet"

① Simulation vorbereiten

1. Stationen definieren, die simuliert werden sollen.
2. Steuerungsebene gegebenenfalls vom Netzwerk trennen.

② SIMIT-Projekt erstellen:

1. Virtual Controller-Kopplung anlegen.
2. Stationen als Virtual Controller aus STEP 7-/PCS 7-Projekt importieren.
   Der SIMIT-PC erhält im SIMIT-Projekt die IP-Adresse 127.0.0.1. Im Netzwerk ist der SIMIT-PC weiterhin unter seiner ursprünglichen IP-Adresse erreichbar, z. B. 192.168.0.1.
3. IP-Adressen der Virtual Controller in den Netzwerkeinstellungen des SIMIT-PC eintragen.
4. Simulationsmodell erstellen.

③ Zusätzlichen PC einrichten (optional)

1. IP-Adresse des SIMIT-PC im Verteilungseditor auf die im Netzwerk sichtbare IP-Adresse ändern.
2. PC in Netzwerk integrieren.
3. Virtual Controller auf PCs verteilen.
4. IP-Adressen in den Netzwerkeinstellungen der PCs anpassen.

④ Simulation starten.

Siehe auch

Virtual Controller anlegen (Seite 130)

HWCNExport-Datei erstellen (Seite 65)

STEP 7-/PCS 7-Projekt importieren (Seite 131)

Zusätzlichen PC konfigurieren (Seite 137)

Eigenschaften der Virtual Controller-Kopplung (Seite 140)

Anforderungen an ein Simulationsnetzwerk (Seite 113)

2.6.2.2 Virtual Controller anlegen

Einleitung

In einem SIMIT-Projekt können Sie eine Virtual Controller-Kopplung anlegen. Unterhalb einer Virtual Controller-Kopplung können Sie Stationen aus einem oder mehreren STEP 7-/PCS 7-Projekten importieren.

Hinweis

Unterstützte STEP 7-Version

Die Virtual Controller-Kopplung wird bis einschließlich STEP 7 V5.5 unterstützt.

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- Im Projekt ist keine Virtual Controller-Kopplung angelegt.
- STEP 7-/PCS 7-Projekt oder HWCNExport-Datei liegt vor.

Vorgehen

Um eine Virtual Controller-Kopplung anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Kopplungen" den Befehl "Neue Kopplung".
   Das Dialogfeld "Auswahl" wird geöffnet.
2. Aktivieren Sie den Kopplungstyp "Virtual Controller".
   Das Dialogfeld "Virtual Controller Import" wird geöffnet.
3. Konfigurieren Sie den Import von Stationen aus einem STEP 7-/PCS 7-Projekt oder der HWCNExport-Datei.

Hinweis

Nach Anlegen eines Virtual Controllers muss die komplette Station in den laufenden Virtual Controller geladen werden, da der Virtual Controller während der Laufzeit Informationen aus Systemdatenbausteinen liest.

Wenn Änderungen an der Konfiguration des Controllers vorgenommen werden, muss die komplette Station erneut geladen werden.

Ergebnis

Das Ergebnis nach dem Import eines STEP 7-/PCS 7-Projekts finden Sie unter STEP 7-/PCS 7-Projekt importieren (Seite 131).

Wenn Sie die Kopplung vom Typ "Virtual Controller" ohne Import eines STEP 7-/PCS 7-Projekts abschließen, wird in der Projektnavigation unter Kopplungen nur der Ordner "Virtual Controller" angelegt.

Siehe auch

Dialogfeld "Virtual Controller Import" (Seite 964)

Dialogfeld "Auswahl" (Seite 980)

Virtual Controller projektieren (Seite 128)

2.6.2.3 STEP 7-/PCS 7-Projekt importieren

Einleitung

Damit Sie mit einer Virtual Controller-Kopplung eine Simulation erstellen können, importieren Sie die Stationen aus einem STEP 7-/PCS 7-Projekt. Folgende Informationen aus dem STEP 7-/PCS 7-Projekt werden übernommen:

- CPU
- Zentrale E/A-Baugruppen
- Dezentrale E/A-Baugruppen
- Verbindungen

Sie können den Import jederzeit erneut ausführen, z. B. um Projektierungsdaten zu aktualisieren oder neue Stationen hinzuzufügen. Identische Stationen werden anhand folgender Kriterien identifiziert:

- Projektname
- Stationsname
- Speicherort im STEP 7-/PCS 7-Projekt

Optional können Sie während der Konfiguration des Imports die zu importierenden Stationen auswählen.

Hinweis
Unterstützte STEP 7-Version

Die Virtual Controller-Kopplung wird bis einschließlich STEP 7 V5.5 unterstützt.

Voraussetzung

- Kopplung vom Typ "Virtual Controller" ist angelegt.
- SIMATIC Manager ist installiert oder HWCNExport-Datei liegt vor.

Vorgehen

Um ein STEP 7-/PCS 7-Projekt zu importieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Kopplungen > Virtual Controller" den Befehl "Importieren".
   Das Dialogfeld "Virtual Controller Import" wird geöffnet.
2. Wählen Sie das gewünschte STEP 7-/PCS 7-Projekt oder die HWCNExport-Datei.
3. Wählen Sie den Modus für den Import der Symbolik.
4. Wählen Sie die gewünschten Stationen aus.
5. Klicken Sie auf "Importieren".

Ergebnis

Für jede Station aus dem STEP 7-/PCS 7-Projekt wird in der Projektnavigation unter "Kopplungen > Virtual Controller" ein Virtual Controller angelegt. Beim ersten Import werden die importierten Virtual Controller im Editor "Verteilung" dem Emulations-PC "Eigener Computer" zugeordnet.

Siehe auch

HWCNExport-Datei erstellen (Seite 65)

Dialogfeld "Virtual Controller Import" (Seite 964)

Virtual Controller projektieren (Seite 128)

Signale einer Kopplung bearbeiten (Seite 188)

2.6.2.4 Virtual Controller konfigurieren

Voraussetzung

- STEP 7-/PCS 7-Projekt ist importiert.
- Virtual Controller-Kopplung ist angelegt.

Vorgehen

Um einen Virtual Controller zu konfigurieren, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in der Projektnavigation unter "Kopplungen > Virtual Controller" auf den gewünschten Virtual Controller.
2. Konfigurieren Sie den Virtual Controller im Eigenschaftsfenster:
   - Wählen Sie die "Zeitscheibe" aus.
   - Geben Sie bei Bedarf die Seriennummer von "CPU" und/oder Speicherkarte ein.
   - Passen Sie bei Bedarf die Zykluszeiten der OBs an.
3. Importieren Sie bei Bedarf ausgewählte Adressbereiche eines oder mehrerer Datenbausteine.

Siehe auch

Grundlagen zur Virtual Controller-Kommunikation (Seite 115)

Eigenschaften der Virtual Controller-Kopplung (Seite 140)

Dialogfeld "Import von Signaleigenschaften" (Seite 967)

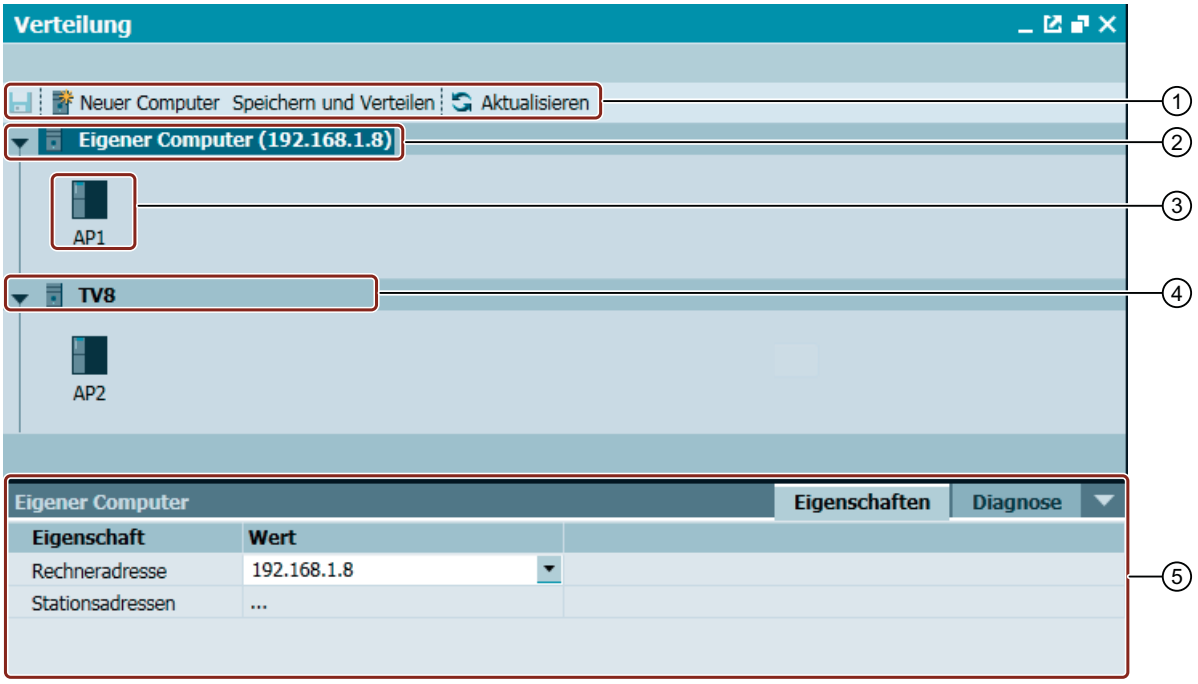
2.6.2.5 Verteilungseditor (Virtual Controller)

Verteilungseditor

Im Verteilungseditor verteilen Sie Virtual Controller auf zusätzliche PCs. Die Rechenlast zur Simulation wird damit auf mehrere PCs verteilt.

Den Verteilungseditor öffnen Sie in der Projektnavigation unter "Kopplungen > Virtual Controller" mit einem Doppelklick.

Die folgende Abbildung zeigt den Aufbau des Verteilungseditors:



① Menüzeile

② Repräsentiert den PC, auf dem das SIMIT-Projekt geöffnet ist.

③ Virtual Controller, die Sie aus einem STEP 7-/PCS 7-Projekt importiert haben. Standardmäßig sind diese Virtual Controller dem PC "Eigener Computer" zugeordnet.

④ Zusätzlicher PC (optional)

Der PC muss sich im selben Netzwerk wie der PC "Eigener Computer" befinden. Die verfügbaren Virtual Controller verteilen Sie zwischen den PCs mit Drag & Drop.

⑤ Eigenschaftsfenster

Im Eigenschaftsfenster konfigurieren Sie die Eigenschaften des selektierten PC oder Virtual Controller. Zusätzlich werden unter "Diagnose" Konflikte, wie mehrfach vergebene IP-Adressen oder nicht erreichbare PCs, angezeigt.

Menüzeile

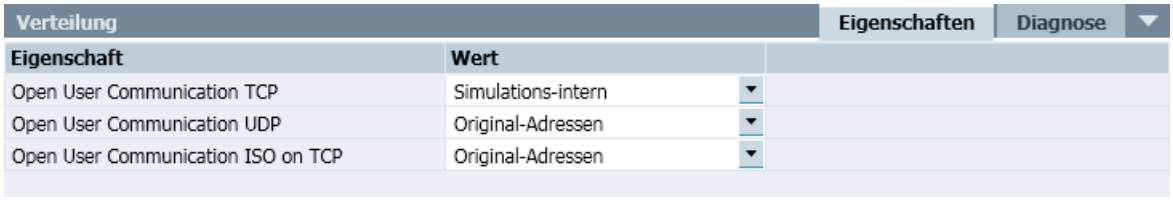
In der Menüzeile des Verteilungseditors können Sie folgende Funktionen durchführen:

- "Speichern"
  Speichert die Konfiguration.
- "Neuer Computer"
  Fügt einen neuen PC hinzu. Der PC muss sich im selben Netzwerk wie der PC "Eigener Computer" befinden.

- "Speichern und Verteilen"
  Speichert die Konfiguration und verteilt das Simulationsprojekt auf die konfigurierten PCs.
- "Aktualisieren"
  Aktualisiert die Ansicht:
  - Erreichbarkeit der konfigurierten PCs wird geprüft.
  - IP-Adressen werden auf Konflikte geprüft. Konflikte werden im Eigenschaftsfenster unter "Diagnose" angezeigt.
  - Wenn Sie einen PC entfernt haben, werden dessen Virtual Controller automatisch auf den PC "Eigener Computer" verschoben.

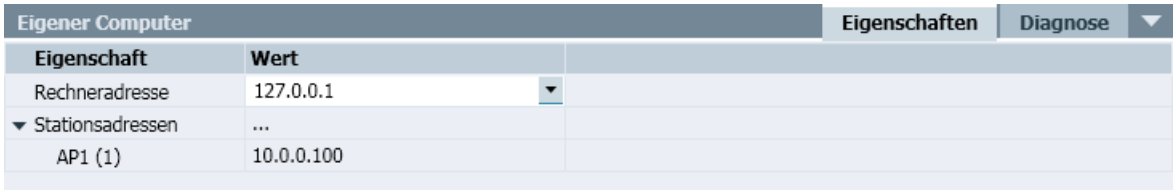
Eigenschaften von "Verteilung"

Um die Eigenschaften von "Verteilung" zu öffnen, doppelklicken Sie in der Projektnavigation auf "Verteilung".



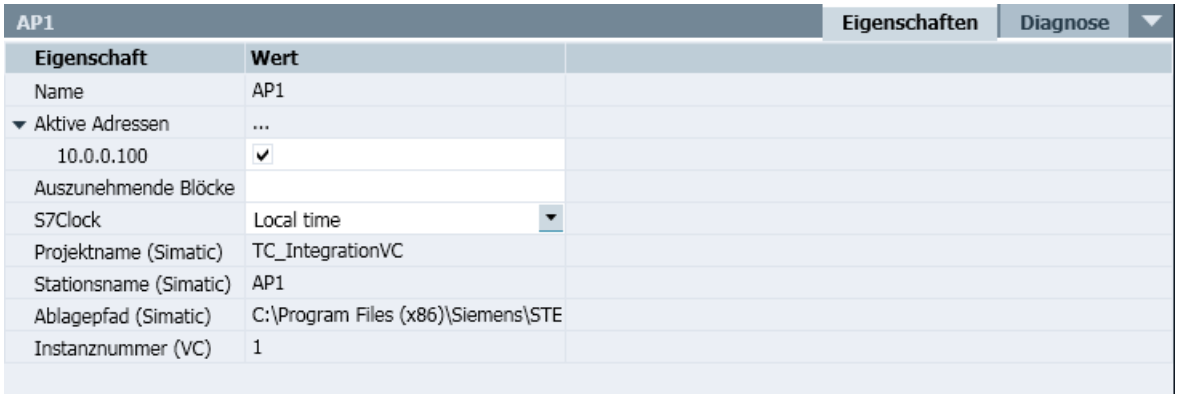
Legt für jedes Protokoll fest, welche Adressen für die Identifikation der Kommunikationspartner verwendet werden. Weiterführende Informationen finden Sie unter Grundlagen zur Virtual Controller-Kommunikation (Seite 115).

Eigenschaften von "Eigener Computer" / "Emulations-PC"



- "Rechnername"
  Legt den Rechnernamen des PCs fest, auf dem ein Virtual Controller gehostet werden soll. Alle beteiligten PCs inklusive "Eigener PC" müssen im gleichen Subnetz liegen. Solange der PC "Eigener PC" auf dem LoopBack-Adapter (127.0.0.1) eingestellt ist, sind keine anderen PCs sichtbar.
- "Rechneradresse"
  Zeigt die IP-Adresse des PCs an, unter der er aktuell im Netz bekannt ist.
- "Stationsadressen"
  Zeigt die IP-Adressen der Virtual Controller an, die diesem PC zugeordnet sind.
  Tragen Sie diese IP-Adressen in den Netzwerkeinstellungen dieses PCs ein. Falls eine IP-Adresse in der Simulation nicht verwendet werden soll, deaktivieren Sie diese IP-Adresse in den Einstellungen des Virtual Controller.

Eigenschaften eines Virtual Controller



- "Name"
  Zeigt den Namen des Virtual Controller an.
  Der Name wird aus dem STEP 7-/PCS 7-Projekt übernommen.
- "Aktive IP-Adressen"
  Zeigt die konfigurierten IP-Adressen des Virtual Controller an.
  Aufgelistet werden die IP-Adressen der PLC und in der Station gesteckten CPs. Um eine IP-Adresse in der Simulation zu verwenden, aktivieren Sie die entsprechende IP-Adresse. Wenn Sie eine IP-Adresse deaktivieren, wird eine von dieser IP-Adresse genutzte Verbindung nicht simuliert. Verbindungen zwischen Virtual Controllern werden unabhängig davon immer simuliert.
- "Auszunehmende Blöcke"
  Legt die STEP 7-Bausteine fest, die in der Simulation nicht berücksichtigt werden sollen. Trennen Sie mehrere Bausteine mit Kommas. Ungültige Einträge werden beim Verlassen des Eingabefeldes automatisch gelöscht.
- "S7 Clock"
  Legt das Uhrzeitformat fest, mit der die interne Uhr des Virtual Controller vom PC versorgt wird.
  - Lokalzeit
  - Universal Time Coordinated
- "Projektname (SIMATIC)"
  Zeigt den Namen des STEP 7-/PCS 7-Projekts an.
- "Stationsname (SIMATIC)"
  Zeigt den Stationsnamen an.
- "Ablagepfad (SIMATIC)"
  Zeigt den Ablagepfad des STEP 7-/PCS 7-Projekts an.
- "Instanznummer (Virtual Controller)"
  Zeigt die Instanznummer des Virtual Controller an.
  Die Instanznummer wird von SIMIT automatisch vergeben.

Diagnose

Beim Import eines Projekts werden die Virtual Controller geprüft.

Fehler, z. B. fehlerhafte IP-Adressen, werden farbig markiert und in der Diagnoseansicht in Klartext gemeldet.

Nach Beseitigung der Fehler verschwinden die Anzeigen in der Diagnose automatisch. Ansonsten klicken Sie im Verteilungseditor auf "Aktualisieren".

Siehe auch

Zusätzlichen PC konfigurieren (Seite 137)

2.6.2.6 Zusätzlichen PC konfigurieren

Einleitung

Bei komplexen Simulationen mit mehreren Virtual Controllern reicht die Rechenleistung eines einzelnen PC häufig nicht aus. Indem Sie zusätzliche PCs in das Simulationsnetzwerk einbinden, können Sie die Rechenlast gleichmäßig verteilen. Jedem zusätzlichen PC weisen Sie im SIMIT-Projekt dann einen oder mehrere Virtual Controller zu.

Hinweis

Zulässige Mengengerüste

Ein SIMIT-Projekt kann maximal 32 Virtual Controller enthalten.

Ein PC kann mehrere Virtual Controller simulieren. Als Faustregel gilt: Ein CPU-Kern kann ca. zwei Virtual Controller berechnen. Die tatsächlich benötigte CPU-Leistung ist abhängig von der Komplexität der Simulation.

Voraussetzung

- PC ist im Netzwerk erreichbar.
- Firewall ist eingerichtet (Seite 54).
- Auf PC ist "SIMIT VC" installiert.
- Kopplung vom Typ "Virtual Controller" ist angelegt.
- Virtual Controller aus STEP 7-/PCS 7-Projekt sind importiert.
- Editor "Verteilung" ist geöffnet.
- Simulation ist gestoppt.

Vorgehen

Um einen zusätzlichen PC zu konfigurieren, gehen Sie folgendermaßen vor:

1. Ändern Sie die IP-Adresse des PC "Eigener Computer" von "127.0.0.1" in die IP-Adresse, unter welcher der PC im Netzwerk erreichbar ist.
2. Erstellen Sie einen zusätzlichen PC:
   - Klicken Sie im Editor "Verteilung" auf "Neuer Computer".
   - Wählen Sie den Namen des PC aus.
3. Ziehen Sie den gewünschten Virtual Controller mit Drag & Drop auf den PC.
4. Aktivieren Sie im Eigenschaftsfenster des Virtual Controller unter "Eigenschaften > Aktive Adressen" die IP-Adressen, die in der Simulation verwendet werden.
5. Fügen Sie diese IP-Adressen in den Netzwerkeinstellungen des PC hinzu.
6. Klicken Sie im Editor "Verteilung" auf "Speichern und Verteilen".

Ergebnis

Die Konfiguration wird gespeichert. Das Simulationsprojekt wird auf die PCs verteilt.

Konflikte, wie mehrfach vergebene IP-Adressen oder nicht erreichbare PCs, werden im Eigenschaftsfenster unter "Diagnose" aufgelistet.

Siehe auch

Anforderungen an ein Simulationsnetzwerk (Seite 113)

Virtual Controller projektieren (Seite 128)

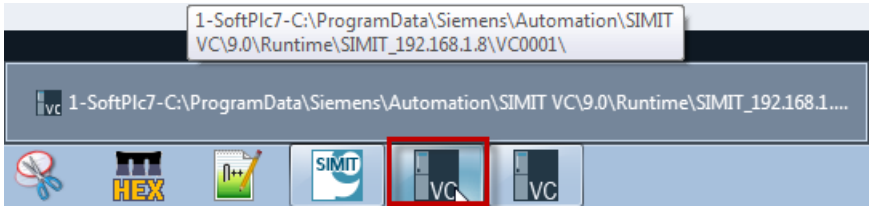
Verteilungseditor (Virtual Controller) (Seite 133)

2.6.2.7 Virtual Controller während Simulation

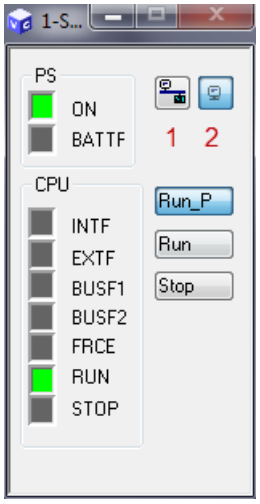
Virtual Controller Anzeige

Bei laufender Simulation wird jeder Virtual Controller als Symbol in der Taskleiste dargestellt.

Über dem Virtual Controller-Symbol wird in einem Tooltip der Pfad angezeigt, in dem die Konfigurationsdaten des Virtual Controller abgelegt sind.



Wenn Sie auf ein Virtual Controller-Symbol klicken, wird folgendes Anzeigefenster geöffnet.



Die Fenstergröße ist fest vorgegeben. Es darf nicht normal geschlossen werden.

Um das Fenster zu minimieren, klicken Sie erneut auf das Symbol auf der Taskleiste.

Das grüne Kästchen vor "RUN" zeigt an, dass der Virtual Controller im Run-Modus ist.

Ein rotes Kästchen vor "STOP" zeigt an, dass der Virtual Controller steht.

Virtual Controller Schaltflächen

- Mit der Schaltfläche über 1 schalten Sie den Virtual Controller online.
- Mit der Schaltfläche über 2 schalten Sie den Virtual Controller offline.
- Mit der Schaltfläche Run_P schalten Sie den Virtual Controller in den Run-Protect-Mode. Der Run-Protect-Mode des Virtual Conroller implementiert das Verhalten des Run-Mode.
- Mit der Schaltfläche Run schalten Sie den Virtual Controller in den Run-Mode.
- Mit der Schaltfläche Stop schalten Sie den Virtual Controller in den Stop-Mode.

Beim Beenden der Simulation werden alle Bestandteile des Simulationsprojekts auf dem Rechner zusammengeführt, auf dem SIMIT installiert ist.

Siehe auch

Dialogfeld "Virtual Controller Import" (Seite 964)

Eigenschaften der Virtual Controller-Kopplung (Seite 140)

Abbildung von SIMATIC-Datentypen in SIMIT (Seite 193)

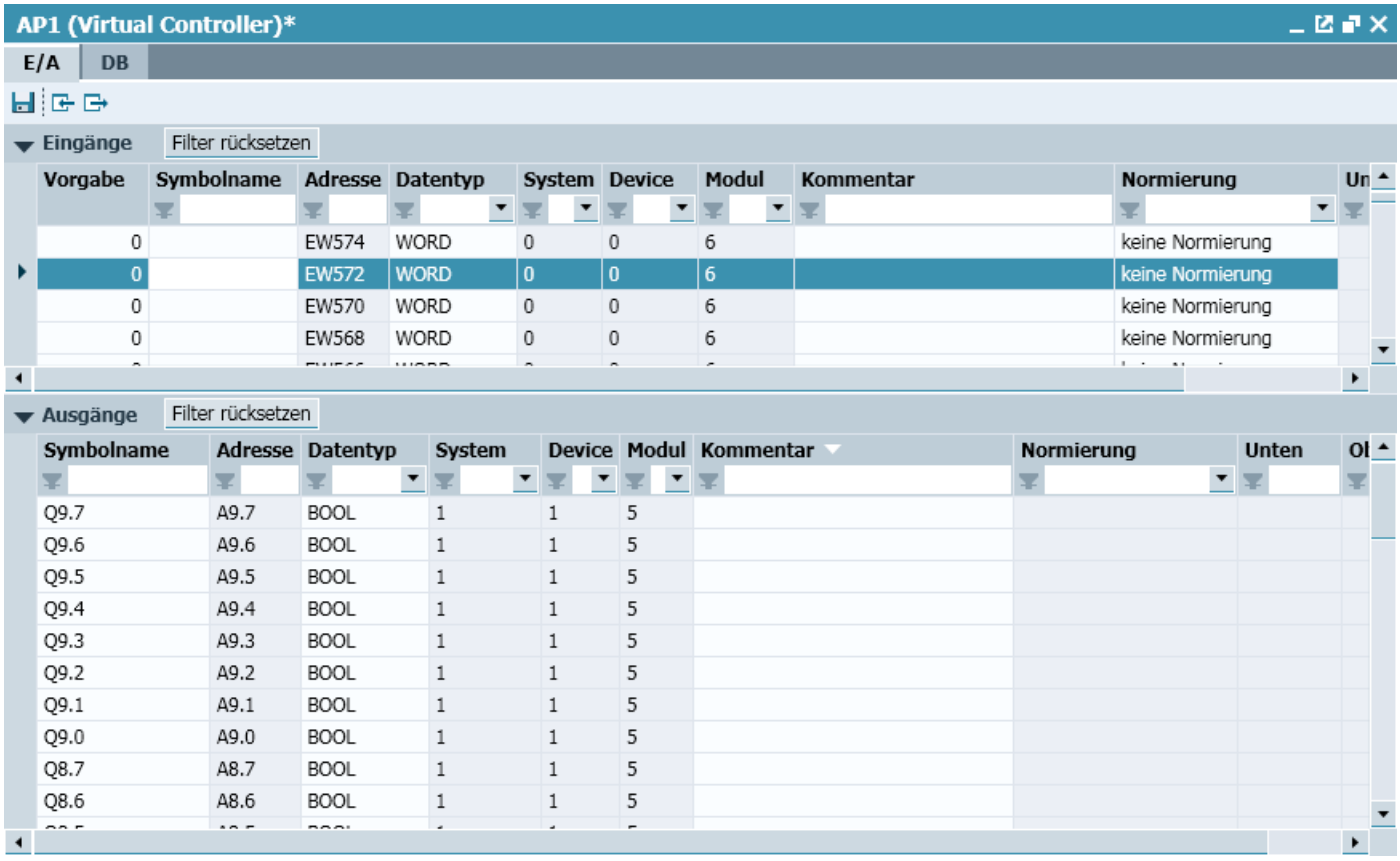
2.6.3 Editieren von Signalen

2.6.3.1 Eigenschaften der Virtual Controller-Kopplung

Eigenschaften der Adressbereiche eines Virtual Controller

Die Adressbereiche der Ein- und Ausgänge werden beim Import aus dem STEP 7-/PCS 7-Projekt übernommen.

Registerkarte "E/A"

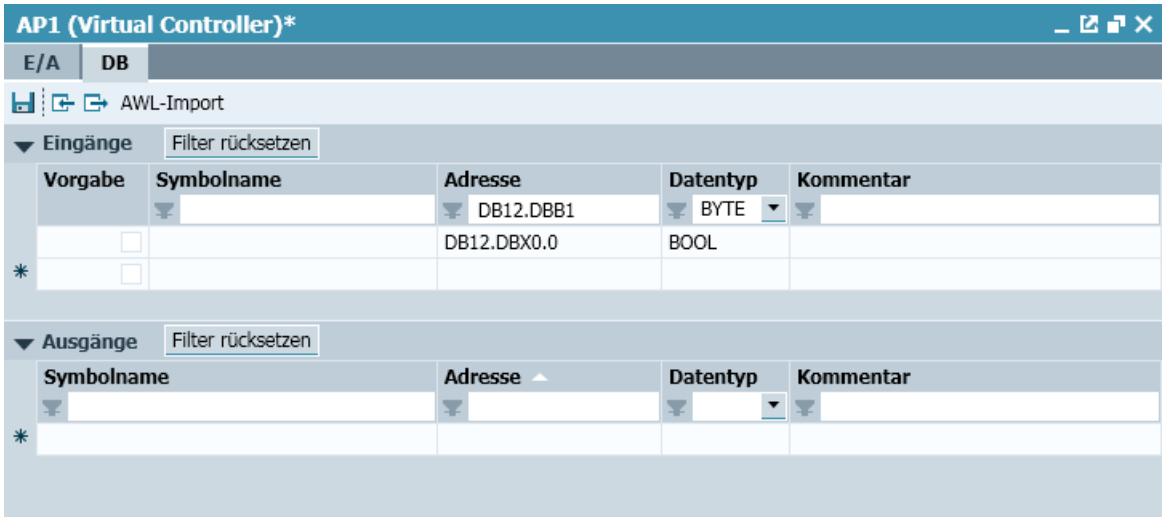


Hier werden die Ein- und Ausgänge der Kopplung aufgelistet. Weiterführende Informationen finden Sie unter Kopplungseditor (Seite 63).

Weiterführende Informationen zum Importieren von Signaleigenschaften finden Sie unter Dialogfeld "Import von Signaleigenschaften" (Seite 967).

Weiterführende Informationen zum Exportieren von Signaleigenschaften finden Sie unter Dialogfeld "Export von Signaleigenschaften" (Seite 970).

Registerkarte "DB"



Wenn Sie für die Simulation Werte von Datenbausteinen der PLC benötigen, können Sie die Adressen eingeben oder über die Schaltfläche "AWL-Import" importieren. Weiterführende Informationen finden Sie unter DB-Daten aus AWL-Quellen importieren (Seite 143).

Hinweis

Die Adressen werden nicht validiert.

Eigenschaften eines Virtual Controller

Die folgenden Eigenschaften werden beim Import aus dem STEP 7-/PCS 7-Projekt übernommen oder manuell eingetragen. Bei einem erneuten Import werden die Eigenschaften nicht überschrieben.

- "Zeitscheibe"
  Legt die Zykluszeit fest, mit welcher der OB1 ausgeführt wird.
  Die Zykluszeit jeder Zeitscheibe konfigurieren Sie zentral in den "Projekteinstellungen". Im Kontextmenü von "Projektnavigation > Kopplungen > Virtual Controller" können Sie die Zeitscheibe für alle Virtual Controller gleichzeitig setzen. Individuelle Einstellungen werden überschrieben.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- "Mnemonik"
  Legt fest, ob die deutsche ("E/A") oder englische ("I/Q") Bezeichnung der Absolutadressen verwendet wird.

  Hinweis

  Legen Sie die Mnemonik direkt nach Import fest. Wenn Sie die Mnemonik nachträglich umschalten, können bereits in einem Diagramm verschaltete Signale nicht mehr der ursprünglichen Adresse zugeordnet werden.

- "Projektname (SIMATIC)"
  Zeigt den Namen des STEP 7-/PCS 7-Projekts an.
- "Stationsname (SIMATIC)"
  Zeigt den Stationsnamen an.
- "Ablagepfad (SIMATIC)"
  Zeigt den Ablagepfad des STEP 7-/PCS 7-Projekts an.
- "Seriennummer der CPU"
  Legt die Seriennummer der CPU fest. Die Angabe ist nur notwendig, wenn das STEP 7-/PCS 7-Projekt die Seriennummer z. B. zur Lizenzierung von Programmbestandteilen benötigt.
- "Seriennummer der Speicherkarte"
  Legt die Seriennummer der Speicherkarte fest. Die Angabe ist optional.
- "Zykluszeit des OBxx"
  Legt die Zykluszeit fest.
  Die Zykluszeit wird aus dem STEP 7-/PCS 7-Projekt übernommen. OBs mit einer niedrigeren Zykluszeit als die der eingestellten Zeitscheibe werden mehrfach ausgeführt. Um Rechenlast bei der Simulation zu sparen, setzen Sie diese Zykluszeiten auf die projektierte Zykluszeit der Zeitscheibe.

Eigenschaften eines Moduls

- "Modulname"
  Zeigt den Modulnamen an.
- "MLFB"
  Zeigt die Siemens-Bestellnummer an.
- "Steckplatz des Moduls"
  Zeigt den Steckplatz des Moduls an.
- "Adressbereich Eingänge"
  Zeigt den Adressbereich der Eingänge in Byte an.
- "Adressbereich Ausgänge"
  Zeigt den Adressbereich der Ausgänge in Byte an.

Siehe auch

Virtual Controller projektieren (Seite 128)

2.6.3.2 DB-Daten aus AWL-Quellen importieren

Einleitung

Beim Import aus einer AWL-Datenquelle werden folgende Informationen nach SIMIT übernommen:

- Kommentar
- Beim Import von Eingängen: Vorbelegung der Eingänge

Folgende Datentypen werden übernommen:

- STRING (wird als BYTE-ARRAY umgesetzt)
- ARRAY
- STRUCT
- UDT
- BOOL
- BYTE
- WORD
- DWORD
- INT
- DINT
- REAL
- CHAR

Hinweis

Wählen Sie beim Generieren der AWL-Quellen in STEP 7 grundsätzlich alle verwendeten AWL-Datenquellen aus. Wenn z. B. ein Datenbaustein einen UDT enthält, fügen Sie beim Generieren der AWL-Quelle auch diese Strukturdefinition hinzu.

Folgende Datentypen werden bei der Adressierung berücksichtigt, aber nicht übernommen:

- S5TIME
- TIME
- DATE
- TIME_OF_DAY
- DATE_AND_TIME

Voraussetzung

- Virtual Controller ist geöffnet.
- Registerkarte "DB" ist aktiviert.
- Aus STEP 7 exportierte AWL-Quelle mit Datenbaustein-Informationen liegt vor.

Vorgehen

Um DB-Daten aus AWL-Quellen zu importieren, gehen Sie folgendermaßen vor:

1. Klicken Sie auf "AWL-Import".
   Das Dialogfeld "AWL-Import" wird geöffnet.
2. Wählen Sie die AWL-Quelle aus.
3. Geben Sie die Datenbaustein-Nummer ein.
4. Aktivieren Sie die entsprechende Richtung.
5. Geben Sie Anfangs- und Endbyte ein.

   Hinweis

   Wenn im angegebenen Speicherbereich von "Anfangsbyte" und "Endbyte" bereits Signale vorhanden sind, werden diese Signale überschrieben.

6. Klicken Sie auf "Importieren".

Ergebnis

Der angegebene Adressbereich des Datenbausteins wird importiert.

Siehe auch

Dialogfeld "AWL-Import" (Seite 966)

Eigenschaften der Virtual Controller-Kopplung (Seite 140)

2.7 PLCSIM Advanced-Kopplung

2.7.1 Funktionsweise der PLCSIM Advanced-Kopplung

Funktionsweise

Mit der PLCSIM Advanced-Kopplung kommuniziert SIMIT mit PLCSIM Advanced über eine Softwareschnittstelle von PLCSIM Advanced. Mit der PLCSIM Advanced-Kopplung tauscht SIMIT zyklisch Daten des E/A-Bereichs der projektierten S7-1500-Stationen aus. Unterstützt werden von SIMIT bis zu 16 Stationen pro PC.

Hinweis

Beachten Sie gegebenenfalls weitergehende Einschränkungen und Hinweise in der Dokumentation von PLCSIM Advanced.

Für den Einsatz der PLCSIM Advanced-Kopplung müssen folgende Anwendungen auf einem PC installiert sein:

- SIMIT
- PLCSIM Advanced V2.0
- TIA Portal Openness
- STEP 7 V14 SP1 (optional)

Die Anwendung PLCSIM Advanced wird zusammen mit der Simulation gestartet und beendet.

Siehe auch

Kopplung vom Typ "PLCSIM Advanced" anlegen (Seite 145)

2.7.2 Kopplung vom Typ "PLCSIM Advanced" anlegen

Einleitung

In einem SIMIT-Projekt können Sie eine Kopplung vom Typ "PLCSIM Advanced" anlegen. Unterhalb der Kopplung legen Sie "Stationen" an, in die Sie die Hardware-Konfiguration einer Station aus STEP 7 importieren.

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- TIA Portal Openness ist auf dem SIMIT-PC installiert.
- STEP V14 SP1 ist auf dem SIMIT-PC installiert oder HWCNExport-Datei liegt vor.

Vorgehen

Um eine Kopplung vom Typ "PLCSIM Advanced" anzulegen, gehen Sie folgendermaßen vor:

1. Fügen Sie in der Projektnavigation unter "Kopplungen" eine neue Kopplung hinzu.
   Das Dialogfeld "Auswahl" wird geöffnet.
2. Aktivieren Sie den Kopplungstyp "PLCSIM Advanced".
   Das Dialogfeld "PLCSIM Advanced Import" wird geöffnet.
3. Konfigurieren Sie den Import von Stationen aus einem STEP 7-Projekt oder der HWCNExport-Datei.

Ergebnis

Das Ergebnis nach dem Import eines STEP 7-Projekts finden Sie unter Hardwarekonfiguration in Station importieren (Seite 146).

Wenn Sie die Kopplung vom Typ "PLCSIM Advanced" ohne Import eines STEP 7-Projekts abschließen, wird in der Projektnavigation unter Kopplungen nur der Ordner "PLCSIM Advanced" angelegt.

Siehe auch

Dialogfeld "PLCSIM Advanced-Import" (Seite 962)

HWCNExport-Datei erstellen (Seite 65)

2.7.3 Hardwarekonfiguration in Station importieren

Einleitung

Die Hardware-Konfiguration importieren Sie direkt aus STEP 7. Wenn Sie die Hardware-Konfiguration in STEP 7 ändern, importieren Sie die Hardware-Konfiguration wieder in die Station in SIMIT.

Beim Import der Hardware-Konfiguration aus dem TIA-Portal werden neben den einfachen Datentypen auch folgende Strukturen ausgewertet und als WORD importiert:

- PD_ZSW1
- PD_ZSW2
- PD_Gx_ZSW
- PD_STW1

- PD_STW2
- PD_Gx_STW
- PD_MELDW

Voraussetzung

- Kopplung vom Typ "PLCSIM Advanced" ist angelegt.
- STEP 7 V14 SP1 ist auf dem SIMIT-PC installiert.
- Hardware-Konfiguration in STEP 7 ist übersetzt und geladen.

Vorgehen

Um die Hardware-Konfiguration in eine Station zu importieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Kopplungen > PLCSIM Advanced" den Befehl "Importieren".
   Das Dialogfeld "PLCSIM Advanced-Import" wird geöffnet.
2. Wählen Sie die Quelle, aus der Sie die Hardware-Konfiguration importieren wollen.
3. Wählen Sie den Modus für den Import der Symbolik.
4. Wählen Sie die gewünschten Stationen aus.
5. Klicken Sie auf "Importieren".

Ergebnis

Für jede Station aus dem STEP 7-Projekt wird in der Projektnavigation unter "Kopplungen > PLCSIM Advanced" eine Station angelegt. Beim ersten Import werden die importierten Stationen im Editor "Verteilung" dem Emulations-PC "Eigener Computer" zugeordnet.

Siehe auch

Dialogfeld "PLCSIM Advanced-Import" (Seite 962)

Kopplung vom Typ "PLCSIM Advanced" anlegen (Seite 145)

HWCNExport-Datei erstellen (Seite 65)

Signale einer Kopplung bearbeiten (Seite 188)

2.7.4 Station einer PLCSIM Advanced-Kopplung konfigurieren

Voraussetzung

- Station ist angelegt.
- Hardware-Konfiguration ist importiert.

Vorgehen

Um eine Station einer PLCSIM Advanced-Kopplung zu konfigurieren, gehen Sie folgendermaßen vor:

1. Setzen Sie die "Zeitscheibe" über das Kontextmenü der PLCSIM Advanced-Kopplung.
2. Öffnen Sie den Verteilungseditor der PLCSIM Advanced-Kopplung.
3. Konfigurieren Sie jede Station im Eigenschaftsfenster:
   - Geben Sie bei Bedarf einen Namen ein.
   - Legen Sie den Kommunikationsweg fest.

Ergebnis

Die Station ist konfiguriert. Um Eingangs- und Ausgangssignale hinzuzufügen, importieren (Seite 204) Sie Signaleigenschaften. Signaleigenschaften können Sie bearbeiten.

Weiterführende Informationen zum Bearbeiten und Exportieren von Signalen finden Sie unter:

- Signale einer Kopplung bearbeiten (Seite 188)
- Signaleigenschaften exportieren (Seite 206)

Siehe auch

Verteilungseditor (PLCSIM Advanced) (Seite 148)

Hardwarekonfiguration in Station importieren (Seite 146)

2.7.5 Verteilungseditor (PLCSIM Advanced)

Verteilungseditor

Im Verteilungseditor konfigurieren Sie die Stationen einer PLCSIM Advanced-Kopplung.

Den Verteilungseditor öffnen Sie in der Projektnavigation unter "Kopplungen > PLCSIM Advanced" mit einem Doppelklick.

Menüzeile

In der Menüzeile des Verteilungseditors können Sie folgende Funktionen durchführen:

- "Speichern"
  Speichert die Konfiguration.

Eigenschaften von "Eigener Computer"

- Rechneradresse
  Zeigt die IP-Adresse des PCs an, unter der er aktuell im Netz bekannt ist.
- Kommunikationsschnittstelle
  Legt den Kommunikationsweg zwischen PLCSIM Advanced und SIMIT fest.
  Weiterführende Informationen zu den Kommunikationswegen finden Sie in der Dokumentation zu PLCSIM Advanced.

Eigenschaften einer Station

- Name
  Legt den Stationsnamen im SIMIT-Projekt fest. Ändern Sie den Namen über das Kontextmenü der Station in der Projektnavigation.
- "Projektname (SIMATIC)"
  Zeigt den Namen des STEP 7-Projekts an.
- "Stationsname (SIMATIC)"
  Zeigt den Stationsnamen an.
- "Ablagepfad (SIMATIC)"
  Zeigt den Ablagepfad des STEP 7-/PCS 7-Projekts an.
- Instanznummer (VC)
  Zeigt die Instanznummer an.
  Die Instanznummer wird von SIMIT automatisch vergeben.

Kontextmenü einer Station

- Speicher löschen
  Löscht das Projekt, das in die Station geladen wurde.

Siehe auch

Station einer PLCSIM Advanced-Kopplung konfigurieren (Seite 147)

2.8 PLCSIM-Kopplung

2.8.1 Funktionsweise der PLCSIM-Kopplung

Mit der PLCSIM-Kopplung kommuniziert SIMIT mit PLCSIM über eine Softwareschnittstelle von PLCSIM (Prosim). Beim Einsatz der PLCSIM-Kopplung müssen PLCSIM und SIMIT auf demselben Rechner installiert sein. PLCSIM muss bereits gestartet sein, bevor Sie in SIMIT eine Simulation mit der PLCSIM-Kopplung starten. Während die Simulation läuft, dürfen Sie PLCSIM nicht beenden, da sonst die Verbindung abbricht und eventuell nicht wieder aufgebaut werden kann.

Hinweis

Für den Einsatz der PLCSIM-Kopplung benötigen Sie zusätzlich die Software PLCSIM in der Version 5.4 SP5 oder höher. Diese Software ist nicht im Lieferumfang von SIMIT enthalten.

PLCSIM ist ein unabhängig von SIMIT einsetzbares Programm, das vom SIMATIC Manager aus mit einem STEP 7-Programm geladen werden kann. PLCSIM simuliert dann eine SIMATIC-Steuerung. Damit das Verhalten der Steuerung möglichst realitätsnah nachgebildet werden kann, sollten Sie immer das STEP 7-Programm inklusive der Systemdaten (SDBs) in PLCSIM laden.

Hinweis

Durch die Systemdaten wird auch die Hardwarekonfiguration festgelegt. In der Kopplung von SIMIT mit PLCSIM ist damit nur ein Datenaustausch für Peripherie-Signale möglich, die durch die projektierte Hardware definiert sind.

2.8.2 Konfigurieren der PLCSIM-Kopplung

2.8.2.1 Anlegen einer PLCSIM-Kopplung

Um eine PLCSIM-Kopplung anzulegen, wählen Sie im Dialogfeld "Auswahl" die Option "PLCSIM" aus.

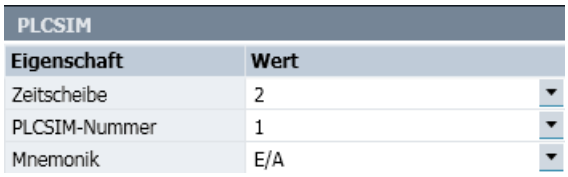
2.8.2.2 Konfigurieren der E/A-Signale in der PLCSIM-Kopplung

Um die Ein-/Ausgangssignale der Kopplung einzugeben, gibt es folgende Möglichkeiten:

- Sie geben die Signale manuell im Kopplungseditor ein.
- Sie importieren die Symboltabelle aus Ihrem SIMATIC-Projekt. Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Import von Signaleigenschaften" (Seite 967).

2.8.2.3 Eigenschaften der PLCSIM-Kopplung

Nach dem Öffnen der Kopplung wird der Kopplungseditor im Arbeitsbereich angezeigt. Im Eigenschaftsfenster können die folgenden Eigenschaften definiert werden:



- **Zeitscheibe**
  Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, entsprechend einem Zyklus von 100 ms.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- **PLCSIM-Nummer**
  Sie können bis zu 8 PLCSIM-Kopplungen in einem Projekt anlegen, um sich mit den entsprechenden Instanzen von PLCSIM zu verbinden. Wählen Sie hier die Nummer der PLCSIM-Instanz aus, mit der diese Kopplung kommunizieren soll. Die entsprechende Nummer steht in der Fenstertitelzeile von PLCSIM.
  Wenn Sie die PLCSIM-Nummer auf "Unzugeordnet" stellen, wird beim Starten der Simulation keine Verbindung mit PLCSIM aufgebaut, und es findet keinerlei Datenaustausch statt.

- **Mnemonik**
  Hier wählen Sie aus, ob für die Bezeichnung der Ein- und Ausgänge die internationale (I/Q) oder deutsche (E/A) Schreibweise verwendet werden soll.

Weitere Funktionen

Kopieren und Einfügen von Normierungen

Weitere Informationen hierzu finden Sie im Abschnitt: Normierung auf anderes Signal übertragen (Seite 216).

Import von Signaleigenschaften

Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Import von Signaleigenschaften" (Seite 967).

Export von Signaleigenschaften

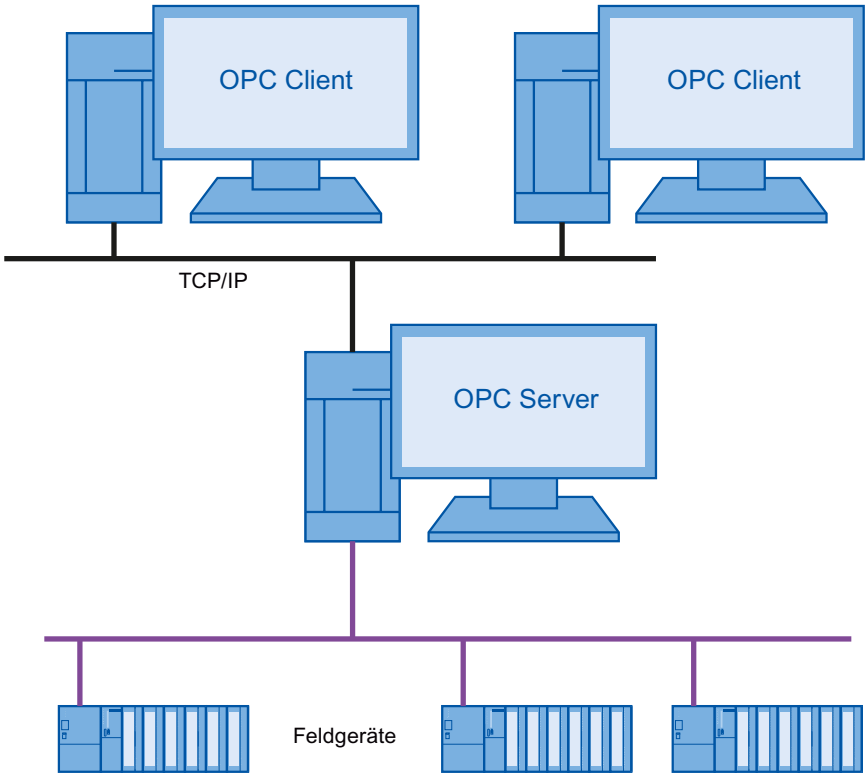
Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Export von Signaleigenschaften" (Seite 970).

2.9 OPC-Kopplung

2.9.1 Funktionsweise der OPC-Kopplung

Einleitung

OPC ist ein weit verbreiteter Kommunikationsstandard in der Automatisierungstechnik, der von der OPC Foundation (www.opcfoundation.org) gepflegt und gefördert wird. Das OPC-Protokoll nutzen Sie zum geräteunabhängigen Datenaustausch zwischen OPC-fähigen Programmen.



In der OPC Classic Architecture besitzt ein OPC-Server eine Verbindung zu einer unterlagerten Geräteebene, deren Signale er über das lokale Netzwerk einem oder mehreren OPC-Clients zugänglich macht.

Hinweis

Benutzen Sie zur Wahrung der Vertraulichkeit Ihrer Daten die OPC-Kopplung nur im Rahmen eines ganzheitlichen Industrial Security-Konzepts, das dem aktuellen Stand der Technik entspricht. Berücksichtigen Sie dabei auch eingesetzte Produkte von anderen Herstellern.

Weitere Informationen finden Sie im Internet (http://www.siemens.com/industrialsecurity).

Unterstützte OPC-Protokolle

Die OPC-Kopplung von SIMIT unterstützt folgende OPC-Protokolle:

- OPC DA 3.0 im Modus "Data Access"
- OPC UA

OPC-Kopplungen in SIMIT

SIMIT unterstützt folgende OPC-Kopplungen:

OPC-Kopplung	Anzahl pro Projekt
OPC DA-Server-Kopplung	1
OPC DA-Client-Kopplung	Bis zu 32[1]
OPC UA-Client-Kopplung	

[1] Zu unterschiedlichen OPC-Servern oder unterschiedlichen Namensräumen von OPC UA-Servern

OPC DA-Server und OPC DA-Client können sowohl lokal auf demselben PC als auch auf verschiedenen PCs betrieben werden. Eine OPC DA-Server- und verschiedene OPC-Client-Kopplungen können auch gleichzeitig in einem SIMIT-Projekt angelegt werden.

Browsing-Funktionalität

SIMIT unterstützt die Browsing-Funktionalität, mit der Sie die von einem OPC-Server bereitgestellten Daten automatisch in die Signaltabelle übernehmen können.

Siehe auch

Funktionsweise des SIMIT OPC DA-Servers (Seite 156)

Funktionsweise der OPC UA-Client-Kopplung (Seite 163)

2.9.2 Signalübertragung zwischen OPC-Server und OPC-Client

Verbindungsaufbau zwischen OPC-Server und OPC-Client

Wenn Sie eine Simulation starten, werden Verbindungen zu allen konfigurierten OPC-Servern automatisch aufgebaut. Wenn nach 5 Sekunden keine Verbindung hergestellt werden kann, startet die Simulation trotzdem. Die Verbindungsversuche laufen im Hintergrund weiter. Statusinformationen werden über das Meldesystem von SIMIT angezeigt. Zusätzlich können Sie den Status jeder OPC-Kopplung über das Signal "Statusanzeige" abfragen.

Beginn der Signalübertragung

Nach erfolgreichem Verbindungsaufbau werden pro OPC-Client die Signale am OPC-Server angemeldet. Ursache für nicht angemeldete Signale ist häufig eine der folgenden:

- Falscher Datentyp
- Signal ist am OPC-Server nicht bekannt.

Nach erfolgreicher Anmeldung werden die Signale zwischen OPC-Server und OPC-Client zyklisch übertragen. Maßgeblich für die Aktualisierung von Werten sind folgende Faktoren:

- Zykluszeit des OPC-Servers
- Zeitscheibe der OPC-Client-Kopplung
- Zyklusmultiplikator eines Signals

Zyklusmultiplikator

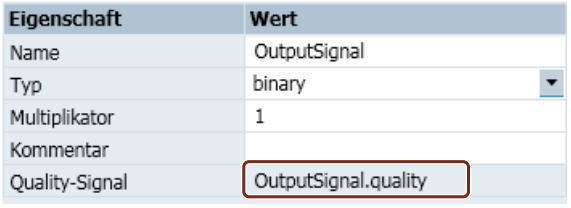
Der Zyklusmultiplikator ist ein ganzzahliger Wert "n", der Werte nur noch in jedem n-ten Zyklus aktualisiert. Mit der Angabe eines Zyklusmultiplikators entlasten Sie die Kommunikationsverbindung zwischen OPC-Server und OPC-Client. Den Zyklusmultiplikator parametrieren Sie für jedes vom OPC-Server bereitgestellte Signal separat.

- Signal ändert sich häufig: Kleiner Zyklusmultiplikator
- Signal ändert sich langsam: Großer Zyklusmultiplikator

Beispiel: Für eine OPC-Client-Kopplung ist über die Zeitscheibe eine Zykluszeit von 250 ms eingestellt. Ein Signal ist mit einem Zyklusmultiplikator "4" parametriert. Das Signal wird damit nur noch alle 250 ms × 4 = 1000 ms aktualisiert.

Quality Code

Übertragene Signale zwischen OPC-Server und OPC-Client liefern standardmäßig einen "Quality Code" mit, der eine Aussage zur Qualität des Signals beinhaltet. Pro Ausgangssignal einer OPC-Client-Kopplung legt SIMIT standardmäßig ein ganzzahliges "Quality-Signal" an, das den Quality Code beinhaltet. Der Signalname setzt sich wie folgt zusammen: <Name des Ausgangssignals>.<quality>



Das Quality-Signal können Sie in Diagrammen wie jedes andere Signal auch verwenden. Während einer laufenden Simulation wird in den Eigenschaften jedes Quality-Signals der "Quality Code" angezeigt.

Vom SIMIT OPC-Client an einen OPC-Server übertragene Signale liefern folgenden Quality Code:

- "Good, non-specific". Die numerische Entsprechung lautet "0xC0" oder "192".

2.9.3 Abbildung der Signale und Datentypen eines OPC-Servers

Abbildung der Datentypen

Signale sind in SIMIT ausschließlich binäre, ganzzahlige oder analoge Signale, wobei ganzzahlige und analoge Signale eine Datenbreite von 8 Byte haben.

Ein OPC-Server kann Signale in verschiedenen anderen Datentypen zur Verfügung stellen. Die folgende Tabelle zeigt die Zuordnung der OPC-Datentypen zu den in SIMIT verwendeten Datentypen.

OPC DA-Server	OPC UA-Server	SIMIT
BOOL, UI1	Boolean	binary
I1, I2, I4, I8, UI1, UI2, UI4, UI8	SByte, Int16, Int32, Int64, Byte, UInt 16, UInt32, UInt64	integer
R4, R8	Float, Double	analog

Für Ausgangssignale der Kopplung entsteht kein Datenverlust bei der Konvertierung, da ganzzahlige und analoge Signale in SIMIT die maximale Datenbreite (8 Byte) haben.

Abbildung der Signale

Die Signale des OPC-Servers können lesbar, beschreibbar oder beides sein. Abhängig davon werden sie in SIMIT als Eingänge oder Ausgänge behandelt.

OPC	SIMIT
readable	Ausgang
writeable	Eingang
readWriteable	Eingang

2.9.4 Status von OPC-Servern

Kernaussage

Im Eigenschaftsfenster der OPC-Client-Kopplung sehen Sie bei laufender Simulation den aktuellen Status des angesprochenen OPC-Servers. Zusätzlich wird der Status an einem Ausgangssignal bereitgestellt, das Sie während der Kopplungskonfiguration definieren.

Die folgende Tabelle zeigt Ihnen die Bedeutung der möglichen Statuswerte.

Hinweis

Welche dieser Statuswerte ein OPC-Server zur Verfügung stellt, ist abhängig von der Implementierung des OPC-Servers.

Statuswert [1]	Bedeutung [2]
0	Keine Verbindung zum OPC-Server
1	The server is running normally. This is the usual state for a server.
2	A vendor specific fatal error has occurred within the server. The server is no longer functioning. The recovery procedure from this situation is vendor specific. An error code of E_FAIL should generally be returned from any other server method.
3	The server is running but has no configuration information loaded and thus cannot function normally. Note this state implies that the server needs configuration information in order to function. Servers which do not require configuration information should not return this state.
4	The server has been temporarily suspended via some vendor specific method and is not getting or sending data. Note that Quality will be returned as OPC_QUALITY_OUT_OF_SERVICE.
5	The server is in Test Mode. The outputs are disconnected from the real hardware but the server will otherwise behave normally. Inputs may be real or may be simulated depending on the vendor implementation. Quality will generally be returned normally.
6	The server is running properly but is having difficulty accessing data from its data sources. This may be due to communication problems, or some other problem preventing the underlying device, control system, etc. from returning valid data. It may be complete failure, meaning that no data is available, or a partial failure, meaning that some data is still available. It is expected that items affected by the fault will individually return with a BAD quality indication for the items.

[1] Die Angaben zu den Statuswerten 1 bis 6 sind der Spezifikation "Data Access Custom Interface Standard, Version 3.00" entnommen.

[2] Die Beschreibung der Statuswerte 1 bis 6 ist der OPC-Spezifikation entnommen

Siehe auch

Eigenschaften der OPC DA-Client-Kopplung (Seite 160)

2.9.5 OPC DA-Client-/Server-Kopplung projektieren

2.9.5.1 Funktionsweise des SIMIT OPC DA-Servers

Verhalten des SIMIT OPC DA-Servers

Wenn Sie die Simulation starten, wird der SIMIT OPC DA-Server automatisch mit den in der Kopplung projektierten Signalen konfiguriert. Ab diesem Zeitpunkt können sich OPC-Clients mit dem Server verbinden und auf die Signale zugreifen.

Wenn Sie die Simulation beenden, bleibt der SIMIT OPC DA-Server so lange aktiv, bis sich alle OPC DA-Clients abgemeldet haben. Signale werden in diesem Zustand nicht mehr aktualisiert.

Wenn Sie die Konfiguration des SIMIT OPC DA-Servers ändern, wird die Konfiguration beim nächsten Start der Simulation automatisch übertragen. Voraussetzung: Der SIMIT OPC DA-Server ist mit keinem OPC DA-Client verbunden.

Quality Codes von gelieferten OPC-Signalen

Der SIMIT OPC DA-Server überträgt OPC-Signale mit folgenden Quality Codes:

Quality Code	Voraussetzung
"Good, non-specific"; 0xC0	Simulation gestartet
"Bad, Out of service"; 0x1C	Simulation beendet, aber noch mindestens ein OPC DA-Client verbunden. Erst wenn alle OPC DA-Clients die Verbindung getrennt haben, wird der SIMIT OPC DA-Server deaktiviert.

Datenübertragung

Der SIMIT OPC DA-Server prüft im Simulationsmodell zyklisch, ob sich Werte geändert haben. Der SIMIT OPC DA-Server unterstützt synchrone und asynchrone Datenübertragung:

- Synchrone Datenübertragung: Alle Werte werden zyklisch übertragen.
- Asynchrone Datenübertragung: Nur geänderte Werte werden übertragen. Der SIMIT OPC DA-Server fasst Werte automatisch in Gruppen zusammen. Kriterien für die Gruppenbildung sind:
  - Datentyp und Multiplikator, jeweils für Eingangs- und Ausgangswerte

  Bei einer Wertänderung werden grundsätzlich alle Werte innerhalb der Gruppe übertragen.

Statusmeldungen

Der SIMIT OPC DA-Server liefert folgende Status, die von einem OPC DA-Client abgefragt werden können:

Wert	Bedeutung	Voraussetzung
1	Running	Simulation gestartet
4	Suspended	Simulation beendet

Bereitgestellte Datentypen

Folgende Datentypen werden im SIMIT OPC DA-Server bereitgestellt:

SIMIT	OPC
Binär	BOOL
Integer	I8, wenn Option "Verwende 64-Bit-Integer" aktiviert, sonst I4
Analog	R8

Siehe auch

Funktionsweise der OPC-Kopplung (Seite 152)

2.9.5.2 SIMIT OPC DA-Server-Kopplung konfigurieren

Voraussetzungen

- Projekt ist in SIMIT geöffnet.

Vorgehen

Um eine SIMIT OPC DA-Server-Kopplung anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Kopplungen" den Befehl "Neue Kopplung".
2. Aktivieren Sie den Kopplungstyp "OPC DA-Server".
3. Wählen Sie eine Zeitscheibe mit einem Aktualisierungszyklus von mindestens 100 ms.
4. Wenn OPC DA-Client und OPC DA-Server auf unterschiedlichen PCs installiert sind, berücksichtigen Sie die "Hinweise zur DCOM-Konfiguration (Seite 161)".

Ergebnis

Der SIMIT OPC DA-Server ist konfiguriert. Um Eingangs- und Ausgangssignale hinzuzufügen, haben Sie folgende Möglichkeiten:

- Signale manuell eingeben
- Signaltabelle importieren (Seite 204)

Siehe auch

Eigenschaften der SIMIT OPC DA-Server-Kopplung (Seite 159)

Abbildung der Signale und Datentypen eines OPC-Servers (Seite 155)

2.9.5.3 OPC DA-Client-Kopplung konfigurieren

Einleitung

Die OPC DA-Client-Kopplung unterstützt nur die asynchrone Datenübertragung von und zu einem OPC DA-Server. Ein OPC DA-Server fasst Werte automatisch in Gruppen zusammen. Bei einer Wertänderung werden alle in der Gruppe enthaltenen Werte übertragen.

Voraussetzungen

- Projekt ist in SIMIT geöffnet.
- ProgID des OPC DA-Servers ist bekannt.
- Hostname des PC ist bekannt, auf dem der OPC DA-Server installiert ist.
- OPC DA-Server ist im Netzwerk erreichbar.

Vorgehen

Um eine OPC DA-Client-Kopplung anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Kopplungen" den Befehl "Neue Kopplung".
2. Aktivieren Sie den Kopplungstyp "OPC DA-Client".
3. Konfigurieren Sie die Verbindung zum OPC DA-Server.
4. Wenn OPC DA-Client und OPC DA-Server auf unterschiedlichen PCs installiert sind, berücksichtigen Sie die "Hinweise zur DCOM-Konfiguration (Seite 161)".

Ergebnis

Die Kopplung mit dem OPC DA-Server ist konfiguriert. Um Eingangs- und Ausgangssignale hinzuzufügen, haben Sie folgende Möglichkeiten:

- Signale von einem OPC-Server lesen (Seite 218)
- Signaltabelle importieren (Seite 204)
- Signale manuell eingeben

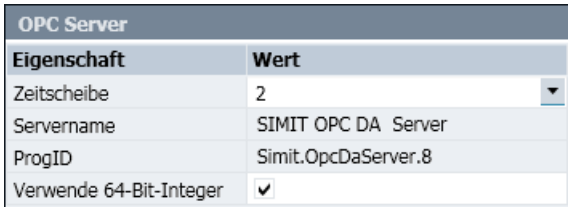
Siehe auch

Eigenschaften der OPC DA-Client-Kopplung (Seite 160)

Signale einer Kopplung bearbeiten (Seite 188)

2.9.5.4 Eigenschaften der SIMIT OPC DA-Server-Kopplung

Nach dem Öffnen der Kopplung wird der Kopplungseditor im Arbeitsbereich angezeigt. Im Eigenschaftsfenster können die folgenden Eigenschaften definiert werden:



- **Zeitscheibe**
  Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, die in der Standardeinstellung einem Zyklus von 100 ms entspricht.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

  Der OPC-Server aktualisiert seine Daten alle 100 ms. Zykluszeiten unterhalb dieses Wertes sollten deshalb nicht eingestellt werden.

- **Servername**
  Der SIMIT-OPC-Servername wird nur zur Information angezeigt. Er ist mit "SIMIT OPC DA Server" fest vorgegeben. Servername oder ProgID werden benötigt, um einen OPC-Client für den Zugriff auf diesen OPC-Server zu konfigurieren.
- **ProgID**
  Die ProgID des SIMIT-OPC-Servers wird nur zur Information angezeigt. Sie ist mit "Simit.OpcDaServer.8" fest vorgegeben. ProgID oder Servername werden benötigt, um einen OPC-Client für den Zugriff auf diesen OPC-Server zu konfigurieren.
- **Verwende 64-Bit-Integer**
  Ganzzahlige Werte (Integer-Werte) in SIMIT haben eine Datenbreite von 64 Bit (8 Byte). Sie werden daher in der OPC-Server-Kopplung auch in diesem Format angelegt.
  Deaktivieren Sie diese Option, wenn ein mit diesem OPC-Server verkoppelter OPC-Client Integerwerte mit 8 Byte Datenbreite nicht verarbeiten kann. SIMIT überträgt dann alle ganzzahligen Werte mit einer Datenbreite von nur 32 Bit (4 Byte). Beachten Sie, dass Datenverlust entsteht, wenn die Zahl nicht mit 32 Bit dargestellt werden kann.

Siehe auch

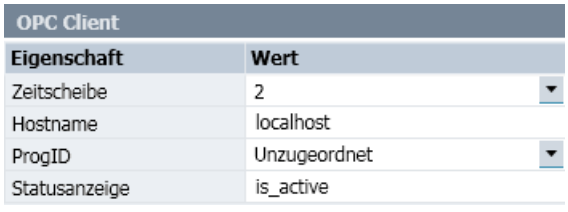
Dialogfeld "Import von Signaleigenschaften" (Seite 967)

Dialogfeld "Export von Signaleigenschaften" (Seite 970)

SIMIT OPC DA-Server-Kopplung konfigurieren (Seite 158)

2.9.5.5 Eigenschaften der OPC DA-Client-Kopplung

Nach dem Öffnen der Kopplung wird der Kopplungseditor im Arbeitsbereich angezeigt. Im Eigenschaftsfenster können die folgenden Eigenschaften definiert werden:



- **Zeitscheibe**
  Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, die in der Standardeinstellung einem Zyklus von 100 ms entspricht.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- **Hostname**
  Der Name des Rechners, auf dem der OPC-Server läuft. Mit diesem OPC-Server kommuniziert die Kopplung. Sie können den Rechnernamen oder dessen IP-Adresse angeben. Voreingestellt ist der lokale Rechner (localhost).
- **ProgID**
  Tragen Sie hier die ProgID des OPC-Servers ein, mit dem sich der OPC-Client auf dem unter Hostname angegebenen Rechner verbinden soll.
  Falls die Verbindung lokal erfolgt, d. h. unter Hostname "localhost" eingetragen ist, werden die ProgIDs der OPC-Server zur Auswahl gegeben, die lokal zur Verfügung stehen. Wählen Sie dann aus, mit welchem OPC-Server die Kopplung kommunizieren soll.
  Wenn die ProgID auf "Unzugeordnet" steht, wird beim Starten der Simulation keine Verbindung zu einem OPC-Server aufgebaut. Die Kopplungssignale werden nicht zyklisch aktualisiert.
- **Statusanzeige**
  Legt den Namen des ganzzahligen Ausgangssignals fest, in dem bei laufender Simulation der Status des verbundenen OPC-Servers bereitgestellt wird. Dieses Ausgangssignal können Sie zur Auswertung des Serverstatus in Ihrem SIMIT-Projekt verwenden. Für jede OPC DA-Client-Kopplung wird je ein Ausgangssignal erzeugt.

Siehe auch

Dialogfeld "Import von Signaleigenschaften" (Seite 967)

Dialogfeld "Export von Signaleigenschaften" (Seite 970)

Status von OPC-Servern (Seite 155)

OPC DA-Client-Kopplung konfigurieren (Seite 158)

2.9.5.6 Hinweise zur DCOM-Konfiguration

Grundlagen zur DCOM-Konfiguration

Wenn Sie OPC DA-Server und OPC DA-Client auf **einem** PC installiert haben, sind keine weiteren Anpassungen in der Betriebssystemkonfiguration erforderlich.

Wenn Sie OPC DA-Server und OPC DA-Client auf **unterschiedlichen** PCs installiert haben, müssen Sie folgende Einstellungen im Betriebssystem überprüfen und bei Bedarf anpassen:

- Firewall-Konfiguration
- Domänen und Benutzer
- Rechte-Vergabe

Hinweis

Alle Einstellungen am Rechtesystem eines Windows-PC sind in starkem Maße abhängig von der Version des Betriebssystems und den bereits auf Ihrem PC installierten Programmen. Wir können Ihnen daher hier nur Hinweise, nicht aber exakte Handlungsanweisungen zur Konfiguration von DCOM in Ihrer speziellen Umgebung geben.

Hinweis

Nehmen Sie Veränderungen an den Einstellungen Ihres PC nur dann vor, wenn Sie deren Bedeutung genau kennen. Achten Sie darauf, dass Sie den Schutz Ihres PC nicht ungewollt aufheben.

Firewall-Konfiguration

Wenn Sie eine Firewall benutzen, müssen Sie für DCOM den TCP-Port 135 für eingehende Anfragen freigeben. Alle beteiligten OPC-Server, OPC-Clients und die beiden Programme "Microsoft Management Console" und "OPCEnum" dürfen nicht geblockt werden.

Domänen und Benutzer

Alle beteiligten PCs müssen derselben Arbeitsgruppe bzw. Domäne angehören. Verwenden Sie auf allen PCs denselben Benutzernamen und dasselbe Passwort.

Rechte-Vergabe

Beim Aufbau einer OPC-Verbindung über DCOM greift ein PC auf Ressourcen eines anderen PC zu oder startet auf diesem bestimmte Prozesse. Dies muss explizit erlaubt werden, weil es sich dabei um Vorgänge handelt, die potentiell den Schutz Ihres PC gefährden könnten.

Grundsätzlich sind sowohl die Zugriffsberechtigungen als auch Start- und Aktivierungsberechtigungen richtig einzustellen. Empfehlung: Richten Sie eine neue Benutzergruppe ein, die diese erweiterten Zugriffsrechte besitzt. Ordnen Sie dieser Benutzergruppe die Benutzer zu, welche die OPC-Verbindung nutzen.

Neben allgemeinen Einstellungen der COM-Sicherheit müssen Sie die Rechte des OPC-Servers sowie des Hilfsprogramms "OPCEnum.exe" einstellen.

Zur Konfiguration dient unter Windows 7 das Programm "dcomcnfg.exe".

Einzelheiten zur DCOM-Konfiguration finden Sie im Dokument "Using OPC via DCOM with Windows XP Service Pack 2" der OPC Foundation im Internet unter http://www.opcfoundation.org.

Siehe auch

SIMIT OPC DA-Server-Kopplung konfigurieren (Seite 158)

OPC DA-Client-Kopplung konfigurieren (Seite 158)

2.9.6 OPC UA-Client-Kopplung projektieren

2.9.6.1 Funktionsweise der OPC UA-Client-Kopplung

Einleitung

In SIMIT können Sie bis zu 32 OPC UA-Client-Kopplungen anlegen. Jeder OPC UA-Client kann zu einem OPC UA-Server Kontakt aufnehmen und Daten austauschen.

Die OPC UA-Client-Kopplung unterstützt die gesicherte Datenübertragung.

Gesicherte Datenübertragung

Bei der gesicherten Datenübertragung authentifizieren sich der OPC UA-Server und OPC UA-Client über den Austausch von Zertifikaten. Das OPC UA-Client-Zertifikat wird bei der Installation von SIMIT automatisch erzeugt. Am SIMIT-PC ist standardmäßig nur die Benutzergruppe "Administratoren" berechtigt, eine OPC UA-Client-Kopplung mit gesicherter Datenübertragung zu konfigurieren.

Siehe auch

Abbildung der Signale und Datentypen eines OPC-Servers (Seite 155)

Funktionsweise der OPC-Kopplung (Seite 152)

Zertifikate (Seite 163)

2.9.6.2 Zertifikate

Einleitung

Beim Datenaustausch über OPC UA wird die Identität der Verbindungspartner durch Zertifikate bestätigt. Die Zertifikate werden beim ersten Verbindungsaufbau zwischen OPC UA-Client und OPC UA-Server automatisch ausgetauscht. Vor jedem weiteren Verbindungsaufbau wird überprüft, ob die Zertifikate noch gültig sind.

Ablage von Zertifikaten

Während der Installation von SIMIT wird im Verzeichnis "C:\ProgramData\Siemens \Automation\SIMIT\8.0\PKI\" folgende Verzeichnisstruktur angelegt. Innerhalb dieser Verzeichnisstruktur haben Mitglieder der Gruppe "Administratoren" Vollzugriff.

- "issuers": Enthält Zertifikate, die zum Verifizieren von CA-Zertifikaten benötigt werden.
  Zugriffsrechte:
  - Gruppe "Benutzer": Lesen
- "own": Enthält das "Application Instance Certificate" und den während der Installation erzeugten private-Key des OPC UA-Clients. Beim ersten Verbindungsaufbau greift der OPC UA-Server lesend auf dieses Verzeichnis zu. Der private-Key wird nur einmal erzeugt und bei einer Software-Aktualisierung nicht überschrieben.
  Zugriffsrechte:
  - Gruppe "Benutzer": kein Zugriff im Unterverzeichnis \private
  - Gruppe "Benutzer": Lesen im Unterverzeichnis \certs
- "trusted\certs": Enthält Zertifikate von OPC UA-Servern, mit denen der OPC UA-Client Daten austauschen kann.
  Zugriffsrechte:
  - Gruppe "Benutzer": Lesen
- "rejected\certs": Enthält abgewiesene OPC UA-Server-Zertifikate.
  Zugriffsrechte:
  - Gruppe "Benutzer": Schreiben

Hinweise zum Zertifikatsaustausch

Aus Sicherheitsgründen werden neue Verbindungen zu einem OPC UA-Server immer abgelehnt. Eine neue Verbindung muss vom Administrator explizit sowohl auf dem OPC UA-Client und dem OPC UA-Server bestätigt werden.

- OPC UA-Server: Der OPC UA-Client muss in die Liste "vertrauenswürdiger Clients" aufgenommen werden.
- OPC UA-Client: Beim ersten Verbindungsaufbau werden vom OPC-Server gesendete Zertifikate im Ordner "rejected\certs" abgelegt. Die Zertifikate müssen danach manuell in das Verzeichnis "trusted\certs" kopiert werden.

Standardmäßig besitzen nur Mitglieder der Gruppe "Administratoren" Zugriffsrechte auf die Ordner "own\private" und "trusted\certs". Wenn ein anderer Benutzer eine OPC UA-Client-Kopplung einrichten will, haben Sie folgende Möglichkeiten:

- Der Administrator richtet diesem Benutzer die benötigten Zugriffsrechte ein.
- Der Administrator gibt dem Benutzer Leserechte auf dem Ordner "own\private" und kopiert die unter "rejected\certs" abgelegten Zertifikate in den Ordner "trusted\certs".

Nach einer dieser beiden Aktionen kann der Benutzer die Verbindung zum OPC UA-Server aufbauen.

Siehe auch

OPC UA-Client-Kopplung konfigurieren (Seite 165)

2.9.6.3 OPC UA-Client-Kopplung konfigurieren

Voraussetzungen

- Projekt ist in SIMIT geöffnet.
- URL des PC mit installiertem OPC UA-Server ist bekannt.
- Der OPC UA-Server ist im Netzwerk erreichbar.
- Der OPC UA-Server unterstützt die Authentifizierungsmethode "Anonymous".
- Zugriffsrechte auf folgende Verzeichnisse sind vorhanden:
  - "C:\ProgramData\Siemens\Automation\SIMIT\8.0\PKI\own\private": Lesen
  - "C:\ProgramData\Siemens\Automation\SIMIT\8.0\PKI\trusted\certs": Schreiben

Vorgehen

Um eine OPC UA-Kopplung anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Kopplungen" den Befehl "Neue Kopplung".
2. Aktivieren Sie den Kopplungstyp "OPC UA-Client".
3. Konfigurieren Sie die Verbindung zum OPC UA-Server.
4. Wenn Sie eine gesicherte Datenübertragung konfigurieren:
   - Kopieren Sie das vom OPC UA-Server gesendete Zertifikat aus dem Ordner "..\rejected \certs" in den Ordner "..\trusted\certs".
   - Nehmen Sie am OPC UA-Server den OPC UA-Client in die Liste "vertrauenswürdiger Clients" auf.

Ergebnis

Die Kopplung mit dem OPC UA-Server ist konfiguriert. Um Eingangs- und Ausgangssignale hinzuzufügen, haben Sie folgende Möglichkeiten:

- Signale von einem OPC-Server lesen (Seite 218)
- Signaltabelle importieren (Seite 204)
- Signale manuell eingeben

Siehe auch

Eigenschaften der OPC UA-Client-Kopplung (Seite 167)

Abbildung der Signale und Datentypen eines OPC-Servers (Seite 155)

Fehlermeldungen beim Verbindungsaufbau zum OPC UA-Server (Seite 166)
Zertifikate (Seite 163)
Signale einer Kopplung bearbeiten (Seite 188)

2.9.6.4 Fehlermeldungen beim Verbindungsaufbau zum OPC UA-Server

OPC UA-Client: Fehlermeldungen und Abhilfe

Fehlermeldung	Ursache	Abhilfe
"The client application does not have a certificate assigned. Secured connections are not possible."	Sie besitzen möglicherweise keine Leserechte auf folgendem Verzeichnis: • "C:\ProgramData\Siemens \Automation\SIMIT\8.0 \PKI\own \private"	Lassen Sie sich vom Administrator Leserechte für das angegebene Verzeichnis einrichten.
"Could not load the application certificate."		
"Certificate is not trusted."	Das Server-Zertifikat ist nicht in der Liste der vertrauenswürdigen Zertifikate enthalten.	Lassen Sie das entsprechende Zertifikat vom Administrator von "C:\ProgramData\Siemens\Automation\SIMIT \8.0\PKI\rejected\certs" nach "C:\ProgramData\Siemens\Automation\SIMIT \8.0\PKI\trusted\certs" kopieren.

OPC UA-Server: Fehlermeldungen und Abhilfe

Fehlermeldung	Ursache	Abhilfe
"Socket was closed gracefully."	Der OPC UA-Server vertraut dem SIMIT OPC UA-Client nicht.	Nehmen Sie den OPC UA-Client in die Liste der vertrauenswürdigen Clients auf.
"Error received from remote host."		

2.9.6.5 Eigenschaften der OPC UA-Client-Kopplung

- **Zeitscheibe**
  Legt den Zyklus fest, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, die in der Standardeinstellung einem Zyklus von 100 ms entspricht.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- **OPC UA Server URL**
  Legt die Adresse fest, unter welcher der OPC UA-Server erreichbar ist. Folgende Adressen sind zulässig:
  - URL inklusive Port
    Wenn Sie keinen Port angeben, wird standardmäßig Port 4840 angenommen. Diesen Port verwendet der Dienst "Local Discovery Server".
  - Servername
  - IP-Adresse
- **Endpunkt**
  Legt den OPC UA-Server und den Übertragungsmodus fest.
  - Unterstützte Protokolle: OPC-UA-binäres Protokoll (Kommunikationsprofil "UA-TCP UA-SC UA Binary")
  - Unterstützte Security Policies: Basic256Sha256, Basic256, Basic128Rsa15, None
  - Unterstützte Message Security Modes: None, Sign, Sign&Encrypt
- **Namensraum URI**
  Legt den Umfang der vom gewählten OPC UA-Server bereitgestellten Signale fest. Ein OPC UA-Server kann mehrere Namensräume zur Verfügung stellen. Wenn Sie Signale aus mehreren Namensräumen nutzen wollen, legen Sie für jeden Namensraum eine eigene OPC UA-Client-Kopplung an.
- **Statusanzeige**
  Legt den Namen des ganzzahligen Ausgangssignals fest, in dem bei laufender Simulation der Status des verbundenen OPC-Servers bereitgestellt wird. Dieses Ausgangssignal können Sie zur Auswertung des Serverstatus in Ihrem SIMIT-Projekt verwenden. Für jede OPC UA-Client-Kopplung wird je ein Ausgangssignal erzeugt.

Siehe auch

OPC UA-Client-Kopplung konfigurieren (Seite 165)

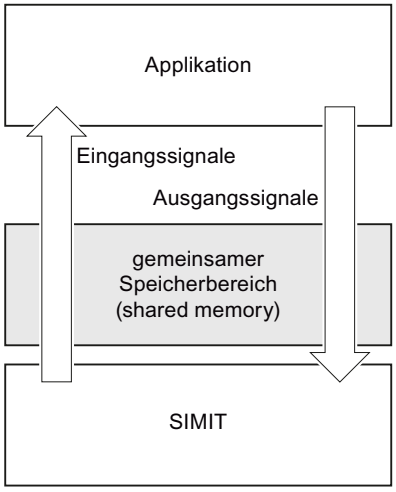
Status von OPC-Servern (Seite 155)

2.10 Shared Memory-Kopplung

2.10.1 Funktionsweise der SHM-Kopplung

Mit der Shared Memory-Kopplung, kurz SHM-Kopplung, kommuniziert SIMIT mit beliebigen anderen Applikationen über einen gemeinsamen Speicherbereich (shared memory, SHM). Diese Kopplung ist universell einsetzbar und hochperformant.

Eingangssignale sind die Signale, die von SIMIT in den Speicherbereich geschrieben werden und Ausgangssignale sind die Signale, die von SIMIT aus dem Speicherbereich gelesen werden.



Hinweis

Der Zugriff auf den gemeinsamen Speicherbereich ist nicht auf einen bestimmten Benutzer beschränkt. Jede auf diesem PC ausgeführte Applikation hat Zugriff auf den gemeinsamen Speicherbereich.

2.10.1.1 Zugriff auf den Speicherbereich

Wenn mehrere voneinander unabhängige Prozesse auf denselben Speicherbereich zugreifen, muss der Zugriff synchronisiert werden, damit die Konsistenz der Werte erhalten bleibt. Dazu wird in der SHM-Kopplung ein Mutex als Synchronisierungsobjekt verwendet.

SIMIT schreibt und liest in jedem Simulationszyklus alle in der SHM-Kopplung definierten Aus- und Eingangssignale und blockiert während dieser Operation den Mutex. Alle weiteren Applikationen, die auch auf diesen Speicherbereich zugreifen, sollten genauso verfahren.

Hinweis

Jede Applikation, die über eine SHM-Kopplung mit SIMIT verkoppelt ist, sollte den Zeitraum, in dem sie den Mutex blockiert, so kurz wie möglich halten, um den Zugriff auf den gemeinsamen Speicherbereich von SIMIT und anderen Applikationen nicht unnötig lange zu blockieren.

2.10.1.2 Struktur des Speicherbereichs

Kopfbereich und Datenbereich

Der Speicherbereich ist unterteilt in einen Kopfbereich (Header) und einen Datenbereich. Der Kopfbereich ist mindestens 8 Byte groß. In den ersten 4 Byte des Kopfbereichs steht die Größe des gesamten Speicherbereichs, in den folgenden 4 Byte die Größe des Kopfbereiches. Für beide Werte gilt das Format "Little Endian".



Struktur des Datenbereichs

SIMIT adressiert den Datenbereich byteweise analog der Adressierung des E/A-Bereichs von SIMATIC-Automatisierungssystemen. Jedes Signal der SHM-Kopplung ist mit einer eindeutigen Adresse im Datenbereich verknüpft. Da unter einer Adresse im Datenbereich ein Signal aber nur als Eingangssignal oder als Ausgangssignal definiert werden kann, kann die gleiche Adresse nicht sowohl einem Ein- wie auch Ausgangssignal zugeordnet werden.

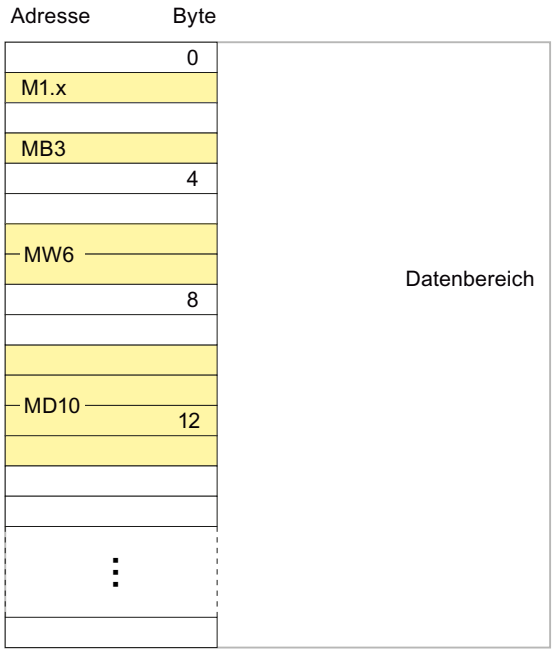
Die Ein- und Ausgangssignale werden auf einen gemeinsamen Speicherbereich abgebildet. Es sind keine Überlappungen zwischen Ein- und Ausgangssignalen zulässig. Entsprechend

überlappende Ein- und Ausgangssignale werden in der Konsistenzprüfung erkannt und als Inkonsistenzen angezeigt.

Ein Signal belegt entsprechend seines Datentyps 1, 2 oder 4 Byte unter seiner Adresse im Datenbereich:

- 1 Byte für die Datentypen BOOL und BYTE,
- 2 Byte für die Datentypen WORD und INT und
- 4 Byte für die Datentypen DWORD, DINT und REAL.

Die folgende Abbildung zeigt ein Beispiel für die Adressierung für verschiedene Datentypen:



Struktur des Kopfbereichs

Der minimale Kopfbereich ist 8 Byte groß. In den ersten 4 Byte des Kopfbereichs steht die Größe des gesamten Speicherbereichs, in den folgenden 4 Byte die Größe des Kopfbereiches.



Wenn SIMIT den gemeinsamen Speicherbereich anlegt, kann SIMIT optional den Kopfbereich durch weitere Größen und eine Liste der Signale ergänzen. Diese Information kann dann von Applikationen zur Konfiguration ihres Zugriffs auf den Datenbereich genutzt werden.



Die Version kennzeichnet die Speicherstruktur. Da die hier definierte Struktur die Versionskennung "0" hat, finden Sie dort immer den Wert "0" eingetragen. Bei Änderungen der Struktur des Speicherbereiches wird die Version entsprechend geändert.

Der für die SHM-Kopplung vorgegebene Abtastzyklus wird in Millisekunden (ms) als ganzzahliger Wert eingetragen.

In jedem Zyklus der SHM-Kopplung wird der ganzzahlige Wert des Zykluszählers um 1 erhöht.

Im Byte 16 des Kopfbereichs ist in einem Byte die Länge des Mutexnamens, d. h. die Anzahl *m* der Zeichen des Mutexnamens abgelegt. Der Mutexname ist im anschließenden Kopfbereich ab Byte 17 abgelegt.

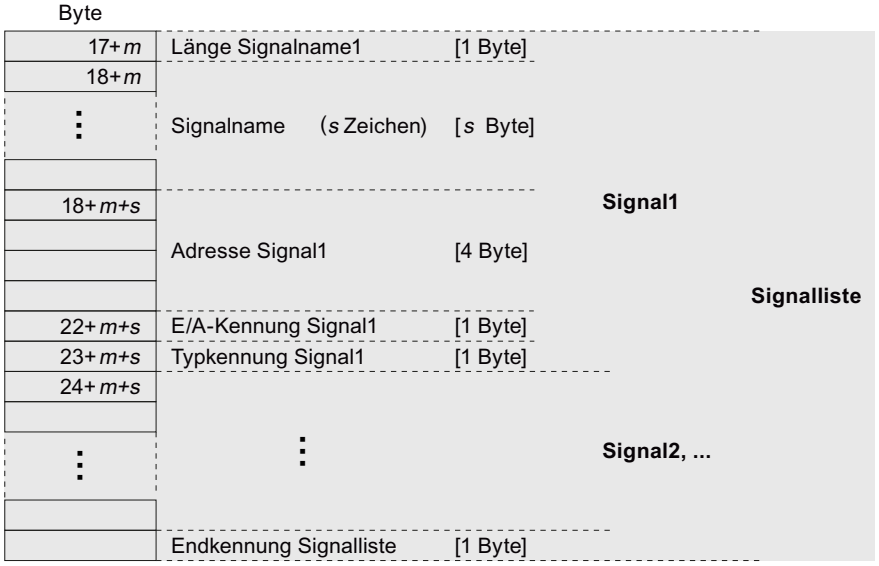
Die ab Byte 17+*m* im Kopfbereich angelegte Signalliste liefert Information über die Signale im Datenbereich. Für jedes Signal sind folgende Größen angegeben:

- Länge des Signalnamens, d. h. die Anzahl der Zeichen des Signalnamens
- Signalname
- Adresse des Signals, d. h. der Offset des Signals im Datenbereich
- E/A-Kennung, d. h. die Kennung, ob das Signal ein Eingangssignal (Kennwert 0) oder Ausgangssignal (Kennwert 1) ist
- Typkennung zur Kennzeichnung des Datentyps

Tabelle 2-1 Typkennung der Signale

Typkennung	Bedeutung (Datentyp)
0	BOOL, Bitadresse 0
1	BOOL, Bitadresse 1
...	...
7	BOOL, Bitadresse 7
8	BYTE
9	WORD
10	INT
11	DWORD
12	DINT
13	REAL

Die Struktur der Signalliste im Kopfbereich ist in der folgenden Abbildung dargestellt:



Die Signalliste ist mit einer Endkennung vom Wert "0" abgeschlossen.

2.10.1.3 Anlegen des Speicherbereichs

Der gemeinsame Speicherbereich kann entweder von SIMIT oder von einer anderen, mit SIMIT über den Speicherbereich gekoppelten Applikation angelegt werden. Entsprechend angepasst verhält sich SIMIT beim Starten einer Simulation, die eine SHM-Kopplung enthält.

Falls der gemeinsame Speicherbereich von einer Applikation angelegt wurde, d. h. beim Starten der Simulation in SIMIT schon existiert, dann wird der Speicherbereich von SIMIT geöffnet. SIMIT verbindet sich nur dann mit dem Speicherbereich, wenn die Größe des Datenbereichs mit der Größe des durch die Ein-/Ausgangssignale in der SHM-Kopplung definierten Adressbereichs übereinstimmt. Andernfalls wird eine Fehlermeldung ausgegeben und SIMIT verbindet sich nicht mit dem Speicherbereich.

Wenn der gemeinsame Speicherbereich beim Starten von SIMIT nicht existiert, dann wird er von SIMIT angelegt. Im Kopfbereich trägt SIMIT die Größe für den gesamten Speicher- sowie

den Kopfbereich ein. Die Größe des Datenbereichs wird durch die höchste Adresse der in der SHM-Kopplung definierten Signale festgelegt. Optional können von SIMIT weitere Größen und eine Liste der Signale im Kopfbereich eingetragen werden. Weitere Informationen hierzu finden Sie im Abschnitt: Struktur des Speicherbereichs (Seite 169).

2.10.2 Konfigurieren der SHM-Kopplung

2.10.2.1 Anlegen einer SHM-Kopplung

Um eine SHM-Kopplung anzulegen, wählen Sie im Dialogfeld "Auswahl" die Option "SHM" aus.

Hinweis

In einem Projekt können maximal 32 SHM-Kopplungen angelegt werden.

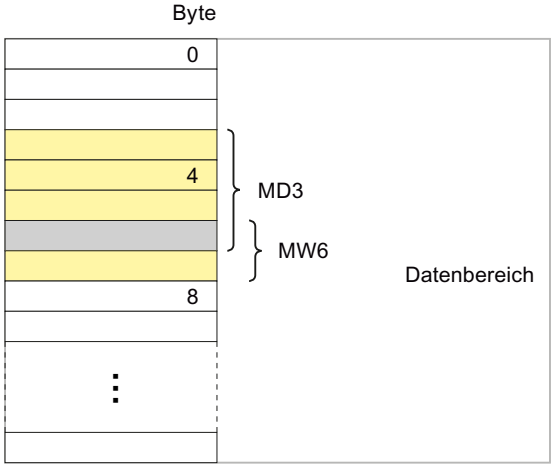
2.10.2.2 Konfigurieren der Signale in der SHM-Kopplung

Um die Ein-/Ausgangssignale der Kopplung einzugeben, gibt es folgende Möglichkeiten:

- Geben Sie die Signale manuell im Kopplungseditor ein.
- Importieren Sie die Symboltabelle. Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Import von Signaleigenschaften" (Seite 967).

SIMIT liest die Ausgangssignale aus und schreibt die Eingangssignale in den gemeinsamen Speicherbereich. Der Zugriff erfolgt zyklisch in dem für die Kopplung definierten Zyklus. Ein Byte des Speicherbereichs muss dabei eindeutig entweder einem Ein- oder Ausgangssignal zugewiesen sein. Jedes Eingangssignal muss eindeutig im Datenbereich abgebildet sein und die Datenbereiche von verschiedenen Eingangssignalen dürfen sich nicht überlappen.

Die folgende Abbildung zeigt ein Beispiel mit 2 Eingangssignalen MD3 und MW6, die in Byte 6 überlappen:



Die Konsistenzprüfung von SIMIT meldet überlappende Signale als Fehler.

Hinweis

Da die kleinste adressierbare Einheit des Speicherbereichs 1 Byte ist, kann 1 Byte nur entweder einem Eingangssignal oder einem Ausgangssignal zugeordnet werden. Binäre Signale mit derselben Byte-Adresse können deshalb entweder alle nur binäre Eingangssignale oder binäre Ausgangssignale sein.

SIMIT greift auf den Speicherbereich auch nicht bitweise schreibend, sondern nur byteweise schreibend zu. 1 Bit in einem Byte des Speicherbereiches, für das kein binäres Eingangssignal in der SHM-Kopplung definiert ist, wird beim Schreibzugriff von SIMIT auf den Speicherbereich "0" gesetzt.

Die Adressbezeichnung eines Signals beginnt mit "M" wie "Memory" und enthält den Datentyp und die Adresse. Angelehnt an die Schreibweise von Adressen in SIMATIC-Automatisierungssystemen, können die in der folgenden Tabelle angegebenen Datentypen verwendet werden.

Tabelle 2-2 Definition der Datentypen

Datentyp	Größe	Schreibweise	Wertebereich
BOOL	1 Bit	M<byte>.<bit>	True/False
BYTE	1 Byte (8 Bit)	MB<byte>	0 ... 255 bzw. −128 ... 127
WORD	2 Byte	MW<byte>	0 ... 65,535
INT	2 Byte	MW<byte>	−32,768 ... 32,767
DWORD	4 Byte	MD<byte>	0 ... 4,294,967,295
DINT	4 Byte	MD<byte>	−2,147,483,648 ... 2,147,483,647
REAL	4 Byte	MD<byte>	$\pm 1{,}5 \times 10^{-45}$ bis $\pm 3{,}4 \times 10^{38}$

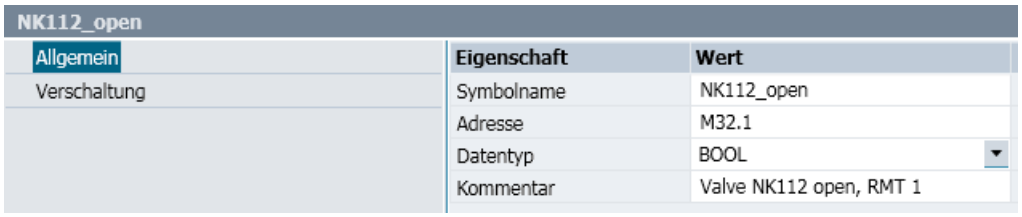
Für die Abbildung der ganzzahligen (Integer) Signale in Simulationsprojekten von SIMIT auf die Datentypen im Speicherbereich gilt folgendes:

- Für den Datentyp BYTE wird das niedrigstwertige Byte (LSB) des Signals übernommen.
- Die Datentypen WORD und DWORD sind vorzeichenlos (unsigned) und die Datentypen INT und DINT sind vorzeichenbehaftet (signed). Ihre Werte werden auf die in der obigen Tabelle angegebenen Wertebereiche begrenzt.

Für den Aufbau einer Gleitpunktzahl gilt der Standard "IEEE Standard for Binary Floating Point Arithmetic" (ANSI/IEEE Std 754-1985).

2.10.2.3 Signaleigenschaften in der SHM-Kopplung

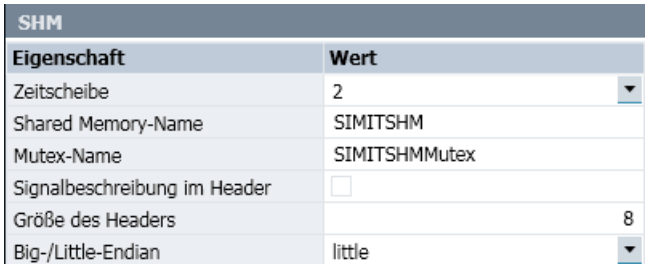
Die Eigenschaften eines Signals werden in den einzelnen Spalten des Kopplungseditors und im Eigenschaftsfenster dargestellt.



- **Symbolname**
  Über diesen Namen wird das Signal in SIMIT identifiziert.
- **Adresse**
  Unter dieser Byte-Adresse werden Werte des Signals im Datenbereich des gemeinsamen Speicherbereichs gespeichert. Die Adresse "0" entspricht dem ersten Byte nach dem Kopfbereich.
- **Datentyp**
  Der Datentyp legt für ein Signal fest, welchen Platz es im Datenbereich beansprucht und wie seine dort gespeicherten Werte zu interpretieren sind. Dabei gibt es folgende Möglichkeiten:
  - Logischer Wert
  - Ganzzahliger vorzeichenbehafteter Wert
  - Vorzeichenloser Wert
  - Gleitkommazahl
- **Kommentar**
  Der Kommentar dient der Dokumentation des Signals. Er wird nicht ausgewertet.

2.10.2.4 Eigenschaften der SHM-Kopplung

Nach dem Öffnen der Kopplung wird der Kopplungseditor im Arbeitsbereich angezeigt. Im Eigenschaftsfenster können die folgenden Eigenschaften definiert werden:



- **Zeitscheibe**
  Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, entsprechend einem Zyklus von 100 ms.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- **Shared Memory-Name**
  Tragen Sie hier den Namen ein, mit dem der gemeinsame Speicherbereich angesprochen werden kann.
- **Mutex-Name**
  Geben Sie hier den Namen des Mutex zur Synchronisierung des Zugriffs auf den gemeinsamen Speicherbereich an.
- **Signalbeschreibung im Header**
  Mit dieser Option können Sie wählen, ob SIMIT einen erweiterten Kopfbereich anlegen soll. Weitere Informationen hierzu finden Sie im Abschnitt: Struktur des Kopfbereichs (Seite 170).
- **Größe des Headers**
  Geben Sie hier die Größe des Kopfbereichs in Byte an. Der Wert ist beliebig, muss aber minimal 8 Byte sein. Wenn SIMIT den gemeinsamen Speicherbereich anlegt, wird ein Kopfbereich mit der angegebenen Größe angelegt.

Hinweis

Falls Sie die Option "Signalbeschreibung im Header" aktivieren, wird die Größe des Kopfbereiches von SIMIT anhand der Ein-/Ausgangssignale in der Kopplung festgelegt. Die Eigenschaft "Größe des Headers" ist in diesem Fall nicht editierbar.

- **Big/Little Endian**
  Mit dieser Eigenschaft wird die Byte-Reihenfolge festgelegt, in der Werte des Datentyps WORD, INT, DWORD bzw. DINT im Datenbereich kodiert sind.

Einstellung	Byte-Reihenfolge
Big Endian	Das höchstwertige Byte wird zuerst, also an der kleinsten Speicheradresse gespeichert.
Little Endian	Das niedrigstwertige Byte wird zuerst, also an der kleinsten Speicheradresse gespeichert.

2.10.2.5 Import und Export der Signale

Das txt-Format, mit dem in SIMIT der Inhalt von Kopplungen abgespeichert werden kann, enthält SIMATIC-Adressen und Angaben zur Normierung. Da in SHM-Kopplungen solche Informationen nur teilweise existieren, ist dieses Format hierfür nur bedingt passend. Trotzdem können Sie Signaltabellen im txt-Format exportieren und importieren, es werden dann die Informationen aus der Signaltabelle herausgelesen, die in einer SHM-Kopplung vorhanden sind bzw. berücksichtigt werden können.

Zusätzlich zum txt-Format können auch SIMATIC-Symboltabellen im asc-, seq- und xlsx-Format importiert werden. Achten Sie aber speziell in diesem Fall darauf, dass in der SHM-Kopplung die Adressen für Ein- und Ausgänge nicht überlappend sein dürfen.

Hinweis

Wenn Sie die Signaltabelle zum Bearbeiten in Excel öffnen, müssen alle Zellen als "Text" formatiert sein, damit von Excel keine ungewollten Formatkonvertierungen vorgenommen werden.

Weitere Informationen hierzu finden Sie in den Abschnitten:

- Signaltabelle (Seite 202)
- Dialogfeld "Import von Signaleigenschaften" (Seite 967)
- Dialogfeld "Export von Signaleigenschaften" (Seite 970)

2.11 PRODAVE-Kopplung

2.11.1 Funktionsweise der PRODAVE-Kopplung

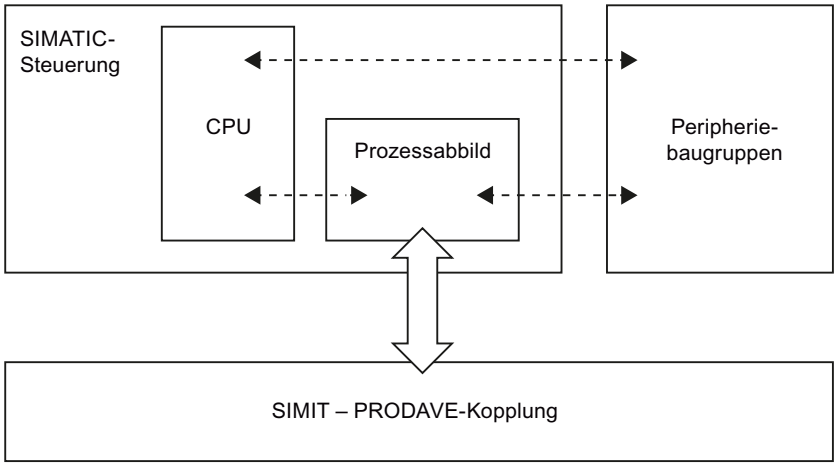
Mit der PRODAVE-Kopplung kommuniziert SIMIT mit einer SIMATIC-Steuerung über die MPI-Schnittstelle. Für Ethernet-fähige Steuerungen kann die Verbindung auch über Ethernet erfolgen. Die Verbindung über Ethernet ist etwas performanter als die Verbindung über MPI.

Stellen Sie im SIMATIC Manager die PG-Schnittstelle auf die gewählte Verbindungsart ein.

Hinweis

Für den Einsatz der PRODAVE-Kopplung benötigen Sie zusätzlich die SIMATIC-Software PRODAVE in der Version 6.2. Diese Software ist nicht im Lieferumfang von SIMIT enthalten. Eventuell benötigen Sie auch zusätzliche Hardware, wie beispielsweise MPI-Adapter oder Verbindungskabel. Diese sind ebenfalls nicht im Lieferumfang von SIMIT enthalten.

Die PRODAVE-Kopplung simuliert keine SIMATIC-Peripherie. Über die PRODAVE-Kopplung wird von SIMIT direkt das Prozessabbild der SIMATIC-Steuerung angesprochen. Welche Adressen damit tatsächlich ansprechbar sind, hängt davon ab, wie groß das Prozessabbild Ihrer CPU ist bzw. von Ihnen in HW-Konfig projektiert wurde.



Hinweis

Benutzen Sie an Ihrer Steuerung keine Peripheriebaugruppen, die dieselben Adressbereiche nutzen wie die PRODAVE-Kopplung. Dies würde zu konkurrierenden Zugriffen auf das Prozessabbild führen.

Hinweis

Wenn Sie in HW-Konfig Peripheriebaugruppen projektiert haben, können Sie diese Adressbereiche von der PRODAVE-Kopplung aus ansprechen, solange diese projektierten Baugruppen nicht wirklich gesteckt sind. Es werden dann in der CPU Zugriffsfehler provoziert, die Sie aber durch Anlegen der entsprechenden OBs im Steuerprogramm abfangen können. Da die PRODAVE-Kopplung nur Zugriff auf das Prozessabbild hat, können Sie grundsätzlich auf keine simulierten Signale zugreifen, die in der CPU als Peripheriesignale (*PEW, PAW*) adressiert sind. Einige SIMATIC-Steuerungen, wie beispielsweise die CPU 313C, sind fest mit E/A-Baugruppen verbunden, deren Adressbereiche nicht veränderbar sind. Diese Adressbereiche können über die PRODAVE-Kopplung nicht angesprochen werden.

2.11.2 Konfigurieren der PRODAVE-Kopplung

2.11.2.1 Anlegen einer PRODAVE-Kopplung

Um eine PRODAVE-Kopplung anzulegen, wählen Sie im Dialogfeld "Auswahl" die Option "PRODAVE" aus.

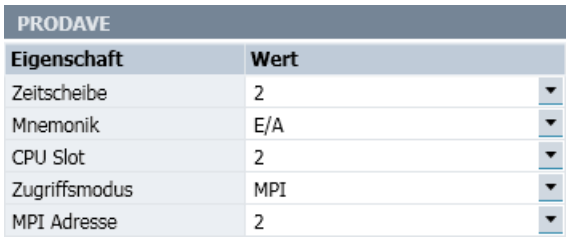
2.11.2.2 Editieren von Signalen in der PRODAVE-Kopplung

Um die Ein-/Ausgangssignale der Kopplung einzugeben, gibt es folgende Möglichkeiten:

- Sie geben die Signale manuell im Kopplungseditor ein.
- Sie importieren die Symboltabelle aus Ihrem SIMATIC-Projekt. Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Import von Signaleigenschaften" (Seite 967).

2.11.2.3 Eigenschaften der PRODAVE-Kopplung

Nach dem Öffnen der Kopplung wird der Kopplungseditor im Arbeitsbereich angezeigt. Im Eigenschaftsfenster können die folgenden Eigenschaften definiert werden:



PRODAVE	
Eigenschaft	**Wert**
Zeitscheibe	2
Mnemonik	E/A
CPU Slot	2
Zugriffsmodus	MPI
MPI Adresse	2

- **Zeitscheibe**
  Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, entsprechend einem Zyklus von 100 ms.

  Hinweis

  Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

- **Mnemonik**
  Hier wählen Sie aus, ob für die Bezeichnung der Ein- und Ausgänge die internationale (I/Q) oder deutsche (E/A) Schreibweise verwendet werden soll.
- **CPU Slot**
  Tragen Sie hier die Slot-Nummer der SIMATIC-CPU ein.
- **Zugriffsmodus**
  Wählen Sie hier aus, ob Sie über MPI (Kabel oder Adapter) oder IP (Ethernet) zugreifen möchten.
- **MPI Adresse / IP-Adresse**
  Tragen Sie hier die MPI-Nummer / IP-Adresse der SIMATIC-CPU ein.

Weitere Funktionen

Kopieren und Einfügen von Normierungen

Weitere Informationen hierzu finden Sie im Abschnitt: Normierung auf anderes Signal übertragen (Seite 216).

Import von Signaleigenschaften

Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Import von Signaleigenschaften" (Seite 967).

Export von Signaleigenschaften

Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Export von Signaleigenschaften" (Seite 970).

2.11.2.4 Importieren der Signaleigenschaften

Importieren Sie die Signaleigenschaften, indem Sie im Kopplungseditor auf das Symbol "" klicken.
Das Dialogfeld "Import von Signaleigenschaften" öffnet sich. In diesem Dialogfeld nehmen Sie Einstellungen zum Import vor.

Weitere Informationen finden Sie im Abschnitt: Dialogfeld "Import von Signaleigenschaften" (Seite 967).

2.11.2.5 Exportieren der Signaleigenschaften

Exportieren Sie die Signaleigenschaften, indem Sie im Kopplungseditor auf das Symbol "" klicken.
Das Dialogfeld "Export von Signaleigenschaften" öffnet sich. In diesem Dialogfeld nehmen Sie Einstellungen zum Export vor.

Siehe auch

Dialogfeld "Export von Signaleigenschaften" (Seite 970)

2.12 MCD-Kopplung

2.12.1 Funktionsweise der MCD-Kopplung

Funktionsweise

Mit der MCD-Kopplung kommuniziert SIMIT mit einer Mechatronics Concept Designer-Anwendung über eine Softwareschnittstelle von SIEMENS NX. SIMIT tauscht über die MCD-Kopplung zyklisch Daten des Signalhaushalts und der technologischen Objekte der gekoppelten MCD-Anwendung aus. Unterstützt wird die Kopplung zu genau einer MCD-Anwendung.



Für den Einsatz der MCD-Kopplung müssen folgende Anwendungen auf einem PC installiert sein:

- SIMIT
- Mechatronics Concept Designer

Wenn Sie die Simulation starten oder beenden, wird die MCD-Anwendung automatisch gestartet bzw. beendet.

Die Kopplung vom Typ "MCD" lässt zwei prinzipielle Arbeitsweisen zu.

- Eine MCD-Anwendung importieren.
  Wird die MCD-Anwendung importiert, werden alle dazugehörigen Dateien in das SIMIT-Projekt kopiert. SIMIT arbeitet anschließend ausschließlich mit der projektinternen Kopie. Änderungen an der originalen MCD-Anwendung wirken sich nicht auf die projektinterne Kopie aus.
- Eine MCD-Anwendung verknüpfen.
  Wird die MCD-Anwendung verknüpft, werden alle enthaltenen Signale in den SIMIT-Signalhaushalt übernommen. Alle von der MCD-Anwendung referenzierten Dateien verbleiben an ihrem originalen Ablageort. Sie werden nicht Teil des SIMIT-Projektes und folglich auch bei einer Projektarchivierung nicht berücksichtigt.

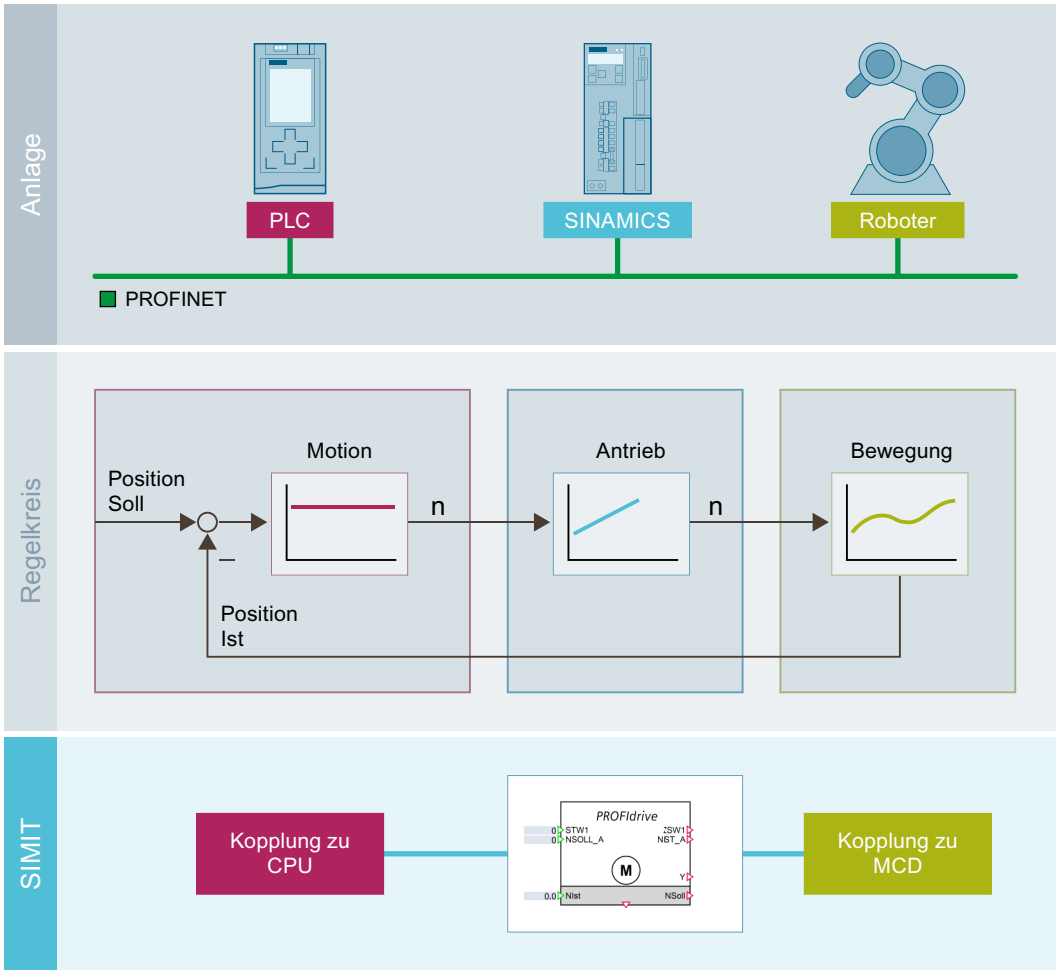
2.12.2 Einsatzszenarien der MCD-Kopplung

Funktionsprinzip

Für die Simulation der Ansteuerung einer Mechatronics Concept Designer-Anwendung benötigen Sie folgende Kopplungen in SIMIT:

- Kopplung zu einem Controller der SIMATIC-Familie
- MCD-Kopplung

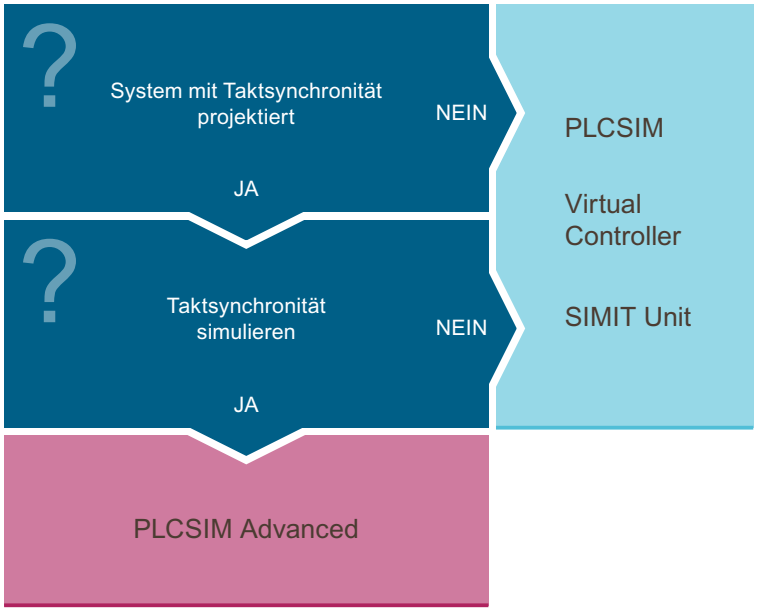
Die folgende Abbildung zeigt schematisch die Simulation einer Ansteuerung am Beispiel einer Positionsänderung einer Roboterachse:



Die "Kopplung zum Controller" übergibt den Ansteuerungsbefehl an die PROFIdrive-Komponente. Die PROFIdrive-Komponente berechnet aus dem Ansteuerungsbefehl die Soll-Position. Die Soll-Position wird über die MCD-Kopplung an die Mechatronics Concept Designer-Anwendung übergeben. Die Mechatronics Concept Designer-Anwendung enthält das CAD-Modell des Roboters und dessen physikalische Eigenschaften wie das Gewicht. Mit einer Physik-Engine zur Modellierung der physikalischen Gegebenheiten berechnet Mechatronics Concept Designer-Anwendung die tatsächliche Bewegung des Bauteils. Anschließend gibt die Mechatronics Concept Designer-Anwendung die berechnete Ist-Position zurück.

Regelkreis

Wenn Sie ein System mit Taktsynchronität projektiert haben und wenn Sie die Taktsynchronität simulieren möchten, verwenden Sie PLCSIM Advanced. Andernfalls verwenden Sie PLCSIM, Virtual Controller oder SIMIT Unit.



PLCSIM Advanced, SIMIT und MCD entsprechen in der Anlage PLC, SINAMICS und Roboter.

Mechatronics Concept Designer-Anwendung mit projektierter Taktsynchronität simulieren

1. SIMIT-Projekt erstellen
2. In Projekteigenschaften als Betriebsart "Bus-synchron" wählen.
3. Eine Zeitscheibe mit der Taktzeit parametrieren
4. PLCSIM Advanced-Kopplung projektieren
5. MCD-Kopplung projektieren
6. PROFIdrive-Komponente projektieren
7. PROFIdrive-Komponente mit Signalen der beiden Kopplungen verschalten

2.12.3 Konfigurieren der MCD-Kopplung

2.12.3.1 MCD-Kopplung anlegen

Einleitung

In einem SIMIT-Projekt können Sie eine Kopplung vom Typ "MCD" anlegen.

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- Mechatronics Concept Designer Version 11.0 oder höher ist auf dem SIMIT-PC installiert. Beachten Sie, dass für eine Kopplung zu Mechatronics Concept Designer Version 12.0 mindestens das Servicepack 1 (Version 12.01) vorausgesetzt wird.

Vorgehen

Um eine Kopplung vom Typ "MCD" anzulegen, gehen Sie folgendermaßen vor:

1. Fügen Sie in der Projektnavigation unter "Kopplungen" eine neue Kopplung hinzu. Das Dialogfeld "Auswahl" wird geöffnet.
2. Aktivieren Sie den Kopplungstyp "MCD".

Ergebnis

Die Kopplung vom Typ "MCD" wird in der Projektnavigation angelegt.

2.12.3.2 Signale importieren

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- Eine Kopplung vom Typ "MCD" ist angelegt.

Vorgehen

Um eine MCD-Anwendung und deren Signale zu importieren, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie die Kopplung vom Typ "MCD" in der Projektnavigation. Der Kopplungseditor der MCD-Kopplung wird geöffnet.
2. Starten Sie den Signalimport durch Klick auf die Schaltfläche "Import" aus der Menüzeile des Kopplungseditors. Das Dialogfeld "MCD Teiledatei auswählen" wird geöffnet.
3. Wählen Sie die Hauptdatei Ihrer MCD-Anwendung.
4. Um die MCD-Anwendung und deren Signale zu importieren, klicken Sie auf "Importieren".

Ergebnis

Die MCD-Anwendung inklusive aller referenzierten Dateien wird vollständig in das SIMIT-Projekt kopiert. Die in der MCD-Anwendung enthaltenen Signale werden im SIMIT-Signalhaushalt angelegt.

2.12.3.3 Signale verknüpfen

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- Eine Kopplung vom Typ "MCD" ist angelegt.

Vorgehen

Um eine MCD-Anwendung und deren Signale zu verknüpfen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie die Kopplung vom Typ "MCD" in der Projektnavigation.
   Der Kopplungseditor der MCD-Kopplung wird geöffnet.
2. Starten Sie die Signalverknüpfung durch Klick auf die Schaltfläche "Verknüpfen" aus der Menüzeile des Kopplungseditors.
   Das Dialogfeld "MCD Teiledatei auswählen" wird geöffnet.
3. Wählen Sie die Hauptdatei Ihrer MCD-Anwendung.
4. Um die MCD-Anwendung und deren Signale zu verknüpfen, klicken Sie auf "Verknüpfen".

Ergebnis

Die in der MCD-Anwendung enthaltenen Signale werden im SIMIT-Signalhaushalt angelegt. Die von der MCD-Anwendung referenzierten Dateien befinden sich weiterhin an ihrem originalen Ablageort.

2.12.3.4 Signaleigenschaften in der MCD-Kopplung bearbeiten

Einleitung

Die Kopplung vom Typ "MCD" lässt eine Konfiguration der physikalischen Einheit jedes ausgetauschten Signales zu. Die konfigurierte physikalische Einheit gilt für die Verwendung innerhalb des Simulationsmodells in SIMIT. Die physikalische Einheit desselben Signals in der MCD-Anwendung kann dabei durchaus unterschiedlich sein.

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- Eine Kopplung vom Typ "MCD" ist angelegt.
- Auf dem SIMIT-PC ist SIEMENS NX in der Version 12.0.1 oder höher installiert.

Vorgehen

Um die physikalische Einheit eines Signals der MCD-Kopplung zu konfigurieren, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie die Kopplung vom Typ "MCD" in der Projektnavigation.
   Der Kopplungseditor der MCD-Kopplung wird geöffnet.
2. Wählen Sie das gewünschte Signal aus dem Bereich "Eingangs- / Ausgangssignal" des Kopplungseditors aus.
   Das Signal ist hervorgehoben.
3. Wählen Sie in der Spalte "Einheit" die gewünschte physikalische Einheit.
4. Speichern Sie die Änderungen im Kopplungseditor.

Ergebnis

Die physikalische Einheit des gewünschten Signals der MCD-Kopplung wurde geändert.

2.12.3.5 Bus-synchrone MCD-Kopplung konfigurieren

Voraussetzung

- MCD-Kopplung ist angelegt.
- Betriebsart "Bus-synchron" ist eingestellt.
- Eine Zeitscheibe ist mit der parametrierten Taktzeit aus STEP 7 parametriert.

Vorgehen

Um eine Bus-synchrone MCD-Kopplung zu konfigurieren, gehen Sie folgendermaßen vor:

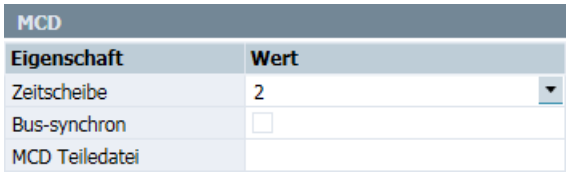
1. Doppelklicken Sie in der Projektnavigation unter "Kopplungen" auf "MCD-Kopplung".
2. Konfigurieren Sie die MCD-Kopplung im Eigenschaftsfenster:
   - Wählen Sie die Zeitscheibe, die mit der Taktzeit aus STEP 7 parametriert ist.
   - Aktivieren Sie die Option "Bus-synchron".

Ergebnis

Die Bus-synchrone MCD-Kopplung ist konfiguriert.

2.12.3.6 Eigenschaften der MCD-Kopplung

Im Eigenschaftsfenster können die folgenden Eigenschaften definiert werden:



Zeitscheibe

Hier wird der Zyklus eingestellt, mit dem die Kopplung Daten austauscht. Die Zuordnung absoluter Zykluszeiten zu den 8 möglichen Zeitscheiben ist für das gesamte Projekt gültig. Voreingestellt ist die Zeitscheibe 2, entsprechend einem Zyklus von 100 ms.

Hinweis

Die Zeitscheibe mit der kleinsten Zykluszeit hat immer die höchste Priorität, unabhängig von der Nummerierung.

Bus-synchron

Legt fest, ob die MCD-Anwendung mit der SIMIT-Modellberechnung synchronisiert wird.

Wenn Sie hier bus-synchron einstellen, müssen Sie in der gewählten Zeitscheibe die parametrierte Taktzeit projektieren.

Wenn Sie hier bus-synchron einstellen, müssen Sie auch im Eigenschaftsfenster im Projektmanager bus-synchron einstellen. Weiterführende Informationen über die bus-synchrone Betriebsart finden Sie unter Betriebsarten (Seite 43).

MCD Teiledatei

Zeigt die referenzierte MCD Teiledatei (Hauptdatei Ihrer MCD-Anwendung) an.

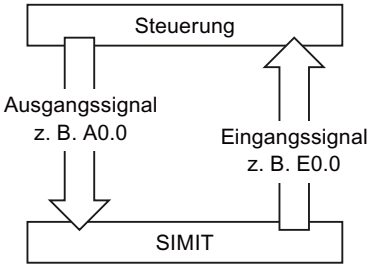
2.13 Signale einer Kopplung bearbeiten

2.13.1 Grundlagen zu Signalen

2.13.1.1 Datenrichtung der Kopplung

Jede Kopplung in SIMIT definiert Signale, die mit Simulationskomponenten verknüpft werden können. Die Begriffe "Eingang" und "Ausgang" werden von der angeschlossenen Steuerung aus gesehen:

- Ein Eingangssignal ist ein Signal, das in der Simulation berechnet bzw. von ihr ausgegeben und von der Steuerung eingelesen wird.
- Ein Ausgangssignal ist ein Signal, das von der Steuerung ausgegeben und einen Eingangswert für die Simulation gibt.



Alle in einer Kopplung enthaltenen Signale sind im Kopplungseditor aufgeführt. Der Zugriff auf ein Signal aus der Kopplung erfolgt über den Symbolnamen oder über die absolute Adresse und zwar unabhängig davon, wie das Signal verwendet wird.

2.13.1.2 Bedeutung des Kopplungsnamens

Signalnamen setzen sich aus einer Quelle und einem Namen zusammen. Bei Signalen aus Kopplungen entspricht der Kopplungsname der Quelle und der Symbolname bzw. die Absolutadresse dem Namen des Signals.

Quelle	Name
Profibus	A32.0

So bleiben Signalnamen im gesamten SIMIT-Projekt eindeutig, auch wenn in mehreren Kopplungen der gleiche Symbolname oder die gleiche Adresse auftauchen.

Hinweis

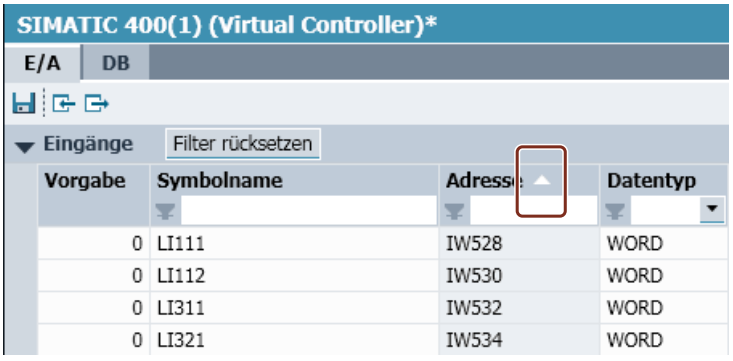
Jede Kopplung in einem SIMIT-Projekt muss einen eindeutigen Namen besitzen.

Wenn Sie z. B. SIMIT mit mehreren PROFIBUS DP-Mastern koppeln wollen, legen Sie für jedes PROFIBUS DP-Mastersystem eine eigene Kopplung an und vergeben eindeutige Kopplungsnamen. So werden alle Kopplungssignale der richtigen Kopplung zugeordnet.

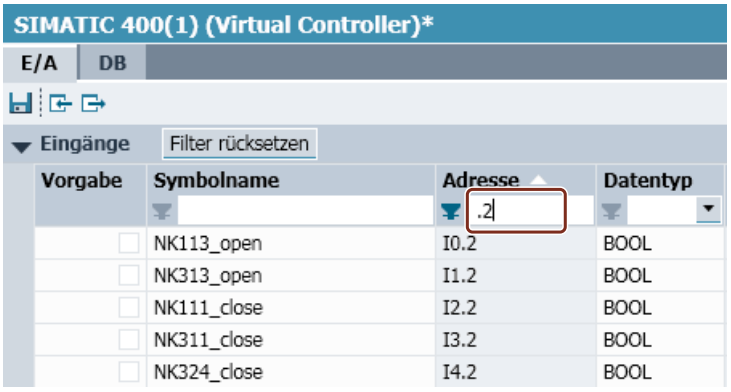
2.13.1.3 Sortieren und Filtern von Signalen im Kopplungseditor

Signale können nach allen in den einzelnen Spalten dargestellten Eigenschaften sortiert und gefiltert werden. Gehen Sie hierzu folgendermaßen vor:

- Klicken Sie auf die Überschrift der Spalte, nach der die Sortierung erfolgen soll.
  Es erscheint ein Pfeil in der Spaltenüberschrift:



- Mit jedem weiteren Klick auf die Spaltenüberschrift wird zwischen alphabetisch aufsteigender und absteigender Sortierung umgeschaltet. Der Pfeil ändert entsprechend seine Ausrichtung:
  - für aufsteigende Sortierung
  - für absteigende Sortierung
- Um die Anzahl der angezeigten Signale weiter zu reduzieren, können Sie zusätzlich in jeder Spalte einen Filter setzen. Es werden dann nur die Signale angezeigt, die allen Filterkriterien entsprechen.
  In der folgenden Abbildung ist ein Textfilter für die Adresse von Signalen gesetzt:



- Wählen Sie einen Filter direkt über eine Auswahl in einem Klappmenü oder geben Sie einen Text ein.

2.13.1.4 Adressierung von Signalen

In einem SIMATIC-Automatisierungsgerät wird zwischen dem Prozessabbild (E/A) und den Peripherieadressen (PE/PA) mit entsprechenden Zugriffsmechanismen unterschieden. Für SIMIT selbst ist diese Unterscheidung im Zugriff nicht relevant. Bei Verwendung der SIMATIC-Kopplungen in SIMIT können Sie entweder *AW* oder *PAW* bzw. *EW* oder *PEW* in der Adresse des E/A-Signals verwenden. Der Signalaustausch erfolgt abhängig vom Typ der Kopplung wie folgt:

- Bei der SIMIT Unit-Kopplung erfolgt der Signalaustausch mit SIMIT über Peripheriesignale der Feldgeräte wie in einer realen Anlage. Für die Steuerung gibt es keinen Unterschied zum Datenaustausch mit realer Peripherie. In der SIMATIC-Steuerung kann damit auf die Peripherie oder das Prozessabbild mit den entsprechenden Mechanismen zugegriffen werden.
- Bei Verwendung der PRODAVE-Kopplung tauscht SIMIT E/A-Signale nur mit dem Prozessabbild der Steuerung aus. Zugriffe in der SIMATIC-Steuerung auf Peripherieadressen führen zu Zugriffsfehlern.
- Für die PLCSIM-Kopplung sind die Mechanismen analog zur SIMIT Unit-Kopplung zu sehen: SIMIT kommuniziert mit PLCSIM auf Basis der Peripheriesignale. Die steuerungsseitigen Zugriffsmechanismen werden von PLCSIM koordiniert.

2.13.1.5 Fixieren von Signalen im Kopplungseditor

Einleitung

Signale können auf einem bestimmten Wert festgehalten und damit "fixiert" werden. Beim Fixieren werden zunächst die aktuell anstehenden Werte der Signale übernommen, um ein stoßfreies Umschalten zu erreichen.

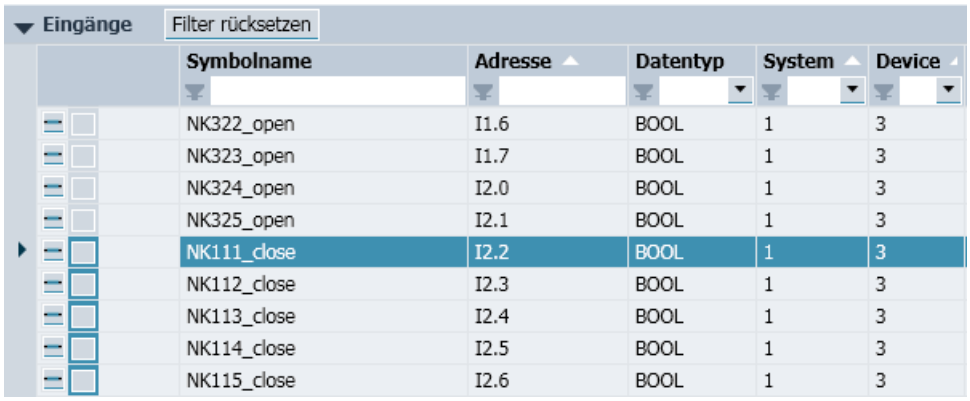
Für die Darstellung der Fixierung gibt es im Kopplungseditor für jedes Signal ein automatisch erzeugtes Control:

	"Umschalter mit Fixierung" für Signale vom Datentyp "Binär"
	"Digitaleingabe mit Fixierung" für Signale vom Datentyp "Analog" und "Integer"

Hinweis

Das Festhalten eines bestimmten Signalzustandes wird bei einem SIMATIC-Controller "Forcen" genannt. Das Fixieren in SIMIT ist eine vergleichbare Funktion, die sich aber nur innerhalb des Simulationsmodells auswirkt und kein "Forcen" im SIMATIC-Controller bewirkt.

Die Controls mit Fixierung stehen nach dem Starten der Simulation für jedes Signal in der Kopplung in der ersten Spalte zur Verfügung:



Eingänge | Filter rücksetzen

	Symbolname	Adresse	Datentyp	System	Device
	NK322_open	I1.6	BOOL	1	3
	NK323_open	I1.7	BOOL	1	3
	NK324_open	I2.0	BOOL	1	3
	NK325_open	I2.1	BOOL	1	3
	NK111_close	I2.2	BOOL	1	3
	NK112_close	I2.3	BOOL	1	3
	NK113_close	I2.4	BOOL	1	3
	NK114_close	I2.5	BOOL	1	3
	NK115_close	I2.6	BOOL	1	3

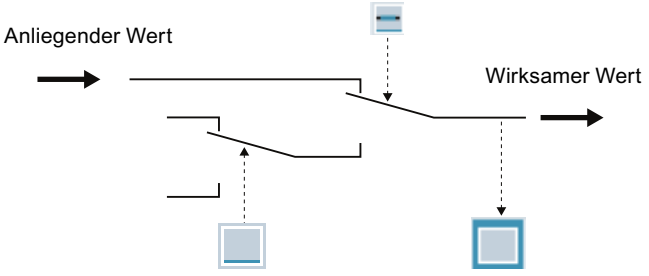
Funktionsweise des Umschalters mit Fixierung

Der Umschalter mit Fixierung für Binärwerte vereint drei Funktionen in sich:

Symbol	Bedeutung
	Fixierschalter nicht aktiv, der aktuelle Wert des Signals wird angezeigt Fixierschalter aktiv, der fixierte Wert wird angezeigt und ist bedienbar
	Dieses Symbol wird nur bei binären Werten angezeigt: Wert ist "0" Wert ist "1"
25	Eingabe- und Anzeigefeld für Werte vom Datentyp "Analog" und "Integer"

Wenn der Fixierschalter und/oder der Wert nicht bedienbar ist, ist das entsprechende Symbol grau hinterlegt, z. B. wenn das Kopplungssignal im Simulationsmodell nicht verwendet wird.

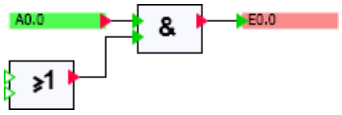
Die folgende Abbildung zeigt die Funktionsweise eines Umschalters mit Fixierung:



Die Fixierung stellt im Prinzip eine Umschaltung dar, die entweder den anliegenden Wert in der Kopplung durchreicht oder auf einen manuell vorzugebenden Wert umschaltet. Sie ist sowohl bei Eingangs- als auch Ausgangssignalen möglich, d. h. alle Peripheriesignale können fest vorgegeben werden.

Beispiel zur Fixierung

Im folgenden Beispiel sind zwei Signale in der Kopplung projektiert und in folgender Weise auf einem Plan verwendet:



In der Kopplung können Sie sehen, welcher Wert des Signals A0.0 von der Steuerung ausgegeben wird:

Von der Steuerung ausgegebener Wert	A0.0 in der Kopplung	Im Simulationsmodell wirksamer Wert
0		0
1		1

Durch Fixierung können Sie bewirken, dass Ihr Simulationsmodell mit einem anderen Wert rechnet, als er von der Peripherie ausgegeben wird. Klicken Sie dazu auf den Fixierschalter des Signals *A0.0*. Dieser wechselt daraufhin in die Darstellung . In diesem Zustand können Sie den Wert, der in Ihrem Simulationsmodell wirksam sein soll, mit dem Umschalter direkt wie folgt vorgeben:

Von der Steuerung ausgegebener Wert	A0.0 in der Kopplung	Im Simulationsmodell wirksamer Wert
0 oder 1		1
0 oder 1		0

Entsprechendes gilt für Eingangssignale. Für ein Eingangssignal E0.0 können Sie zunächst sehen, welchen Wert das Simulationsmodell an die Steuerung ausgibt:

Im Simulationsmodell berechneter Wert	E0.0 in der Kopplung	An die Steuerung ausgegebener Wert
0		0
1		1

Mithilfe der Fixierung können Sie den Wert, der an die Steuerung ausgegeben wird, unabhängig von dem im Simulationsmodell berechneten Wert fest vorgeben. Geben Sie hierfür mit dem Fixierschalter den Vorgabewert vor:

Im Simulationsmodell berechneter Wert	E0.0 in der Kopplung	An die Steuerung ausgegebener Wert
0 oder 1		1
0 oder 1		0

Die Ausführungen gelten analog für Analog- und Integer-Signale. Nur die Vorgabewerte werden, im Unterschied zu den Binärwerten, mit Hilfe einer Digitaleingabe eingegeben. Anstelle der Binäranzeige wird eine Digitalanzeige verwendet.

Siehe auch

Signal fixieren (Seite 214)

2.13.1.6 Abbildung von SIMATIC-Datentypen in SIMIT

Ein Signal in den SIMATIC-Kopplungen hat einen der SIMATIC-spezifischen Datentypen *BOOL, BYTE, WORD, INT, DWORD, DINT* oder *REAL* entsprechend dem Zugriff auf den Adressbereich des Automatisierungsgeräts. In der Verwendung als Peripheriesignal hat es den Datentyp binary, integer oder analog. Folgende Tabelle zeigt die Abbildung der SIMATIC-Datentypen auf die Datentypen des Peripheriesignals.

Tabelle 2-3 Abbildung der SIMATIC-Datentypen auf die Signaldatentypen

SIMATIC-Datentyp	Wertebereich der SIMATIC	SIMIT-Datentyp	Abbildung auf SIMATIC-Datentypen
BOOL	true, false	Binary	true, false
BYTE	0 .. 255	Integer	[−128 .. 255]
WORD (ohne Normierung)	0 .. 65535	Integer	[−32768 .. 65535]
INT	−32768 .. 32767	Integer	[−32768 .. 65535]
DWORD	0 .. 4294967295	Integer	[−2147483648 .. 4294967295]
DINT	−2147483648 .. 2147483647	Integer	[−2147483648 .. 4294967295]
REAL	$\pm 1.175495 \times 10^{-38}$ $\pm 3.402823 \times 10^{38}$	Analog	$\pm 1.175495 \times 10^{-38}$ $\pm 3.402823 \times 10^{38}$

Werte, die in der SIMIT-Kopplung eingelesen werden, werden entsprechend dem oben aufgeführten Wertebereich der SIMATIC-Datentypen in die SIMIT-Datentypen übernommen.

Alle Werte, die von SIMIT über die Kopplung ausgegeben werden und den Wertebereich der SIMATIC einhalten, werden von SIMIT unverändert ausgegeben. Werte, die außerhalb dieses Intervalls liegen, werden auf das in der letzten Spalte angegebene Intervall begrenzt und dann auf die Datenbreite des entsprechenden SIMATIC-Datentyps abgebildet. Negative Zahlen werden als Zweierkomplement gespeichert. Beispielsweise wird die Zahl −1 als Byte mit dem gleichen Bitmuster übertragen wie die Zahl 255.

Eine etwas andere Behandlung erfahren normierte Signale. Rohwerte, die in der SIMIT-Kopplung eingelesen werden, werden auf den Wertebereich der SIMATIC begrenzt und dann in physikalische Werte umgerechnet.

Werte, die von SIMIT über die Kopplung ausgegeben werden, werden in Rohwerte umgerechnet und anschließend auf den Wertebereich der SIMATIC begrenzt:

SIMATIC-Datentyp	Wertebereich der SIMATIC	SIMIT-Datentyp	Abbildung auf SIMATIC-Datentypen
WORD (bipolar normiert)	[−32768 .. 32767]	Analog	[−32768 .. 32767]
WORD (unipolar normiert)	[−32768 .. 32767]	Analog	[−32768 .. 32767]
WORD (benutzerspezifisch normiert)	[−32768 .. 32767]	Analog	[−32768 .. 32767]

Weitere Informationen hierzu finden Sie im Kapitel: Normierung von analogen Signalen (Seite 196).

2.13.1.7 Zugriff auf einen Datensatz oder Speicherbereich

Mit folgenden Kopplungstypen ist der Zugriff auf Datensätze in der dezentralen Peripherie oder auf den Speicherbereich eines SIMATIC-Controllers möglich:

Kopplungstyp	Zugriffsart	Komponente
SIMIT Unit Virtual Controller PLCSIM Advanced	Datensatz in dezentraler Peripherie lesen / schreiben	ReadDataRecord WriteDataRecord
Virtual Controller PLCSIM PRODAVE	Bytes aus Merkerbereich lesen / schreiben	ReadMemory WriteMemory
	Bytes aus Datenbaustein lesen / schreiben	ReadDataBlock WriteDataBlock

Dieser Zugriff erfolgt nicht über Signale, die im Kopplungseditor gelistet sind und zyklisch mit der Steuerung ausgetauscht werden, sondern mit Hilfe von Komponenten, die getriggert einen Schreib- oder Lesevorgang ausführen. Die dafür notwendigen Komponententypen finden Sie in der Basisbibliothek im Verzeichnis *COMMUNICATION | SIMATIC*.

Sie verknüpfen die Komponente mit der zugehörigen Kopplung einfach dadurch, dass Sie im Eigenschaftsfenster des Unit-Konnektors den Namen der Kopplung eintragen, auf die Sie mit dieser Komponente zugreifen möchten. Alternativ können Sie den Unit-Konnektor auch per Drag & Drop erzeugen, indem Sie das zu adressierende Modul aus dem Eigenschaftsfenster der Kopplung in das Diagramm ziehen.

Um diese Zugriffsarten für eine Kopplung nutzen zu können, müssen Sie die Kopplung einmal gespeichert haben. Öffnen Sie dazu die Kopplung im Editor, definieren Sie beispielsweise ein Ein- oder Ausgangssignal und speichern Sie dann die Kopplung.

Zugriff auf den Speicherbereich eines Controllers

Geben Sie im Eigenschaftsfenster des Unit-Konnektors den Kopplungsnamen ein. Verbinden Sie den Unit-Konnektor mit dem Eingang "Gateway" der Komponente.

Zugriff auf Datensatz in der dezentralen Peripherie

Geben Sie im Eigenschaftsfenster des Unit-Konnektors den Kopplungsnamen und die Adresse der Baugruppe mit folgender Syntax ein:

- [<Nummer des Mastersystems>][<Slave-Nummer>][<Slot-Nummer>]

Verbinden Sie den Unit-Konnektor mit dem Eingang "Unit" der Komponente.



Der Zugriff auf einen Datensatz in der dezentralen Peripherie ist sowohl mit PROFIBUS als auch mit PROFINET möglich.

Ein Beispiel zum Einsatz des Unit-Konnektors finden Sie unter Verknüpfung der SIWAREXU-Komponenten mit der Kopplung (Seite 547).

Bei folgenden Komponenten ist die Datensatzkommunikation standardmäßig integriert. Sie verschalten den Eingang Unit direkt mit einem Unit-Konnektor:

- SIWAREXU-Komponenten (Seite 545)
- ASM452 – Anschaltung für Identifikationssysteme (Seite 900)
- ASM454 – Anschaltung für Identifikationssysteme (Seite 903)
- ASM456 – Anschaltung für Identifikationssysteme (Seite 904)
- ASM473 – Anschaltung für Identifikationssysteme (Seite 907)
- ASM475 – Anschaltung für Identifikationssysteme (Seite 910)
- ASM754 – Anschaltung für Identifikationssysteme (Seite 914)
- ASM850 – Anschaltung für Identifikationssysteme (Seite 915)
- ASM854 – Anschaltung für Identifikationssysteme (Seite 917)
- RF170C – Anschaltung für Identifikationssysteme (Seite 919)
- RF180C – Anschaltung für Identifikationssysteme (Seite 922)

Siehe auch

ReadMemory – Lesen eines Merkerbereichs (Seite 568)

WriteMemory – Schreiben eines Merkerbereichs (Seite 568)

ReadDatablock – Lesen eines Datenbausteins (Seite 569)

WriteDatablock – Schreiben eines Datenbausteins (Seite 570)

ReadDataRecord – Lesen eines Datensatzes (Seite 571)

WriteDataRecord – Schreiben eines Datensatzes (Seite 571)

Unit-Konnektor (Seite 411)

Datensatzkommunikation (Seite 126)

2.13.1.8 Umwandeln der Datenbreite von Signalen

Einleitung

Beim Import von Systembausteinen werden die zur importierten Konfiguration gehörenden Signale automatisch in der Kopplung angelegt. Dabei wird unterschieden, ob es sich um Binärsignale mit der Datenbreite "Bit" oder um Signale mit der Datenbreite "Byte", Wort (2 Byte), Doppelwort (4 Byte) oder andere handelt.

Sie können die Datenbreite ändern, indem Sie z. B. acht aufeinander folgende Binärsignale zu einem Byte-Signal zusammenfassen.

Sie können nur Signale wandeln, die in der Kopplung durch den Import der Systembausteine angelegt worden sind. Die Adressbereiche sind durch die Projektierung festgeschrieben. Sie sind nicht veränderbar, auch nicht durch Wandlung der Datenbreite der Signale.

Umwandeln von Signalen

Die folgende Tabelle zeigt, wie Sie Signale wandeln können:

- Signale zusammenfassen: Lesen Sie eine Tabellenzeile von links nach rechts
- Signal aufspalten: Lesen Sie eine Tabellenzeile von rechts nach links

Anzahl Signale	Datentyp	Datenbreite		Anzahl Signale	Datentyp	Datenbreite
8	BOOL	1 Bit	↔	1	BYTE	1 Byte
2	BYTE	1 Byte	↔	1	WORD (INT)	2 Byte
2	WORD (INT)	2 Byte	↔	1	DWORD (DINT, REAL)	4 Byte

Siehe auch

Signal aufspalten (Seite 211)

Signale zusammenfassen (Seite 212)

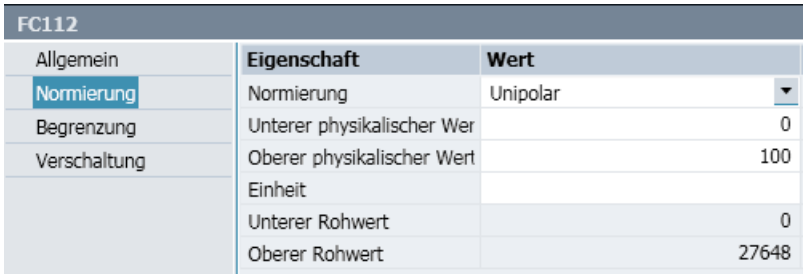
2.13.1.9 Normierung von analogen Signalen

In SIMATIC-Systemen wie auch üblicherweise in anderen Automatisierungssystemen werden die Werte von Analogsignalen in ein festes, ganzzahliges Format umgewandelt. Der Wertebereich wird aus der Auflösung der A/D-Wandler abgeleitet und liegt bei SIMATIC S7 üblicherweise im Bereich von −27648 bis +27648.

In der Simulation werden Analogwerte als Gleitkommazahlen behandelt. Physikalische Größen wie Drücke, Temperaturen etc. werden in der Simulation in der Regel mit ihren absoluten Werten angesetzt. Als Eingangswerte für die angeschlossenen Automatisierungssysteme sind diese Größen dann nicht nur in das Festkommaformat zu wandeln, sondern über die entsprechenden Messbereiche auf den Wertebereich abzubilden. Bei der Übertragung von SIMIT zur SIMATIC sind die Analogwerte also zu normieren bzw. bei entgegengesetzter Übertragung zu denormieren.

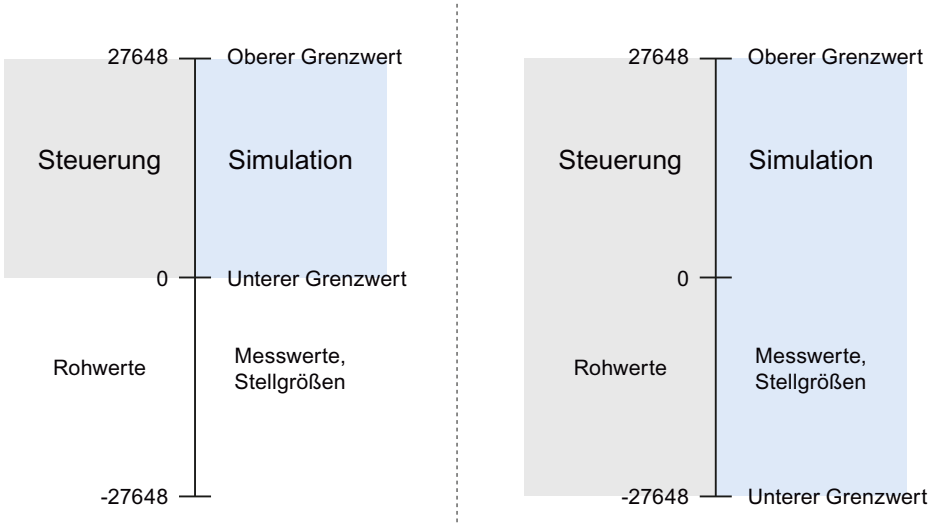
Die Normierung von Eingangssignalen und Denormierung von Ein- und Ausgangssignalen ist somit eine Anpassung der Simulation an die Eigenheiten der Steuerung. Deshalb erfolgen Normierung und Denormierung an der Schnittstelle zwischen Simulationsmodell und Steuerung in den Kopplungen.

Die Normierung für ein Signal definieren Sie entweder über dessen Eigenschaften oder über die Signaltabelle.



Da im Baugruppenspektrum der SIMATIC alle Analogbaugruppen mit einer Auflösung von 2 Byte arbeiten, die als ein Wort übertragen werden, ist in SIMIT die Normierung und Denormierung nur für Analogwerte des Datentyps *WORD* vorgesehen.

Die folgende Abbildung zeigt das Schema der Normierung beispielhaft für unipolare und bipolare Messungen:



Unipolare Normierung/Denormierung

Bipolare Normierung/Denormierung

- Unipolare Messungen liefern nur positive Rohwerte (0 bis 27648),
- bipolaren Messungen liefern sowohl positive, als auch negative Rohwerte (−27648 bis +27648).

In der folgenden Tabelle sind die Normierungen/Denormierungen aufgelistet, die von SIMIT unterstützt werden:

Normierungstyp		Messbereich		Rohwerte	
Nummer	Name	Anfang	Ende	unten	oben
0	keine Normierung				
1	Unipolar	0 (Default)	100	0	27648
2	Bipolar	−100 (Default)	100 (Default)	−27648	27648
3	Benutzerdefiniert	0 (Default)	100 (Default)	0 (Default)	27648 (Default)
4	PT x00 Standard	−200 °C	850 °C	−2000	8500
5	PT x00 Klima	−120 °C	130 °C	−12000	13000
6	Ni x00 Standard	−60 °C	250 °C	−600	2500
7	Ni x00 Klima	−60 °C	250 °C	−6000	25000
8	Cu 10 Standard	−200 °C	260 °C	−2000	2600
9	Cu 10 Klima	−50 °C	150 °C	−5000	15000
10	Thermoelement Typ B	0 °C	1820 °C	0	18200
11	Thermoelement Typ E	−270 °C	1000 °C	−2700	10000
12	Thermoelement Typ J	−210 °C	1200 °C	−2100	12000
13	Thermoelement Typ K	−270 °C	1372 °C	−2700	13720
14	Thermoelement Typ L	−200 °C	900 °C	−2000	9000
15	Thermoelement Typ N	−270 °C	1300 °C	−2700	13000
16	Thermoelement Typ R, S	−50 °C	1769 °C	−500	17690
17	Thermoelement Typ T	−270 °C	400 °C	−2700	4000
18	Thermoelement Typ U	−200 °C	600 °C	−2000	6000

Für die Typen 1 und 2 sind der "Untere physikalische Wert" und der "Obere physikalische Wert" mit 0 bzw. −100 und 100 vorbelegt. Diese Werte können aber an die Mess- oder Stellbereiche angepasst werden.
Unipolare und bipolare Normierungen (Typen 1 und 2) und der Typ "Benutzerdefiniert" (Typ 3) sind als Denormierungen auch für Ausgangssignale anwendbar.
Die Normierungstypen 4 bis 18 sind als Normierungen für Temperaturmessungen nur für Eingangssignale anwendbar.

Hinweis

Für die Temperaturmess-Signale in einer Kopplung werden die Messbereichsgrenzen nur angezeigt. Sie sind nicht editierbar.

Obere und untere Rohwerte sind nur editierbar, wenn der Normierungstyp "Benutzerdefiniert" ausgewählt wurde.

Ändern der Normierung bei laufender Simulation

Bei laufender Simulation können die Messbereiche der Normierung und die Rohwertbereiche (beim Typ "Benutzerdefiniert") geändert werden. Die Änderungen werden sofort übernommen.

Hinweis

Diese Funktion ist bei der Virtual Controller-Kopplung nicht verfügbar.

Hinweis

Die Änderungen beziehen sich nur auf die laufende Simulation. Sie werden nicht automatisch in die Projektierung übernommen.

Übertragung von Gleitkommawerten (Float)

Einige Peripheriegeräte – unter anderem aus dem PROFIBUS PA-Spektrum – übertragen ihre Messwerte direkt an die Steuerung als physikalische Größen im Gleitkommaformat mit einer Datenbreite von 4 Byte (Doppelwort). Um diese Gleitkommawerte korrekt darzustellen, müssen die entsprechenden Signale in der Kopplung auf den Datentyp *REAL* eingestellt werden.

Siehe auch

Signaltabelle (Seite 202)

Signal normieren (Seite 215)

2.13.2 Import und Export der Signale

2.13.2.1 Dateiformate für Signale

Mit dem Exportieren und Importieren von Signaleigenschaften können Sie Signalkonfigurationen zwischen Kopplungen austauschen. Signale einer Kopplung werden in bestimmten Formaten exportiert und importiert. Folgende Dateiformate werden abhängig vom Kopplungstyp unterstützt:

Kopplungstyp	Signaltabellen	
	Importformat[1]	Exportformat
SIMIT Unit	asc, seq, txt oder xlsx	txt, seq
Virtual Controller[2]	asc, seq, txt oder xlsx	txt, seq
PLCSIM	asc, seq, txt oder xlsx	txt, seq
PRODAVE	asc, seq, txt oder xlsx	txt, seq
SHM	asc, seq, txt oder xlsx	txt, seq

Kopplungstyp	Signaltabellen	
	Importformat[1]	Exportformat
OPC DA-Server	ini, txt	txt, ini
OPC DA-/UA-Client	ini, txt	txt, ini

[1] Unterstützt werden nur Dateien, die als SBCS (Single Byte Character Set) kodiert sind und die fest eingestellte Codepage 1252 ("Western European") benutzen.

[2] Für den Import und Export von Datenbausteintabellen wird nur das Format "*.txt" unterstützt.

Weitere Informationen um Signalimport und -export:

- Standardformat ist das txt-Format.
- Eine Signaltabelle im asc-Format bzw. seq-Format entspricht der Symboltabelle eines SIMATIC-Projekts.
  Weitere Informationen hierzu finden Sie im Abschnitt: Symboltabelle (Seite 200).
- Exportierte Signaltabellen enthalten alle Signale einer Kopplung mit deren Eigenschaften wie Name (Symbol), Adresse, Kommentar etc.
  Weitere Informationen hierzu finden Sie im Abschnitt: Signaltabelle (Seite 202).

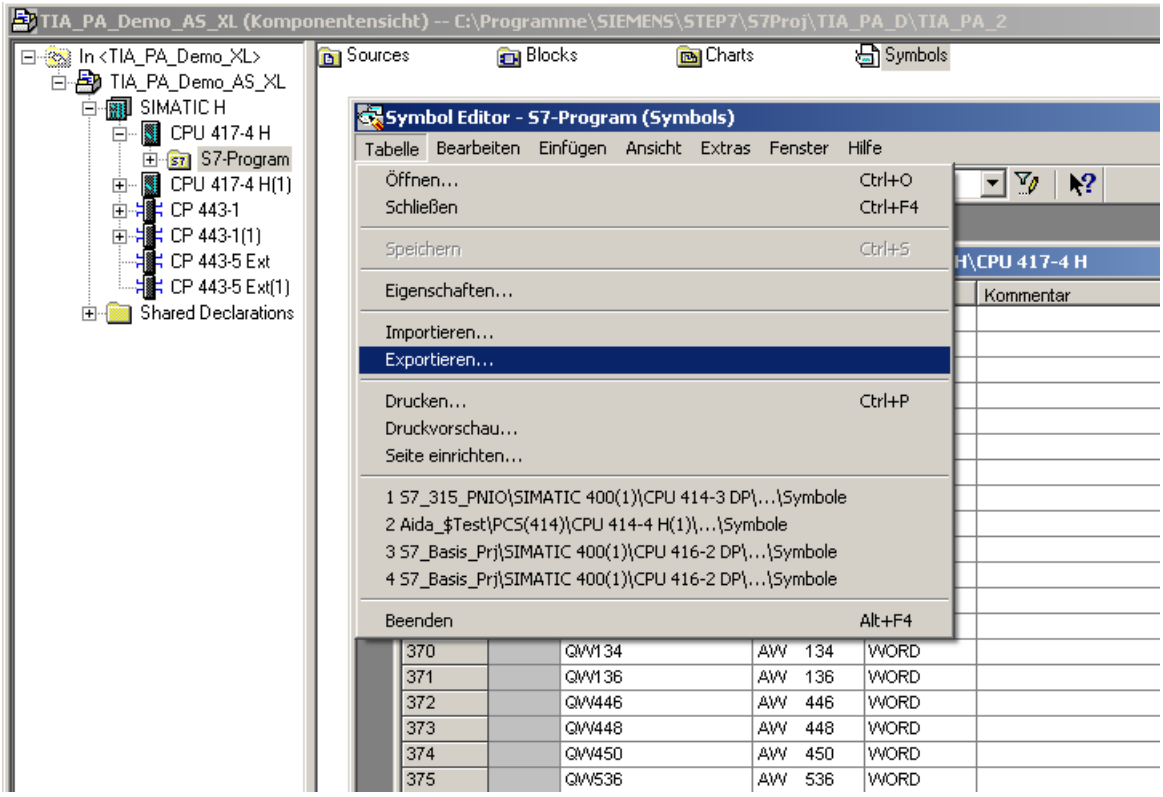
2.13.2.2 Symboltabelle

Die Symboltabelle wird aus dem SIMATIC-Manager aus einem SIMATIC-Projekt in verschiedenen Formaten exportiert. Für den Import nach SIMIT wählen Sie das asc-Format oder das seq-Format.

Hinweis

Beachten Sie, dass im seq-Format der Kommentar auf 40 Zeichen begrenzt ist und dass keine Datentypen enthalten sind.

Datenworte erhalten bei Import des seq-Formates in SIMIT immer den Datentyp *WORD*, Doppelworte den Datentyp *DWORD*. Umstellungen auf die Datentypen *INT* oder *REAL* müssen dann manuell vorgenommen werden.



Der Import der Symboltabelle wird in SIMIT von allen SIMATIC-Kopplungen unterstützt.

Hinweis

Das asc-Format benutzt feste Spaltenbreiten. Wenn Sie diese Datei nicht mit dem Symboleditor, sondern mit einem anderen Texteditor bearbeiten, darf die Anzahl Zeichen pro Zeile nicht verändert werden.

2.13.2.3 Variablentabelle

Die PLC-Variablenlisten (Symboltabellen) werden aus dem TIA-Portal importiert. Das TIA-Portal exportiert diese Liste im xlsx-Format. Es werden nur Ein- und Ausgangssignale mit den folgenden Datentypen übernommen:

TIA-Portal	STEP 7	SIMIT
Bool	BOOL	Binär
Byte	BYTE	integer
Word	WORD	integer
Int	INT	integer
DWord	DWORD	integer
Dint	DINT	integer
Real	REAL	analog

2.13.2.4 Signaltabelle

Die Signaltabelle hat folgenden Aufbau:

Spalte	Bezeichnung	Überschrift	Bedeutung
1	Symbol	Symbol	Symbolischer Name des Signals
2	E/A	InOut	Kennzeichen *E, A, EB, AB, EW, AW, ED, AD* bzw. in entsprechender internationaler Schreibweise (I/Q)
3	Adresse	Address	Absolute Adresse des Signals, z. B. *0.0* oder *512*
4	Typ	Type	Datentyp des Signals: *BOOL, BYTE, WORD* oder *DWORD*
5	Kommentar	Comment	Text als Kommentar
6	Vorbelegung	Default	Initialisierungswert, mit dem dieses Signal vorbelegt ist
7	Signalquelle	ImplicitSource	Signalquelle des implizit verschalteten Signals
8	Signalname	ImplicitSignal	Signalname des implizit verschalteten Signals
9	Einheit	Unit	Einheit des Signals, z. B. physikalische Einheit bei Messwerten
10	Normierung	Scaling	Typnummer der Normierung (Informationen hierzu finden Sie in der Tabelle im Kapitel: Normierung von analogen Signalen (Seite 196))
11	Untergrenze	ScalingLowerPhys	Physikalische Untergrenze bzw. Obergrenze von Analogsignalen
12	Obergrenze	ScalingUpperPhys	
13	Unterer Rohwert	ScalingLowerRAW	Unterer bzw. oberer Rohwert normierter Analogsignale. Nur beim Normierungstyp "Benutzerdefiniert" frei wählbar.
14	Oberer Rohwert	ScalingUpperRAW	
15	Begrenzung Ein/Aus	LimitActive	Begrenzung eines Analogsignals ist wirksam/unwirksam
16	Untere Begrenzung	LimitLowerPhys	Unterer physikalischer Wert der Begrenzung
17	Obere Begrenzung	LimitUpperPhys	Oberer physikalischer Wert der Begrenzung
18	Multiplikator	Multiplier	OPC-Client, Zyklusmultiplikator
19	Rücklesbar	Readback	OPC-Client, rücklesbares Signal

Im txt-Format wird der Tabulator als Spaltentrennzeichen benutzt.

In der ersten Zeile muss die Kennung "#Signal properties;" stehen. Diese Zeile enthält Informationen zum Dateityp. Daran werden beim Export aus SIMIT noch die SIMIT-Versionsnummer, mit der diese Tabelle exportiert wurde, und der Kopplungsnamen angefügt.

In der zweiten Zeile stehen die ausgewählten Eigenschaften als Spaltenüberschriften. Diese sind auch dann vorhanden, wenn kein Inhalt dazu verfügbar ist.

In den folgenden Zeilen wird jeweils ein Signal mit seinen Eigenschaften beschrieben.

Hinweis

Die Signaltabelle wurde mit der SIMIT-Version 8.1 überarbeitet und mit Spaltenüberschriften erweitert. Wenn Sie eine Signaltabelle importieren, die mit einer älteren SIMIT-Version erstellt wurde, müssen Sie diese entsprechend anpassen.

Durch den neuen Normierungstyp "Benutzerdefiniert" hat sich die Bedeutung der Typnummern der Normierung verschoben.

Hinweis

Wenn Sie eine Signaltabelle in EXCEL bearbeiten, müssen alle Spalten als "Text" formatiert sein, damit keine ungewollten Formatkonvertierungen von EXCEL vorgenommen werden.

2.13.2.5 INI-Format der OPC-Kopplungen

OPC-Signale haben aus den folgenden Gründen ein eigenes Import/Export-Format:

- Sie kennen keine Adressierung.
- Die Übertragungsrichtung kann nicht am Namen erkannt werden.
- Eine Umwandlung zwischen Rohwerten und physikalischen Werten findet nicht statt.

Die Signaltabelle für OPC-Signale hat das ini-Format. Das Format besteht aus sechs Sektionen, die mit folgenden Schlüsselwörtern gekennzeichnet sind:

Schlüsselwort	Bedeutung
[AIN]	Analoge Eingangssignale
[IIN]	Ganzzahlige Eingangssignale
[BIN]	Binäre Eingangssignale
[AOUT]	Analoge Ausgangssignale
[IOUT]	Ganzzahlige Ausgangssignale
[BOUT]	Binäre Ausgangssignale

Die Schlüsselwörter müssen jeweils alleine in einer Zeile stehen. Dann folgen die Signalnamen, die dieser Sektion zugeordnet werden sollen. Die Reihenfolge der Sektoren ist beliebig. Nicht alle Sektionen müssen verwendet werden.

In jeder Sektion sind die zugehörigen Signale zeilenweise in der Signaltabelle abgelegt. Die Signale mit ihren Eigenschaften werden in Anführungszeichen und durch ein Semikolon getrennt exportiert.

	OPC-Server	OPC-Client
Eingänge	• Name • Kommentar • Vorbelegung • Implizite Verschaltung (Quelle/Ziel) • Implizite Verschaltung (Signal)	• Name • Kommentar • Vorbelegung • Implizite Verschaltung (Quelle/Ziel) • Implizite Verschaltung (Signal) • Multiplikator • Rücklesbar ("True/False")
Ausgänge	• Name • Kommentar • Implizite Verschaltung (Quelle/Ziel) • Implizite Verschaltung (Signal)	• Name • Kommentar • Implizite Verschaltung (Quelle/Ziel) • Implizite Verschaltung (Signal) • Multiplikator

Wenn beim Import der Signaltabelle in einer Zeile nur ein Signalname ohne Anführungszeichen steht, wird das Signal mit diesem Namen, ohne Kommentar und mit der Standardvorbelegung übernommen.

Hinweis

Beim OPC-Client ist das Import-Format von SIMIT V8.1 nicht abwärtskompatibel zu älteren SIMIT-Versionen. Wenn Sie den Multiplikator bzw. die Rücklesbarkeit in bereits vorhandenen Import-Dateien angegeben haben, müssen Sie jetzt 2 zusätzliche leere Spalten für die implizite Verschaltung ergänzen.

2.13.2.6 Signaleigenschaften importieren

Einleitung

Signaleigenschaften können Sie aus folgenden Dateien importieren:

- Symboltabelle
- Variablentabelle
- Signaltabelle
- INI-Datei

Voraussetzung

- Projekt ist geöffnet.
- Import-Datei liegt vor.
- Kopplung ist angelegt.
- Kopplungseditor ist geöffnet.

Vorgehen

Um Signaleigenschaften zu importieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie im Kopplungseditor das Dialogfeld zum Importieren der Signaleigenschaften.
2. Wählen das Format der Import-Datei.
3. Wählen Sie die Import-Datei aus.
4. Wählen Sie die zu importierenden Signaleigenschaften aus.
5. Wählen Sie über den "Modus" aus, wie mit Eigenschaften bereits vorhandener Signale verfahren wird.
6. Aktivieren Sie bei Bedarf "Datenbreite automatisch anpassen".
7. Klicken Sie auf "Importieren".

Ergebnis

Die Signaleigenschaften werden importiert.

Siehe auch

Dialogfeld "Import von Signaleigenschaften" (Seite 967)

2.13.2.7 Import von PLC-Variablen-Listen

Aus dem TIA-Portal werden PLC-Variablen-Listen (Symboltabellen) importiert. Diese Listen werden vom TIA-Portal im EXCEL-Format (*.xlsx) exportiert. Aus der Variablenliste werden nur E/A-Signale mit den in der folgenden Tabelle aufgeführten Datentypen übernommen:

Tabelle 2-4 Datentypen im Vergleich

TIA-Portal	Step 7	SIMIT
Bool	BOOL	binär
Byte	BYTE	integer
Word	WORD	integer
Int	INT	integer
DWord	DWORD	integer
DInt	DINT	integer
Real	REAL	analog

2.13.2.8 Signaleigenschaften exportieren

Voraussetzung

- Projekt ist geöffnet.
- Kopplung ist angelegt.
- Kopplungseditor ist geöffnet.

Vorgehen

Um Signaleigenschaften zu exportieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie im Kopplungseditor das Dialogfeld zum Exportieren der Signaleigenschaften.
2. Wählen das Format der Export-Datei.
3. Wählen Sie den Ablageort der Export-Datei aus.
4. Wählen Sie die zu exportierenden Signaleigenschaften aus.
5. Klicken Sie auf "Exportieren".

Ergebnis

Die gewählten Signaleigenschaften werden exportiert.

Siehe auch

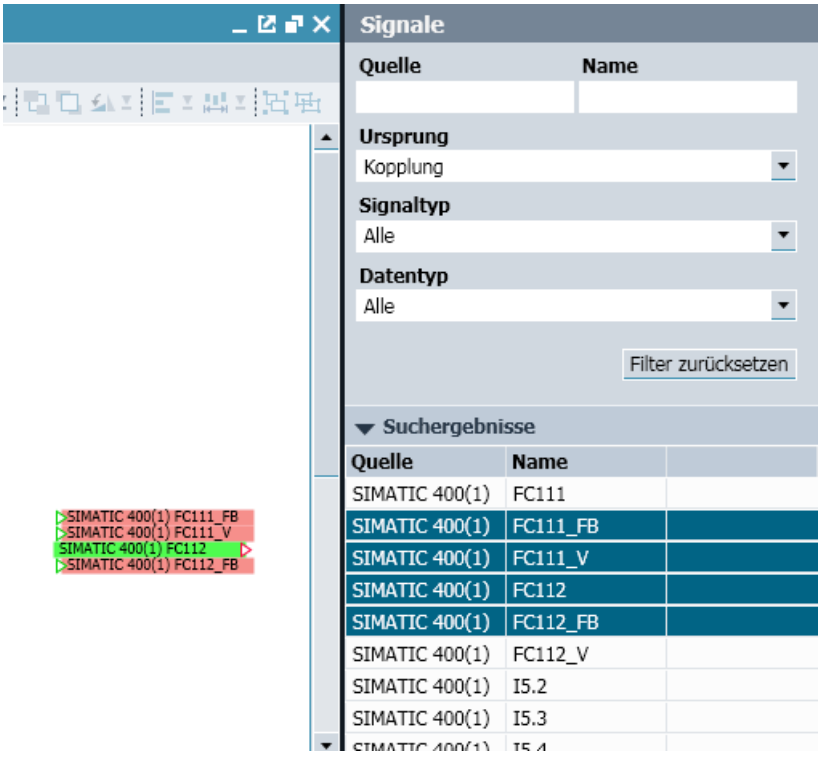
Dialogfeld "Export von Signaleigenschaften" (Seite 970)

2.13.3 Verwendung von Peripheriesignalen

2.13.3.1 Peripheriesignale zur Verschaltung auf Diagrammen

Um Peripheriesignale in das Simulationsmodell zu übernehmen, können sie auf Diagrammen verwendet werden.
Gehen Sie hierzu folgendermaßen vor:

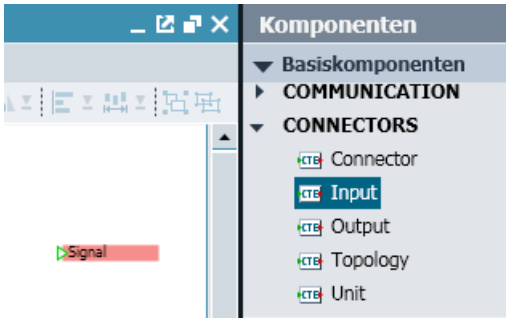
1. Öffnen Sie das gewünschte Diagramm im Diagrammeditor.
2. Öffnen Sie die Task-Card "Signale".
   Stellen Sie den Filter "Ursprung" auf "Kopplung", um nur die Peripheriesignale anzeigen zu lassen.
   Weitere Informationen zur Task-Card finden Sie im Abschnitt: Task-Card "Signale" (Seite 242).
3. Selektieren Sie ein oder mehrere Signale in der Task-Card und ziehen Sie sie mit Drag & Drop auf das Diagramm.
   Auf dem Diagramm werden jetzt Controls für die Bedienung der Signale erzeugt.
   Wenn Sie stattdessen Peripherie-Konnektoren für die Signale erzeugen wollen, halten Sie beim Herausziehen zusätzlich <Shift> gedrückt. Dadurch werden Peripherie-Konnektoren erzeugt, die bereits mit den richtigen Signalnamen versehen sind.



Die Peripherie-Konnektoren können jetzt beliebig mit einem Anschluss einer anderen Komponente verbunden werden.

Peripherie-Signale aus der Task-Card "Komponenten" ziehen

Peripherie-Konnektoren können auch aus der Task-Card "Komponenten" verwendet werden, der Kopplungs- und Signalnamen muss dann aber von Hand eingetragen werden.



Um die Namen einzutragen, klicken Sie zur Selektion auf den Peripherie-Konnektor und tragen Sie im Eigenschaftsfenster Kopplungsname und Signalname ein.

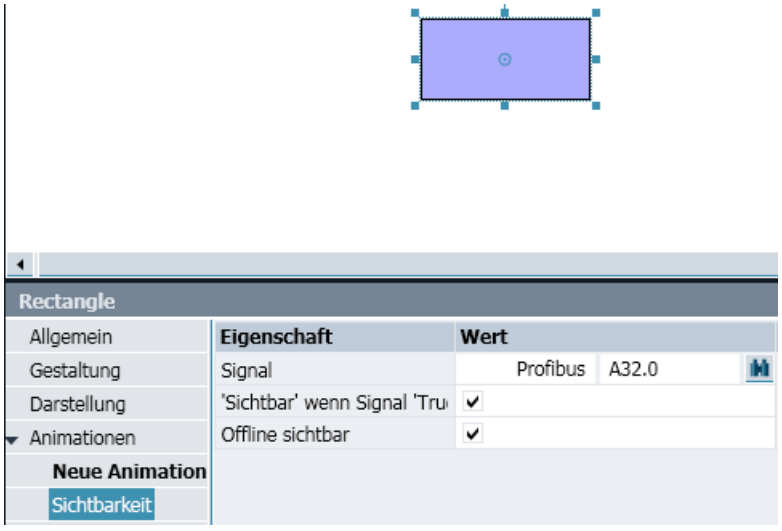


Hinweis

Der Name des Peripherie-Konnektors muss genau so geschrieben werden, wie er im Konfigurator eingetragen ist. Beachten Sie deshalb die Groß-/Kleinschreibung und fügen Sie in Adressbezeichnungen wie beispielsweise *E0.0* keine Leerzeichen ein.

2.13.3.2 Peripheriesignale für Animationen

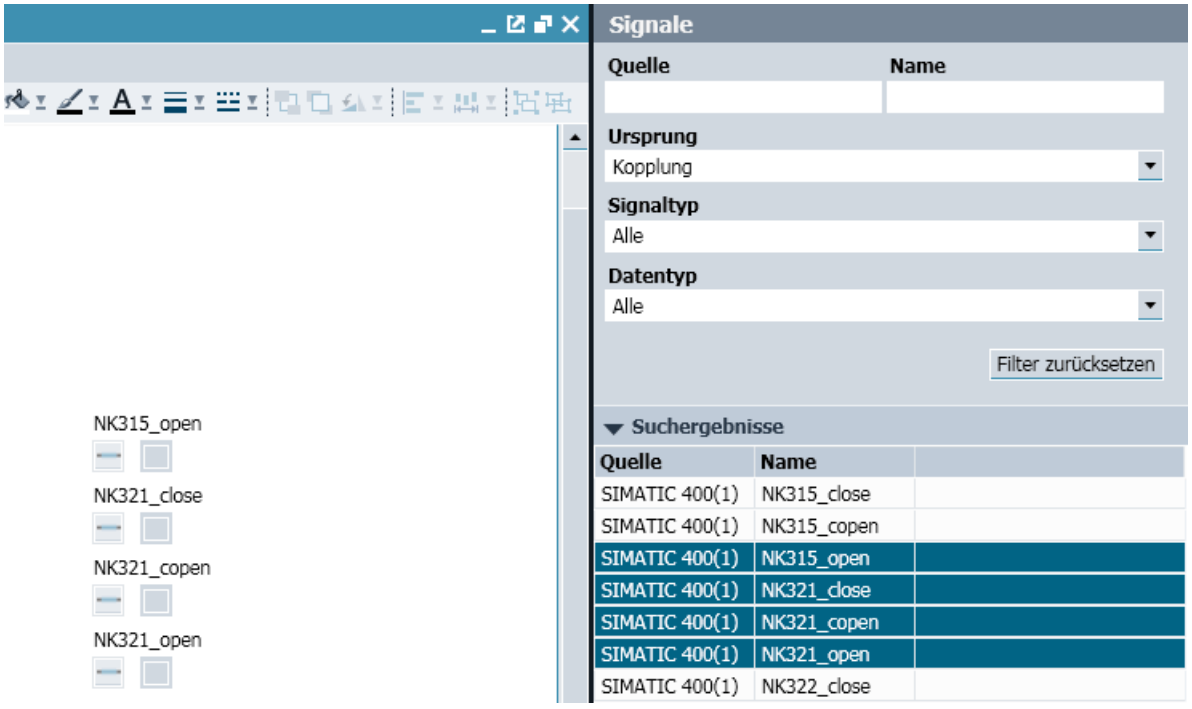
Sie können Peripheriesignale benutzen, um Grafikobjekte zu animieren. Tragen Sie dazu den Signalnamen im Eigenschaftsfenster der Animation ein.



Übernehmen Sie die Signale per Drag & Drop aus der Task-Card "Signale".

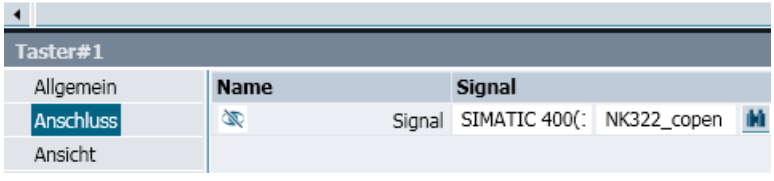
2.13.3.3 Peripheriesignale zur Bedienung auf Diagrammen

Die fixierbaren Controls können Sie aus der Kopplung auf das Diagramm übernehmen. Dazu ziehen Sie die Peripheriesignale aus der Task-Card "Signale" per Drag & Drop auf das Diagramm. Der Name zum Control wird als Textelement auf das Diagramm übernommen.



Sie können Peripheriesignale auch direkt in anderen Controls, wie z. B. einem Taster verwenden.



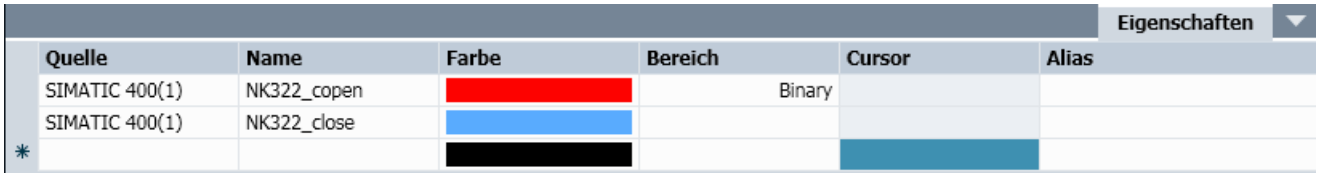


Tragen Sie dazu den Namen des Peripheriesignals in die Eigenschaftsmaske des Controls ein oder übernehmen Sie das Peripheriesignal per Drag & Drop aus der Task-Card "Signale" in das Eigenschaftsfenster des Controls.

2.13.3.4 Peripheriesignale in Kurvenbildern

Sie können Peripheriesignale auch in Kurvenbildern verwenden. Gehen Sie hierzu folgendermaßen vor:

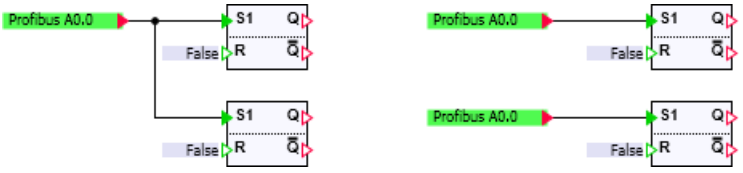
- Tragen Sie das Signal mit Quelle und Namen im Eigenschaftsfenster des Kurvenbildes ein.
- Ziehen Sie das Signal aus der Task-Card "Signale" per Drag & Drop in das Eigenschaftsfenster des Kurvenbildes.



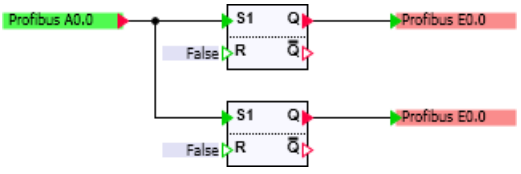
Weitere Informationen zu den Kurvenbildern finden Sie im Kapitel: Kurvenbilder (Seite 290).

2.13.3.5 Mehrfachverwendung von Peripherie-Konnektoren

Peripherie-Eingangskonnektoren gleichen Namens können mehrfach auf einem Plan verwendet werden, das heißt die beiden folgenden Darstellungen sind gleichwertig:



Ein Peripherie-Ausgangskonnektor kann nur einmal im gesamten Projekt verwendet werden. Die folgende Verschaltung würde beim Starten des Simulationsprojekts zu einem Fehler führen:

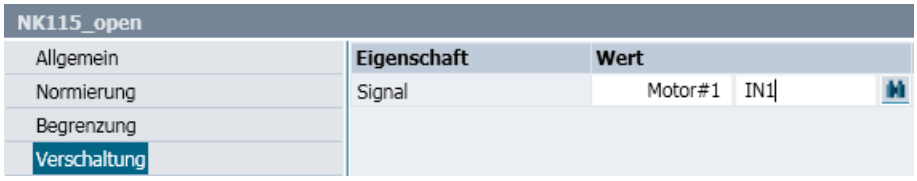


2.13.4 Signal verschalten

Einleitung

Eine Verschaltung ist ein Signalaustausch zwischen zwei Objekten:

- Zwei Kopplungssignalen
- Einem Kopplungssignal und dem Ein- oder Ausgang einer Komponente



Voraussetzung

- Kopplung ist angelegt.
- Kopplungseditor ist geöffnet.
- Signale sind angelegt.
- Zu verschaltende Signale sind vom gleichen Datentyp.
- Signal ist nicht verschaltet.

Vorgehen

Um ein Signal zu verschalten, gehen Sie folgendermaßen vor:

1. Wählen Sie das Signal.
2. Geben Sie in den Eigenschaften des Signals unter "Verschaltung" das Verschaltungsziel ein.

Ergebnis

Das Signal ist verschaltet.

2.13.5 Signal aufspalten

Einleitung

Wenn Sie ein Signal aufspalten, wird dessen symbolischer Name entfernt.

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller

- Kopplungseditor ist geöffnet.
- Für das aufzuspaltende Signal gilt:
  - Wurde aus einem Systembaustein importiert
  - Ist nicht fehlersicher

Vorgehen

Um ein Signal aufzuspalten, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Signaltabelle ein Signal mit einem der folgenden Datentypen aus:
   - BOOL
   - WORD
   - DWORD
2. Wählen Sie im Kontextmenü der Auswahl "Aufspalten".

Ergebnis

Das Signal wird abhängig vom Datentyp in bis zu acht Signale aufgespaltet.

Siehe auch

Umwandeln der Datenbreite von Signalen (Seite 196)

2.13.6 Signale zusammenfassen

Einleitung

Wenn Sie Signale zusammenfassen, werden deren symbolische Namen entfernt.

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller
- Kopplungseditor ist geöffnet.
- Für die zusammenzufassenden Signale gilt:
  - Wurden aus Systembausteinen importiert
  - Befinden sich im selben Modul
  - Sind nicht fehlersicher

Vorgehen

Um Signale zusammenzufassen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Signaltabelle aufeinander folgende Signale aus:
   - Datentyp BOOL: 8 Signale
   - Datentyp BYTE oder WORD: 2 Signale
2. Wählen Sie im Kontextmenü der Auswahl "Zusammenfassen".

Ergebnis

Die Signale werden zusammengefasst.

Siehe auch

Umwandeln der Datenbreite von Signalen (Seite 196)

2.13.7 Signal vorbelegen

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller
  - PLCSIM
  - PLCSIM Advanced
  - PRODAVE

Vorgehen

Um ein Signal mit einem Wert vorzubelegen, gehen Sie folgendermaßen vor:

1. Öffnen Sie den Kopplungseditor.
2. Belegen Sie unter "Eingänge > Vorgabe" das Signal vor:
   - Analoges Signal: Geben Sie den gewünschten Wert ein.
   - Binäres Signal: Aktivieren Sie das Optionskästchen.

Ergebnis

Beim Starten der Simulation wird das Signal mit dem eingegebenen Wert vorbelegt. Während der laufenden Simulation kann sich der Wert ändern.

Siehe auch

Signal fixieren (Seite 214)

2.13.8 Signal fixieren

Einleitung

Im Kopplungseditor können Sie bei laufender Simulation Signale setzen und Signalwerte anzeigen.

Voraussetzung

- Kopplung ist angelegt.
- Simulation ist gestartet.
- Kopplungseditor ist geöffnet.

Vorgehen

Um ein Signal zu fixieren, gehen Sie folgendermaßen vor:

1. Geben Sie abhängig vom Datentyp des Signals einen Wert ein oder aktivieren Sie das Optionskästchen.
2. Aktivieren Sie den Fixierungsschalter.

Ergebnis

Der Wert des Signals ist fixiert. Der Wert wird während der laufenden Simulation durch keinerlei Einflüsse mehr verändert.

Siehe auch

Fixieren von Signalen im Kopplungseditor (Seite 190)

Signal vorbelegen (Seite 213)

2.13.9 Signal symbolisch adressieren

Einleitung

In SIMIT werden Signale standardmäßig über die Adresse eindeutig identifiziert. Da die Adresse wenig über die Funktion eines Signals aussagt, können Sie bei einigen Kopplungen einen aussagekräftigen symbolischen Namen für ein Signal vergeben. Adressen oder symbolische Namen werden z. B. an Ein- und Ausgangskonnektoren angezeigt.

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller
  - PLCSIM
  - PLCSIM Advanced
  - PRODAVE

Vorgehen

Um ein Signal symbolisch zu adressieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie den Kopplungseditor.
2. Geben Sie unter "Symbolname" einen symbolischen Namen für das Signal ein.
3. Wählen Sie den Menübefehl "Extras > Kopplungssignale zuweisen".

Ergebnis

Die Identifikation des Signals ist von absoluter Adressierung auf symbolische Adressierung umgestellt. In SIMIT können Sie nur noch über den symbolischen Namen auf das Signal zugreifen.

Siehe auch

Extras > Kopplungssignale zuweisen (Seite 959)

2.13.10 Signal normieren

Einleitung

Statt in den Eigenschaften eines Signals können Sie Normierungswerte auch in der Signaltabelle eingeben.

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller
  - PLCSIM
  - PLCSIM Advanced
  - PRODAVE
- Signal ist analog.
- Signal ist vom Datentyp WORD.

Vorgehen

Um ein Signal zu normieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie den Kopplungseditor.
2. Wählen Sie das Signal.
3. Parametrieren Sie in den Eigenschaften des Signals unter "Normierung" die Normierungswerte:
   - Wählen Sie den Normierungstyp.
   - Geben Sie den unteren und oberen physikalischen Wert ein.

Ergebnis

Die Normierung für das Signal ist definiert.

Siehe auch

Signaltabelle (Seite 202)

Normierung von analogen Signalen (Seite 196)

Normierung auf anderes Signal übertragen (Seite 216)

2.13.11 Normierung auf anderes Signal übertragen

Einleitung

Normierungen können Sie von einem Signal auf ein oder mehrere andere Signale übertragen. Normierungen übertragen Sie im Kopplungseditor spaltenweise.

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller
  - PLCSIM
  - PLCSIM Advanced
  - PRODAVE
- Kopplungseditor ist geöffnet.
- Signal ist normiert.
- Am Zielsignal ist ein Normierungstyp eingestellt.

Vorgehen

Um die Normierung auf ein anderes Signal zu übertragen, gehen Sie folgendermaßen vor:

1. Wählen Sie das Signal in der Spalte "Normierung" aus.
2. Wählen Sie im Kontextmenü der Zelle "Zelle kopieren".
3. Wählen Sie das Zielsignal in der Spalte "Normierung" aus.
4. Wählen Sie im Kontextmenü der Zelle "Einfügen in Zelle".

Ergebnis

Die Normierung wird auf das Zielsignal übertragen:

- Normierungstyp
- Unterer und oberer physikalischer Wert

Siehe auch

Signal normieren (Seite 215)

2.13.12 Signal begrenzen

Einleitung

Sie können für alle analogen Kopplungssignale einen Grenzwertbereich definieren. Außerhalb dieses Grenzwertbereichs liegende Werte werden abhängig vom Signal nicht an den Kopplungspartner oder die Simulation übertragen:

- Grenzwertbereich für Eingangssignal: Kopplungspartner
- Grenzwertbereich für Ausgangssignal: Simulation

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - SIMIT Unit
  - Virtual Controller
  - PLCSIM
  - PLCSIM Advanced
  - PRODAVE
- Signal ist vom SIMIT-Datentyp "analog":
  - Normierter Messwert (WORD)
  - Gleitkommazahl (REAL)

Vorgehen

Um ein Signal zu begrenzen, gehen Sie folgendermaßen vor:

1. Öffnen Sie den Kopplungseditor.
2. Wählen Sie das Signal.
3. Parametrieren Sie in den Eigenschaften des Signals unter "Begrenzung" die Grenzwerte:
   - Schalten Sie die Begrenzung ein.
   - Geben Sie den unteren und oberen physikalischen Wert ein.

Ergebnis

Für das Signal ist ein Grenzwertbereich definiert. Während einer laufenden Simulation können Sie die Begrenzung deaktivieren oder die Grenzwerte ändern.

2.13.13 Signale von einem OPC-Server lesen

Hinweis

Beim Lesen von OPC-Signalen werden alle Signale im Kopplungseditor überschrieben. Wenn Sie an Signalen z. B. Zyklusmultiplikatoren oder Kommentare eingegeben haben, exportieren Sie die Signale vor dem Lesen. Wenn Sie die Signale vom OPC-Server gelesen haben, importieren Sie die Signale wieder.

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - OPC DA-Client
  - OPC UA-Client
- Verbindung zum OPC-Sever ist konfiguriert.
- OPC-Server ist im Netzwerk erreichbar.

Vorgehen

Um Signale von einem OPC-Server zu lesen, gehen Sie folgendermaßen vor:

1. Öffnen Sie den Kopplungseditor der OPC-Client-Kopplung.
2. Importieren Sie die OPC-Signale mit "Browse".

Ergebnis

Die OPC-Signale werden vom OPC-Server gelesen. Eingetragen werden nur OPC-Signale, deren Datentypen von SIMIT unterstützt werden. Ob ein Signal als "Eingang" oder "Ausgang" angelegt wird, ist abhängig von der Konfiguration des Signals am OPC-Server.

Siehe auch

Abbildung der Signale und Datentypen eines OPC-Servers (Seite 155)

OPC DA-Client-Kopplung konfigurieren (Seite 158)

OPC UA-Client-Kopplung konfigurieren (Seite 165)

2.13.14 Signal als "Rücklesbar" konfigurieren

Einleitung

Signale, die von SIMIT an einen OPC-Server übertragen werden, müssen vom OPC-Server nicht in jedem Fall akzeptiert und übernommen werden. Um im Simulationsmodell den im OPC-Server tatsächlich wirksamen Wert zur Verfügung zu haben, konfigurieren Sie das Signal als "Rücklesbar".

Voraussetzung

- Projekt ist geöffnet.
- Einer der folgenden Kopplungstypen ist angelegt:
  - OPC DA-Client
  - OPC UA-Client

- Kopplungseditor ist geöffnet.
- Signal ist am OPC-Server als "ReadWriteable" konfiguriert.
- Signal ist im Kopplungseditor unter "Eingänge" angelegt.

Vorgehen

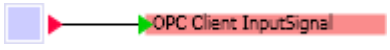
Um ein Signal als "Rücklesbar" zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie das Signal aus.
2. Aktivieren Sie in den Eigenschaften des Signals die Option "Rücklesbares Signal".

Ergebnis

Das Signal ist als "Rücklesbar" konfiguriert. Sie können das Eingangssignal in einem Output-Konnektor verwenden, um den tatsächlich auf dem OPC-Server wirksamen Wert zu lesen.

- Wert, der von SIMIT an den OPC-Server geschickt wird:



- Wert, der vom OPC-Server zurückgelesen wird:

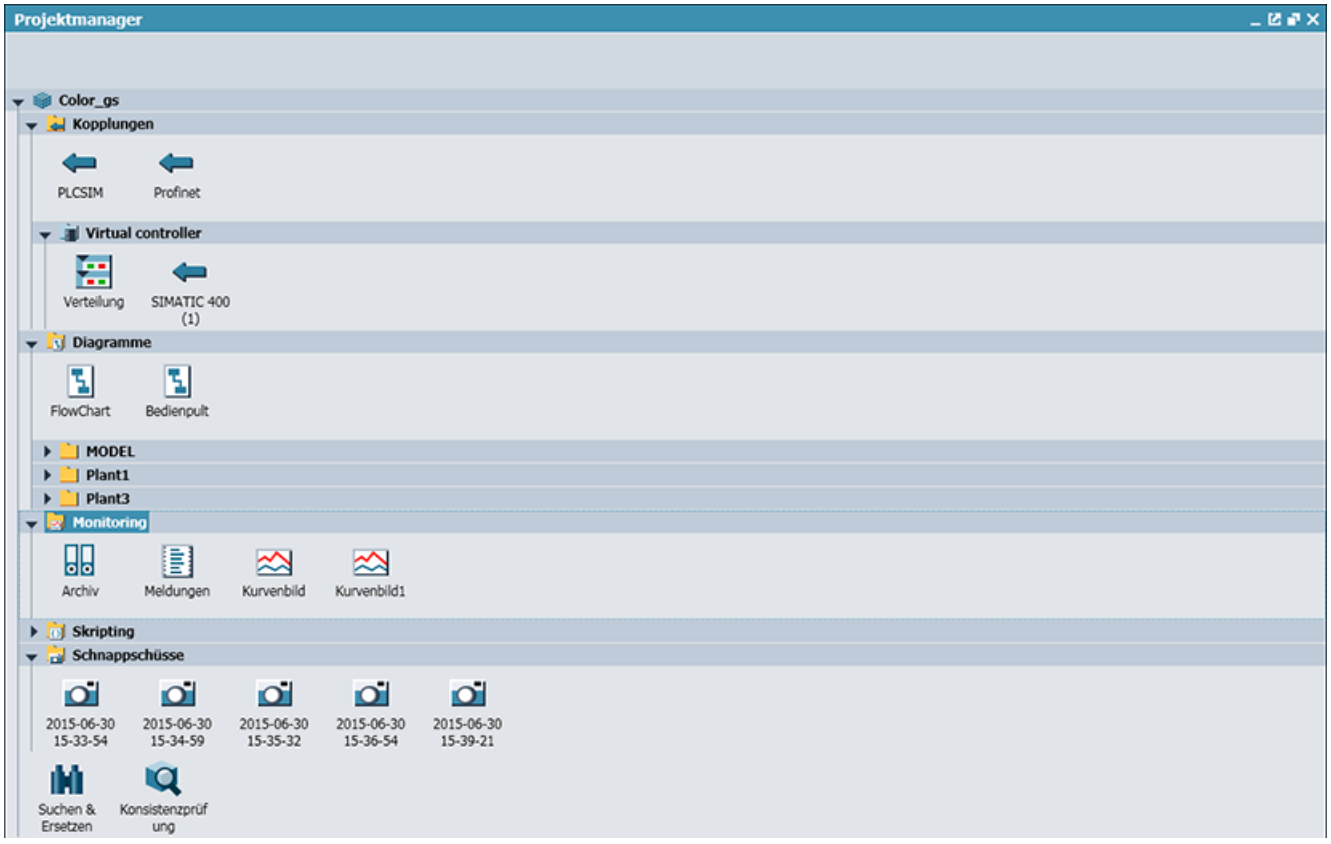


Simulationsmodell 3

3.1 Projektmanager

3.1.1 Ansicht und Funktionen des Projektmanagers

Sie öffnen den Projektmanager durch einen Doppelklick auf den Baumeintrag Projektmanager in der Projektnavigation.
Im Arbeitsbereich wird eine alternative Ansicht des Projekts zur Projektnavigation angezeigt:



Im Projektmanager haben Sie folgende Bearbeitungsmöglichkeiten:

- Per Drag & Drop Diagramme, Komponenten des Projekts und ganze Ordner verschieben
- Mit Kontextmenübefehlen Elemente kopieren, einfügen, löschen und umbenennen
- Mit Doppelklick Elemente des Projekts, z. B. Diagramme, zur Bearbeitung öffnen

Im Eigenschaftsfenster werden die folgenden Eigenschaften Ihres SIMIT-Projekts dargestellt und können ebenfalls bearbeitet werden:

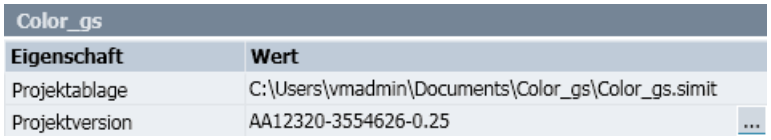
- Projektablage
  Der Ordner, in dem Sie das Projekt abgelegt haben. Diese Eigenschaft ist hier nicht veränderbar.
- Projektversion
  Weitere Informationen hierzu finden Sie im Kapitel: Versionierung (Seite 222).
- Eigenschaft "Schreibgeschützt"
  Weitere Informationen hierzu finden Sie im Kapitel: Schreibschutz (Seite 223).
- "Voreinstellung Maßstab", falls Sie die CONTEC Bibliothek nutzen
  Weitere Informationen hierzu finden Sie im Kapitel: Maßstäblichkeit (Seite 830).
- Acht verschiedene Zeitscheiben (Zeitscheibe 1 … Zeitscheibe 8)
  Für die acht Zeitscheiben, die Sie den Komponenten, Controls und Kopplungen in Ihrem Projekt zuordnen können, können Sie einen beliebigen Wert in Millisekunden vorgeben, mindestens jedoch 1 ms.
- Betriebsarten "Synchron", "Asynchron" und "Bus-synchron".
  Weitere Informationen hierzu finden Sie im Abschnitt: Betriebsarten (Seite 43).

3.1.2 Versionierung

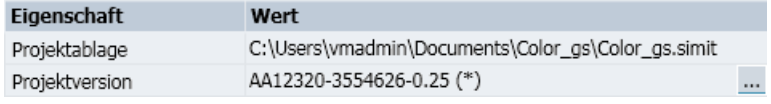
Ein Simulationsprojekt wird automatisch mit einer Versionskennung versehen. Sie setzt sich aus mehreren Informationen zusammen:

- Lizenznummer (z. B. AB12345)
- Zeitstempel (Auflösung beträgt ca. 38 Sekunden)
- Hauptversion (0 .. 127)
- Unterversion (0 .. 999)

Wenn das Simulationsprojekt verändert wurde, wird diese Versionskennung beim nächsten Starten der Simulation automatisch aktualisiert.



Wenn das Simulationsprojekt verändert, aber seitdem noch nicht wieder gestartet wurde, wird hinter der Versionsangabe ein Stern angezeigt.



Die Unterversion wird bis auf maximal 999 inkrementiert, dann springt sie auf 0 zurück und die Hauptversion wird um 1 erhöht. Das Hochzählen der Hauptversion können Sie auch manuell ausführen, indem Sie auf die Schaltfläche "..." klicken. Beim manuellen Hochzählen erscheint

eine Meldung, die Sie bestätigen müssen. Anschließend wird das Projekt übersetzt und die neue Versionskennung wird eingetragen.

Hinweis

Wenn Sie ein neues Simulationsprojekt anlegen oder ein mit einer älteren SIMIT-Version erstelltes Simulationsprojekt öffnen oder dearchivieren, dann ist die Version auf "unversioned" gesetzt. Eine Versionskennung wird dann automatisch gesetzt, wenn Sie die Simulation starten.

Versionierung einsehen mit dem Komponententyp "ProjectVersion"

Eine weitere Möglichkeit, die Versionskennung anzeigen zu lassen, ist über den Komponententyp "ProjectVersion". Sie finden diesen Komponententyp in der Basisbibliothek im Unterordner "Misc."

Weitere Informationen zu diesem Komponententyp finden Sie im Abschnitt: ProjectVersion – Projektversion (Seite 468).

Versionierung einsehen über eine Systemvariable

Bei Verwendung des Automatic Control Interface kann auf die Projektversion über die Systemvariable *_ProjectVersion* zugegriffen werden. Damit kann die Projektversion auch in der Protokolldatei ausgegeben werden. Erstellen Sie dazu ein Skript mit folgendem Programmcode:

```
OPEN-LOG "C:\\Logfile.txt"
PRINTF "project version: %s", "_ProjectVersion"
CLOSE-LOG
```

3.1.3 Schreibschutz

Das ganze Simulationsprojekt oder auch nur Teile davon können gegen Änderungen geschützt werden. Dazu besitzen alle schützbaren Elemente eines Projekts die zusätzliche Eigenschaft "Schreibgeschützt".

Setzen und Aufheben des Schreibschutzes

Wenn Sie den Schreibschutz auf einen Ordner anwenden oder aufheben möchten, kann diese Aktion auf alle Unterordner und darin befindliche Objekte übertragen werden. Es erscheint dann eine Meldung, die Sie bestätigen müssen.

In Unterordnern kann der Schreibschutz der einzelnen Objekte auch wieder individuell geändert werden, er ist nicht fest an den Status des Ordners gebunden.

Zum Setzen oder Aufheben des Schreibschutzes für ein Objekt gehen Sie folgendermaßen vor:

1. Selektieren Sie das Objekt.
2. Aktivieren oder deaktivieren Sie das Optionskästchen der Eigenschaft "Schreibgeschützt":

Auswirkungen des Schreibschutzes

Der Schreibschutz hat unterschiedliche Auswirkungen, je nachdem, auf welche Objekte eines Projekts er angewendet wird:

- **Diagramm**
  Schreibgeschützte Diagramme können nicht gelöscht oder umbenannt werden. Sie können im Editor geöffnet, aber nicht verändert werden. Objekte des Diagramms (Komponenten, Controls und Grafiken) können selektiert und damit auch auf andere Diagramme kopiert werden. Die Bedienung bei laufender Simulation ist möglich.
- **Diagrammordner**
  Schreibgeschützte Ordner können nicht gelöscht oder umbenannt werden. In schreibgeschützten Ordnern können keine neuen Diagramme oder Unterordner angelegt werden, auch nicht durch Verschieben oder Kopieren von Diagrammen oder Ordnern.
- **Kopplungen**
  Schreibgeschützte Kopplungen können nicht gelöscht oder umbenannt werden. Sie können geöffnet, aber nicht verändert werden.
- **Schnappschüsse**
  Schreibgeschützte Schnappschüsse können zur Laufzeit einer Simulation geladen, aber nicht gelöscht oder umbenannt werden.
- **Schnappschussordner**
  Schreibgeschützte Ordner können nicht gelöscht oder umbenannt werden. In schreibgeschützten Ordnern können keine Unterordner oder Schnappschüsse angelegt werden, auch nicht durch Verschieben oder Kopieren von Schnappschüssen oder Ordnern. Wenn der oberste Schnappschussordner im Projektbaum schreibgeschützt ist, können keine Schnappschüsse mehr erstellt werden, da diese automatisch im obersten Schnappschussordner abgelegt würden. Das entsprechende Symbol in der Symbolleiste ist in diesem Fall inaktiv.
- **Kurvenbild**
  Schreibgeschützte Kurvenbilder können nicht gelöscht oder umbenannt werden. Sie können geöffnet, aber nicht verändert werden.
- **Archiv**
  Das schreibgeschützte Archiv kann geöffnet, aber nicht verändert werden.
- **Skript**
  Schreibgeschützte Skripte können nicht gelöscht oder umbenannt werden. Sie können geöffnet, aber nicht verändert werden.
- **Skriptordner**
  Schreibgeschützte Ordner können nicht gelöscht oder umbenannt werden. In schreibgeschützten Ordnern können keine neuen Skripte oder Unterordner angelegt werden, auch nicht durch Verschieben oder Kopieren von Skripten oder Ordnern.
- **Listen**
  Schreibgeschützte Listen können nicht gelöscht oder umbenannt werden. Sie können geöffnet, aber nicht verändert werden.

- **Projektmanager**
  Wenn der Projektmanager schreibgeschützt ist, können die Projekteigenschaften nicht verändert werden. Alle Aktionen bezüglich der Projektelemente sind aber genauso möglich wie im Projektbaum selbst, wobei deren Schreibschutz berücksichtigt wird.
- **Gesamtprojekt**
  Schreibgeschützte Projekte können nicht umbenannt werden und es können keine Kopplungen angelegt werden.
  Beachten Sie, dass der Schreibschutz auf dem Projekt nicht automatisch bedeutet, dass alle Bestandteile des Projekts gegen Veränderung geschützt sind. Sie können aber beim Setzen des Schreibschutzes auf dem Projekt den Schreibschutz auf alle Objekte des Projekts vererben.

Schreibgeschützte Objekte werden im Arbeitsbereich mit einer weißen Titelzeile geöffnet, um auf diesen besonderen Status hinzuweisen.

3.2 Diagrammeditor

3.2.1 Erstellen und Bearbeiten von Diagrammen

Ein Diagramm wird mit vor- oder selbst definierten Komponenten und Controls im Diagrammeditor erstellt. Um ein neues Diagramm zu erstellen, gehen Sie folgendermaßen vor:

1. Führen Sie in der Projektnavigation auf den Baumeintrag "Neues Diagramm" einen Doppelklick aus.
   Ein neues Diagramm wird direkt unterhalb dieses Baumeintrags angelegt.
2. Geben Sie dem neuen Diagramm einen Namen oder übernehmen Sie die Voreinstellung "Diagramm". Weitere neue Diagramme erhalten als Voreinstellung ebenfalls den Namen "Diagramm" und eine fortlaufende Nummer.
3. Führen Sie einen Doppelklick auf das neue Diagramm aus.
   Im Arbeitsbereich öffnet sich der Diagrammeditor.
4. Ziehen Sie per Drag & Drop die Komponenten und Controls für Ihr Projekt auf den Diagrammeditor.
   - Die entsprechenden Komponenten für verschiedene logische und arithmetische Funktionen, für Antriebe, Sensoren, Verbindungen und Kommunikation finden Sie unter "Basiskomponenten" in der Task-Card "Komponenten".
   - Die entsprechenden Objekte zur Eingabe und zur Anzeige von Werten finden Sie in der Task-Card "Controls".
     Unter "Anzeige" finden Sie Objekte zum dynamischen Anzeigen von Werten aus einer laufenden Simulation.
     Unter "Eingabe" finden Sie Objekte zur Vorgabe von bestimmten Werten in eine laufende Simulation.
5. Erstellen Sie mit <Strg+Shift+Drag & Drop> Verknüpfungssichten von bereits platzierten Komponenten.
   Über die Verknüpfungssicht können Sie bei laufender Simulation das Bedienfenster der Komponente mit Doppelklick öffnen. Weiterführende Informationen finden Sie unter Das Verknüpfungssymbol (Seite 365).

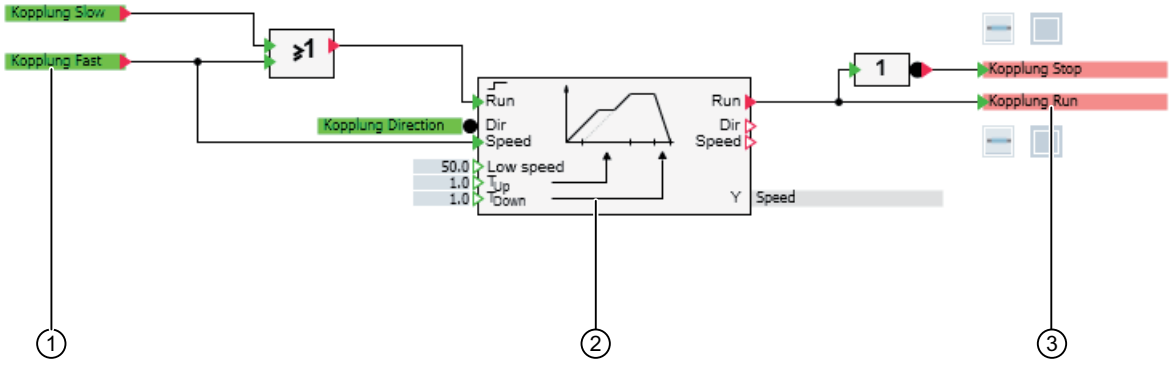
6. Wenn Sie Steuerelemente von Bedienfenstern von Komponenten auf dem Diagramm platzieren wollen:
   - Öffnen Sie mit Doppelklick das Bedienfenster der Komponente.
   - Selektieren Sie die gewünschten Steuerelemente.
   - Ziehen Sie die Steuerelemente mit Drag & Drop auf das Diagramm.

   Ob eine Komponente ein Bedienfenster besitzt, erfahren Sie in der Hilfe zur Komponente.
7. Verbinden Sie die Anschlüsse der Komponenten und Controls untereinander und mit der Steuerung. Die Verbindung mit der Steuerung erfolgt über Konnektoren.

Hinweis

Die Mindestgröße für ein Diagramm ist 20 × 20 Pixel.

Die folgende Abbildung zeigt ein Beispiel für eine Komponente mit Eingangs-und Ausgangskonnektoren im Diagrammeditor:



① Ausgangskonnektor
② Komponente
③ Eingangskonnektor

Verbinden von Anschlüssen

Zum Verbinden von Anschlüssen gehen Sie folgendermaßen vor:

1. Bewegen Sie den Mauszeiger über den zu verbindenden Anschluss.
   Wenn der Mauszeiger seine Darstellung ändert, ist eine Verbindung an dieser Stelle möglich.
2. Klicken Sie auf den zu verbindenden Anschluss.
3. Bewegen Sie den Mauszeiger über den Verbindungspartner.
4. Klicken Sie erneut mit der linken Maustaste.
   Die Verbindung ist jetzt hergestellt und wird durch eine Verbindungslinie dargestellt.

Alternativ können Sie auch einen Anschluss mit gedrückter Maustaste auf einen anderen ziehen und dort loslassen.

Parameter und Werte einem Anschluss zuweisen

Parameter für eine Komponente oder ein Control setzen Sie in den Feldern an den Eingängen oder im Eigenschaftsfenster der Komponente oder des Controls:

- Zur Eingabe am Control/an der Komponente direkt öffnen Sie das Eingabefeld mit einem Doppelklick.
- Zur Eingabe über das Eigenschaftsfenster öffnen Sie das Eigenschaftsfenster, indem Sie auf die Komponente/das Control klicken.

Tragen Sie den gewünschten Wert ein und bestätigen Sie die Eingabe mit <Return>.

Verbinden von Konnektoren

Die Ein- und Ausgangssignale der Steuerung werden in den Kopplungen verwaltet. Auf Diagrammen werden diese Signale durch Konnektoren dargestellt:

- Ausgangssignale durch grüne Ausgangskonnektoren
- Eingangssignale durch rote Eingangskonnektoren

Ein- und Ausgangskonnektoren können Sie per Drag & Drop aus der Kopplung auf die Diagramme ziehen. Gehen Sie hierzu folgendermaßen vor:

1. Teilen Sie den Arbeitsbereich über den Menübefehl "Fenster > Horizontal".
2. Öffnen Sie die Kopplung und das Diagramm.
3. Ziehen Sie das gewünschte Signal aus der Kopplung auf das Diagramm, indem Sie das Signal im Kopplungsfenster am linken Rand anfassen und <Shift> gedrückt halten.
4. Verbinden Sie den Anschluss des Konnektors mit dem Anschluss einer Komponente.

Hinweis

Die Konfiguration der Kopplung muss gespeichert sein, um ein Signal herausziehen zu können.

Alternativ können Sie die Ein- und Ausgangskonnektoren auch aus der Task-Card "Signale" auf ein Diagramm ziehen. Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Signale" (Seite 242).

3.2.2 Grafiken visualisieren

Mit Grafiken visualisieren Sie das Simulationsmodell und fügen erläuternde Texte ein. Sie können statische und animierte Grafiken erstellen und miteinander verknüpfen. Grafiken werden im Diagrammeditor dargestellt und bearbeitet.

Statische Grafik

Zur Verwendung einer statischen Grafik ziehen Sie das gewünschte Grafikelement aus der Task-Card "Grafik" per Drag & Drop in den Diagrammeditor. Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Grafik" (Seite 238).

Bearbeiten Sie das Grafikelement mit den in der Symbolleiste des Diagrammeditors zur Verfügung gestellten Funktionen:



Über diese Symbolleiste können folgende Funktionen aufgerufen werden:

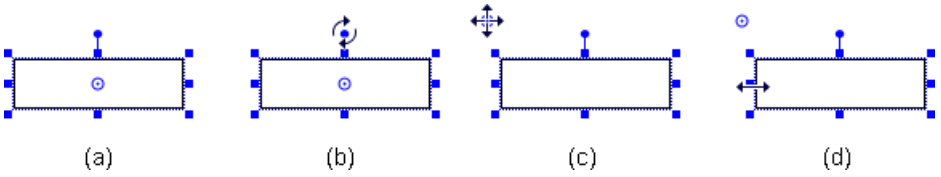
- Text: Zeichensatz, Zeichengröße und Farbe für Text einstellen.
- Linien: Farbe für die Füllung und Linien auswählen, die Stärke und Strichart von Linien verändern.
- Grafikelement: Grafiken spiegeln, mehrere Grafiken zueinander ausrichten, verteilen oder gruppieren. Grafiken gruppieren, einzelne oder gruppierte Grafiken in den Vorder- oder Hintergrund verschieben.

Selektieren Sie mehrere Grafiken mit einem Fangrahmen oder durch Drücken von <Shift> oder <Strg>.

Grafiken werden in ein vorgegebenes Raster eingeordnet. Mit <Alt> heben Sie die Rasterung beim Bearbeiten von Grafik auf.

Die Parameter von Grafiken können auch im Eigenschaftsfenster verändert werden.

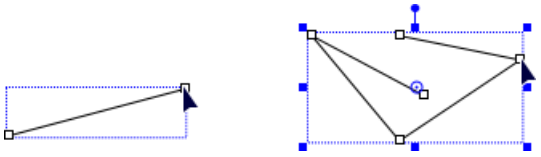
Änderung von Größe und Winkel einer statischen Grafik



Haben Sie ein Grafikelement ausgewählt, wird ein Selektionsrahmen eingeblendet (in der Abbildung oben unter (a)). Über die eingeblendeten blauen Quadrate verändern Sie folgendermaßen Größe und Winkel des Grafikelements:

- Führen Sie den Mauszeiger über eines der Quadrate, bis er seine Darstellung ändert wie in der Abbildung oben unter (d) zu sehen. Mit gedrückter Maustaste ziehen Sie das Grafikelement dann auf die gewünschte Größe.
- Mit dem oberen blauen Punkt an einem Grafikelement können Sie dieses um einen beliebigen Winkel drehen. Führen Sie dazu den Mauszeiger über den blauen Punkt, bis er seine Darstellung wie in der Abbildung oben unter (b) ändert. Mit gedrückter Maustaste drehen Sie das Grafikelement auf den gewünschten Winkel.
- Den Drehpunkt können Sie beliebig verschieben (Abbildung oben unter (c)).

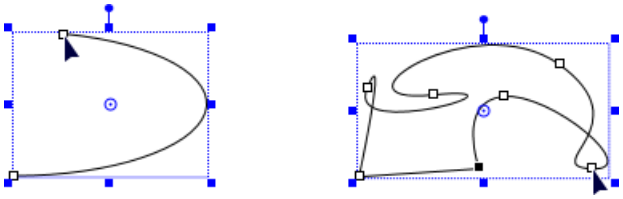
Änderung von Linien



Bei einfachen geraden Linien wie auch bei Linien in einem Polygonzug können Sie die beiden Enden verschieben. Gehen Sie hierzu folgendermaßen vor:

1. Selektieren Sie die Linie.
   An beiden Enden der Linie erscheint ein kleines Rechteck.
2. Führen Sie nun den Mauszeiger über einen Endpunkt, bis er seine Darstellung wie in der Abbildung oben ändert.
3. Mit gedrückter Maustaste verschieben Sie nun den Endpunkt.

Auf die gleiche Art können Sie die Endpunkte bei gekrümmten Linien, also die Endpunkte von Ellipsenbögen oder die Punkte einer Bezierkurve, verschieben:



Animierte Grafik

Für jedes Grafikelement können Animationen erstellt werden. Wenn Sie z. B. ein Rechteck zeichnen, sehen Sie im Eigenschaftsfenster des Rechtecks die Eigenschaft "Animationen".



Mit einem Doppelklick auf "Neue Animation" legen Sie eine Animation für das ausgewählte Grafikelement an.

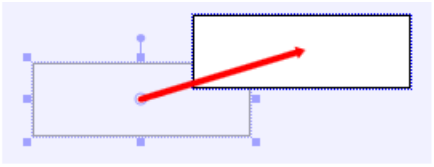
Ein Dialogfeld wird geöffnet, wo sie aus den folgenden Animationen wählen können:

- **Bewegung**
  Das Grafikelement wird auf der Diagrammfläche bewegt.
- **Rotation**
  Das Grafikelement rotiert um die Rotationsachse.
- **Skalierung**
  Das Grafikelement verändert seine Größe.
- **Sichtbarkeit**
  Das Grafikelement wird ein- und ausgeblendet.
- **Bildwechsel**
  In das Grafikelement werden Bilddateien eingeblendet.
- **Bildfolge**
  In das Grafikelement wird eine Bildfolge aus Bilddateien eingeblendet.

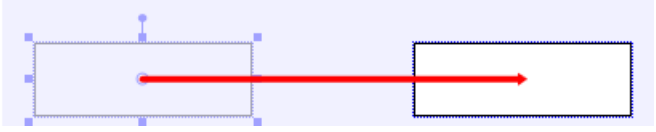
Für ein Grafikelement können Sie mehrere Animationen anlegen und so ein Objekt beispielsweise sowohl bewegen als auch skalieren. Bildwechsel und Bildfolge schließen sich aber gegenseitig aus.

Beispiel für die Animation "Bewegung"

(a)



(b)

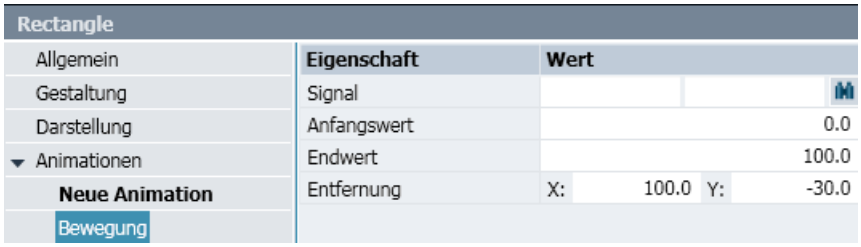


Wenn Sie die Animation "Bewegung" auswählen, erscheint versetzt ein Duplikat des Grafikelements. Ein roter Pfeil verbindet das Duplikat mit dem Ursprungselement wie in der Abbildung oben unter (a). Dieser rote Pfeil veranschaulicht die Bewegung des Objekts.

Verschieben Sie nun das Duplikat mit der Maus an die gewünschte Endposition.
In der Abbildung oben unter (b), ist die Bewegung und Endposition für eine horizontale Bewegung zu sehen.

Im Eigenschaftsfenster (s. Abbildung unten) tragen Sie nun das Signal ein, das die Werte für die Bewegung liefert.

- Ziehen Sie dazu das Signal per Drag & Drop aus der Task-Card "Signale" in das Eigenschaftsfeld.
- Geben Sie einen Anfangs- und Endwert ein.



Wenn Sie nun die Simulation starten und den Signalwert verändern, wird das Grafikelement horizontal bewegt.
Für Signalwerte innerhalb des mit Anfangs- und Endwert definierten Bereichs erfolgt eine Bewegung längs des vorgegebenen Pfeils, für Signalwerte außerhalb des definierten Bereichs wird die Bewegung geradlinig außerhalb des Pfeils fortgesetzt. Der Pfeil gibt damit die Richtung der geradlinigen Bewegung vor, Anfangs- und Endwert definieren lediglich den Maßstab des Grafikelements auf der Diagrammoberfläche.

Mehrere Bewegungsanimationen für ein Grafikobjekt werden überlagert und ermöglichen so auch Bewegungen längs gekrümmter Bahnen.

Bei Gruppierungen von mehreren Grafikobjekten können Sie sowohl die Gruppe insgesamt als auch jedes einzelne Grafikobjekt der Gruppe animieren.

3.2.3 Signale visualisieren

Signale werden über Controls auf den Diagrammen angezeigt und gesetzt. Die Controls können beliebig auf einem Diagramm angeordnet werden und auf verschiedenen Diagrammen in beliebigen Gruppierungen zusammengefasst werden. Je Diagramm bestimmen Sie damit nicht nur die dargestellten Signale, sondern auch deren Anordnung.

Um ein Signal im Diagrammeditor darzustellen, ziehen Sie das gewünschte Signal aus der Task-Card "Signale" per Drag & Drop in den Diagrammeditor. Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Signale" (Seite 242).
Alternativ können Sie das Signal auch aus dem Kopplungseditor ziehen.

Die Darstellung des Signals ist abhängig vom Datentyp:

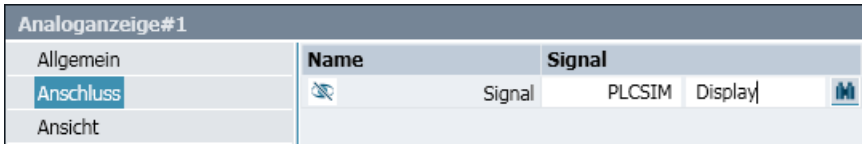
Darstellung von binären Signalen	Darstellung von analogen Signalen

Der aktuelle Signalname und ein Signaltrenner werden mit angezeigt.

Ein Signal kann auch mit mehreren Controls gleichzeitig angezeigt werden, z. B. mit einer Digitalanzeige und einer Analoganzeige. Die Digitalanzeige erhalten Sie, wenn Sie das Signal aus der Kopplung oder der Task-Card ziehen und den Signaltrenner löschen.
Die Digitalanzeige und die Analoganzeige erzeugen Sie, indem Sie das entsprechende Control aus der Task-Card "Controls" auf das Diagramm ziehen.

Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Controls" (Seite 234).

Den Signalnamen tragen Sie entweder selbst ein oder er wird durch Ziehen des Signals per Drag & Drop aus der Task-Card "Signale" in das Eigenschaftsfenster automatisch erzeugt.



3.2.4 Drucken von Diagrammen

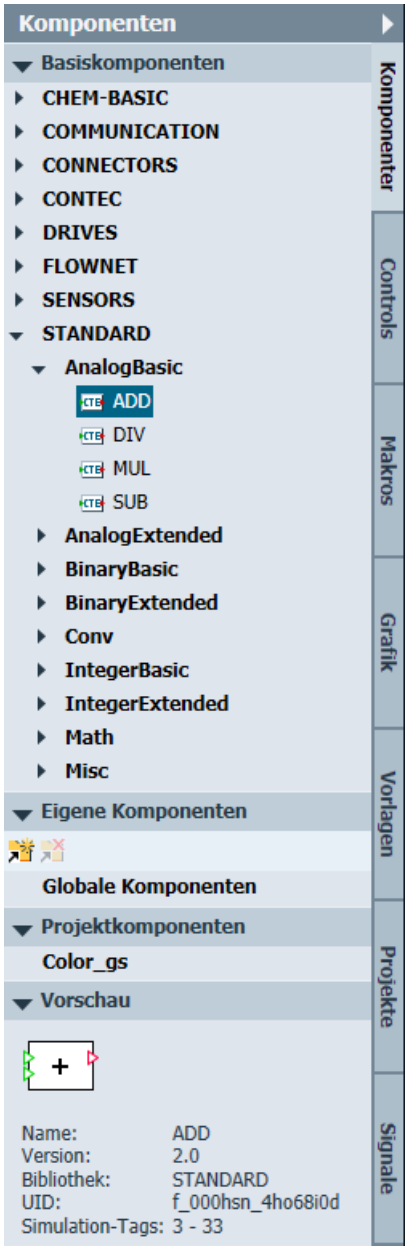
Diagramme und Vorlagen können ausgedruckt werden. Dafür gibt es folgende Eigenschaften:

- Ist in der Projektnavigation ein Ordner markiert, werden alle Diagramme in diesem Ordner und in allen Unterordnern als einzelne Druckaufträge generiert.
- Passt ein Diagramm nicht auf das ausgewählte Papierformat, wird es automatisch skaliert.
- Die Größe des Diagramms wird im Ausdruck durch einen schwarzen Rahmen markiert.
- Als zusätzliche Information wird mit ausgedruckt:
  - bei Diagrammen der Planname mit zugehöriger Ordnerhierarchie
  - bei Vorlagen der absolute Dateiname mit Pfad

3.3 Task-Cards

3.3.1 Task-Card "Komponenten"

Beim Starten von SIMIT werden die Basisbibliothek, die globale Bibliothek und die Projektbibliothek eingelesen und in der Task-Card "Komponenten" angezeigt. Wenn beim vorhergehenden Schließen von SIMIT Bibliotheksverzeichnisse im Bereich "Eigene Komponenten" geöffnet waren, dann werden diese wieder geöffnet.



Die Task-Card besteht aus den folgenden Bereichen:

Basiskomponenten

Hier finden Sie die Komponententypen der Basisbibliothek. Die Basisbibliothek wird bei der Installation von SIMIT angelegt. Die Komponententypen der Basisbibliothek können nicht verändert und es können auch keine weiteren Komponententypen hinzugefügt werden. Die Komponententypen der Basisbibliothek können aber als Kopie in den Bereichen "Eigene Komponenten" und "Projektkomponenten" abgelegt werden.

Weitere Informationen hierzu finden Sie im Kapitel: Basisbibliothek (Seite 391).

Hinweis

Im Projekt verwendete Komponenten einer Bibliothek hinzufügen

In folgende Bereiche können Sie im Projekt verwendete Komponenten per Drag & Drop einfügen:

- Eigene Komponenten
- Projektkomponenten

Eigene Komponenten

Hier finden Sie die Komponententypen, die Sie selbst erstellt haben oder die Ihnen von Dritten zur Verfügung gestellt worden sind. Wenn Sie Komponententypen im Bereich "Globale Komponenten" ablegen, werden diese in jedem SIMIT-Projekt zur Verfügung stehen.

Mit dem Befehl öffnen Sie beliebige Bibliotheksverzeichnisse und haben dann Zugriff auf die darin abgelegten Komponententypen. Mit dem Befehl entfernen Sie ein selektiertes Verzeichnis wieder aus diesem Bereich.

Wenn Sie eigene Komponententypen mit dem CTE erstellen, müssen Sie diese entweder in einem Bibliotheksverzeichnis oder unter *Globale Komponenten* abspeichern, damit sie Ihnen in SIMIT zur Verfügung stehen. SIMIT aktualisiert den Bereich "Eigene Komponenten" automatisch, wenn Sie dort einen Komponententyp mit dem Komponententypeditor speichern.

Weitere Informationen zum Erstellen eigener Komponenten finden Sie in der Hilfe "SIMIT – Component Type Editor".

Projektkomponenten

Hier finden Sie die Komponententypen, die Sie mit dem geöffneten SIMIT-Projekt verwenden können. Wenn Sie das Projekt archivieren, werden alle Komponententypen in diesem Bereich als Projektbibliothek mit dem Projekt archiviert. Wenn das Projekt wieder dearchiviert wird, stehen diese Komponententypen dort auch wieder zur Verfügung.

Sie können Komponententypen innerhalb der beiden Bereiche "Eigene Komponenten" und "Projektkomponenten" beliebig verschieben oder als Kopie einfügen. Die Komponententypen des Bereichs "Basiskomponenten" können Sie nur als Kopie in die beiden anderen Bereiche übernehmen.

Öffnen Sie Komponententypen über den Kontextmenübefehl "Öffnen" im Editor.

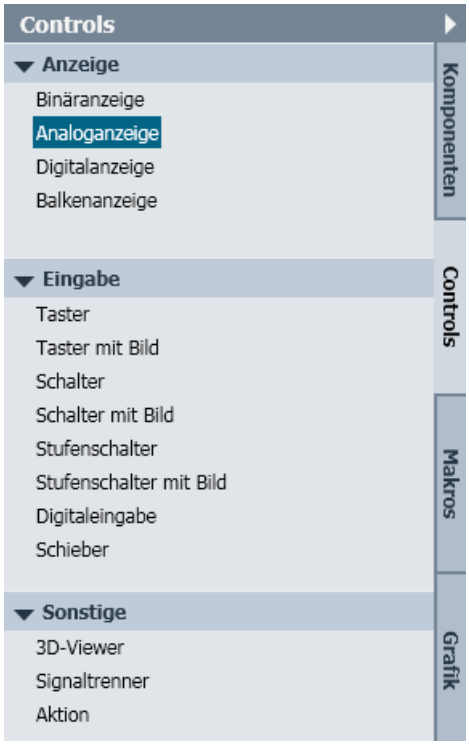
Vorschau

In der Vorschau der Task-Card "Komponenten" werden zu einem selektierten Komponententyp die folgenden Informationen angezeigt:

- **Symbol**
  Das Grundsymbol des Komponententyps.
- **Name**
  Der im Komponententyp eingetragene Name.
- **Version**
  Die im Komponententyp eingetragene Information zur Version.
- **Bibliothek**
  Die im Komponententyp eingetragene Information zur Bibliothek.
- **UID**
  Die dem Komponententyp automatisch gegebene eindeutige Kennung.
- **Simulation-Tags**
  Die im Komponententyp eingetragene Information zur Anzahl der Simulation-Tags.

3.3.2 Task-Card "Controls"

Mit Controls werden Signalwerte angezeigt und dargestellt.



Die Task-Card besteht aus den folgenden Bereichen:

Anzeige

Hier finden Sie folgende Controls zur Anzeige von Signalwerten.

- Binäranzeige (Seite 577)
- Analoganzeige (Seite 578)
- Digitalanzeige (Seite 580)
- Balkenanzeige (Seite 582)

Eingabe

Hier finden Sie folgende Controls zur Eingabe von Signalwerten:

- Taster (Seite 583)
- Taster mit Bild (Seite 584)
- Schalter (Seite 585)
- Schalter mit Bild (Seite 586)
- Stufenschalter (Seite 587)
- Stufenschalter mit Bild (Seite 588)
- Digitaleingabe (Seite 590)
- Schieber (Seite 592)

Bei der Option "mit Bild" kann das Aussehen des Controls auf dem Diagramm durch Bilder ergänzt werden. So erhalten Sie z. B. mit Fotografien eine realitätsnahe Darstellung der Controls.

Sonstige

Hier finden Sie folgende weitere Controls:

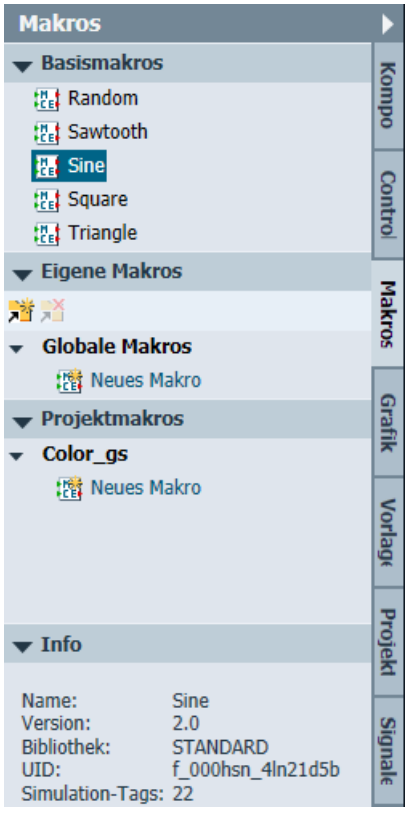
- 3D Viewer-Control (Seite 596)
- Signaltrenner (Seite 593)
- Aktion (Seite 595)

Die Controls können mit allen Signalen verknüpft werden, die sich in der Task-Card "Signale" befinden. Sie können nicht nur Ein- und Ausgangssignale von Kopplungen und Komponenten, sondern auch Zustandswerte von Komponenten anzeigen lassen und bei laufender Simulation änderbare Parameter über Controls vorgeben.

Weitere Informationen hierzu finden Sie im Abschnitt: Signale visualisieren (Seite 231).

3.3.3 Task-Card "Makros"

In einer Makrokomponente können Sie immer wiederkehrende Funktionen eines Simulationsmodells zusammenfassen. Mit dem Makro haben Sie dann einen einfachen Zugriff auf die Funktion: Sie müssen jetzt nicht mehr diese Funktion aus einem Diagramm kopieren und in anderen Diagrammen einfügen, sondern können die Makrokomponente wie Komponententypen auch aus der Task-Card ziehen, parametrieren und mit anderen Komponenten oder Makrokomponenten verknüpfen.



Die Task-Card besteht aus den folgenden Bereichen:

Basismakros

Hier finden Sie 5 Makros, die als Signalgeneratoren einsetzbar sind. Sie sind bezüglich der Periodendauer und Amplitude einstellbar und liefern unterschiedliche Signalformen:

- Zufallszahlen (Random)
- Sägezahn (Sawtooth)
- Sinus (Sine)
- Rechteck (Square)
- Dreieck (Triangle)

Hinweis

Im Projekt verwendete Makros einer Bibliothek hinzufügen

In folgende Bereiche können Sie im Projekt verwendete Makros per Drag & Drop einfügen:

- Eigene Makros
- Projektmakros

Eigene Makros

Hier können Sie eigene Makrokomponenten anlegen. Diese Makrokomponenten werden im Arbeitsbereich von SIMIT abgelegt, sie stehen also für alle Projekte zur Verfügung. Über das Symbol "" öffnen Sie Ordner, die Makrokomponenten enthalten. Mit dem Symbol "" schließen Sie einen selektierten Ordner und entfernen ihn damit wieder aus diesem Bereich.

Projektmakros

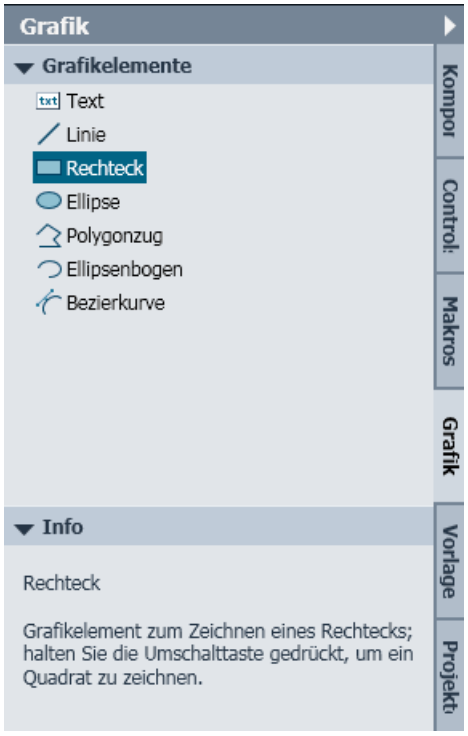
Hier können Sie eigene Makrokomponenten anlegen. Diese Makrokomponenten werden im Projektordner abgelegt. Sie sind nur verfügbar, solange dieses Projekt geöffnet ist. Alle hier abgelegten Makrokomponenten werden mit dem Projekt archiviert. Sie stehen Ihnen damit wieder im Projekt zur Verfügung, wenn Sie es dearchivieren.

Info

Hier werden zu einer selektierten Makrokomponente der Name, die Version, die Bibliothek, die UID und die Anzahl der Simulation-Tags angezeigt.

3.3.4 Task-Card "Grafik"

In der Task-Card "Grafik" finden Sie Grafikelemente zur Verwendung in Diagrammen.



Grafikelemente

Die unterschiedlichen Grafikelemente sind aufgelistet.

Ziehen Sie das gewünschte Grafikelement per Drag & Drop in den Diagrammeditor. Dort wird es angezeigt und bearbeitet. Zur Bearbeitung stehen Ihnen die Symbole aus dem Diagrammeditor zur Verfügung.

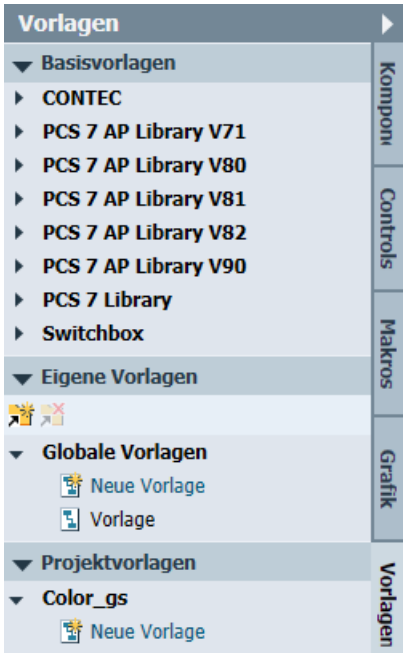
Weitere Informationen finden Sie im Abschnitt: Visualisieren einer Simulation (Seite 31).

Info

Hier finden Sie eine Kurzbeschreibung des aktuell markierten Grafikelements.

3.3.5 Task-Card "Vorlagen"

Die Task-Card "Vorlagen" ist bei offenem Projektmanager verfügbar.



Die Task-Card besteht aus den folgenden Bereichen:

Basisvorlagen

Hier sind die Vorlagen abgelegt, die Ihnen von SIMIT zur Verfügung gestellt werden. Im Verzeichnis "PCS 7 Library" finden Sie die zu den Messstellentypen (Templates) der PCS 7 Library passenden Vorlagen. Vorlagen, die an die Bibliothek APL (Advanced Process Library) von PCS 7 Vx.y angepasst sind, finden Sie im Vorlagenordner "PCS 7 AP Library Vxy". Basisvorlagen können Sie per Drag & Drop in die Bereiche "Eigene Vorlagen" oder "Projektvorlagen" kopieren und dort editieren.

Eigene Vorlagen

Legen Sie hier die Vorlagen für Ihre SIMIT-Installation ab. Diese Vorlagen werden dann im Bereich "Globale Vorlagen" angezeigt. Eigene Bibliotheken können Sie mit dem Symbol "" öffnen und mit dem Symbol "" wieder schließen.

Projektvorlagen

Die hier abgelegten Vorlagen werden im Projekt gespeichert und mit dem Projekt archiviert. Nach dem Öffnen oder Dearchivieren eines Projekts stehen Ihnen diese Vorlagen wieder zur Verfügung.

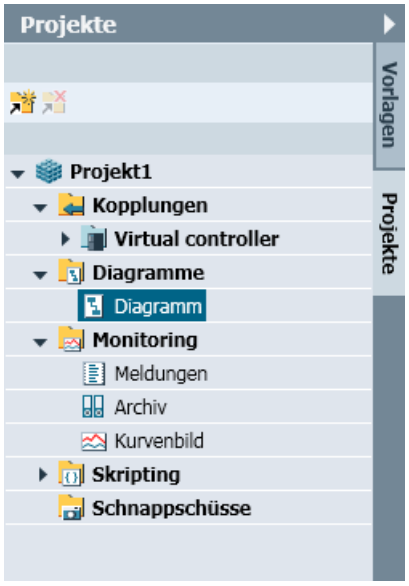
Vorlagen in den Bereichen "Eigene Vorlagen" und "Projektvorlagen" können Sie per Drag & Drop beliebig innerhalb dieser beiden Bereiche verschieben oder kopieren.

Weitere Informationen finden Sie im Kapitel: Vorlagen (Seite 252).

3.3.6 Task-Card "Projekte"

In der Task-Card "Projekte" sind mehrere Projekte gleichzeitig geöffnet. Damit kann z. B. ein Projekt aus mehreren "Standardprojekten" zusammengestellt werden.

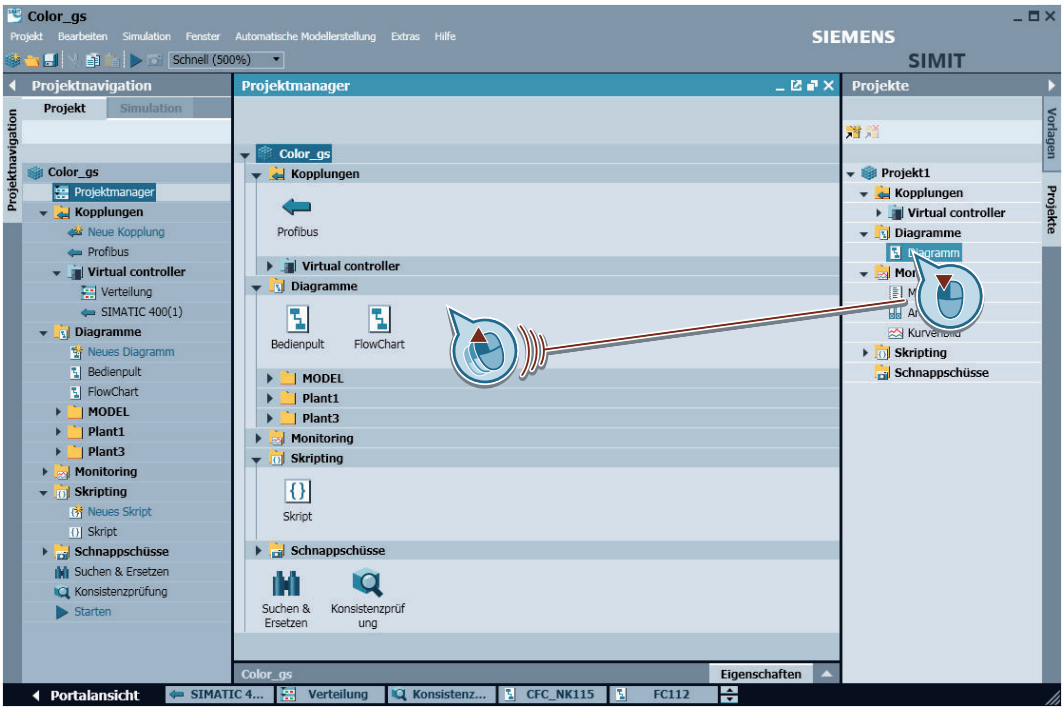
Die Task-Card "Projekte" ist bei geöffnetem Projektmanager verfügbar. Der Projektmanager steht Ihnen auch bei einer laufenden Simulation zur Verfügung. Projektelemente können dann ebenfalls aus dem Projektmanager heraus geöffnet werden.



Beispiel:

1. Dearchivieren Sie 2 Projekte in jeweils unterschiedliche Projektordner.
2. Öffnen Sie ein weiteres Projekt in SIMIT.
3. Öffnen Sie den Projektmanager.
4. Klicken Sie in der Task-Card "Projekte" auf das Symbol "".
   Das Dialogfeld "Projekt öffnen" öffnet sich.
5. Navigieren Sie zum Projektordner des geöffneten Projekts und wählen Sie dort die Projektdatei *.simit aus.
   Das Projekt wird Ihnen jetzt in der Task-Card "Projekte" im Lesemodus geöffnet.

Sie können nun einzelne Elemente, wie Diagramme oder ganze Ordner, per Drag & Drop als Kopie in den Projektmanager und damit in das bearbeitete Projekt ziehen, wie in der folgenden Abbildung dargestellt:

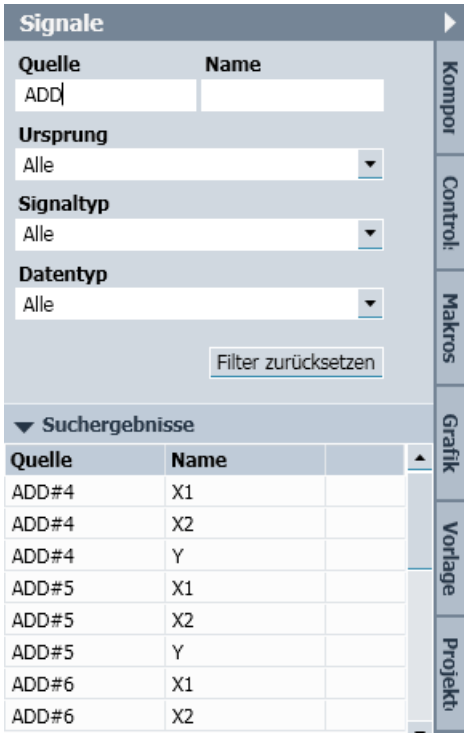


Sie können auch einzelne Diagramme aus einem Projekt im Diagrammeditor öffnen und Teile dieser Diagramme kopieren und in Diagramme eines anderen Projekts einfügen.

Schließen Sie Projekte in der Task-Card, indem Sie auf das Symbol "" klicken.

3.3.7 Task-Card "Signale"

In der Task-Card "Signale" können Sie nach den Signalen im geöffneten Projekt suchen und sie anzeigen lassen.



Folgende Suchkriterien können gewählt werden:

Quelle
Geben Sie hier einen beliebigen Text als Suchkriterium ein.

Name
Geben Sie hier einen beliebigen Text als Suchkriterium ein.

Ursprung
Wählen Sie ein Suchkriterium aus der Klappliste aus.

Signaltyp
Wählen Sie ein Suchkriterium aus der Klappliste aus.

Datentyp
Wählen Sie ein Suchkriterium aus der Klappliste aus.

Mit der Schaltfläche "Filter zurücksetzen" werden die Einstellungen entfernt und es werden alle Signale des geöffneten Projekts unter "Suchergebnisse" angezeigt.

Es müssen nicht alle Suchkriterien befüllt sein. Zur Anzeige von Suchergebnissen genügt bereits die Eingabe eines einzigen Suchkriteriums.

Info



Wenn Sie ein Signal markiert haben, wird im Bereich "Info" der Ursprung, der Signaltyp und der Datentyp angezeigt.

Weitere Informationen finden Sie in folgenden Abschnitten:

- Peripheriesignale zur Verschaltung auf Diagrammen (Seite 207)
- Signal symbolisch adressieren (Seite 214)

3.4 Makrokomponenteneditor

3.4.1 Makroeditor

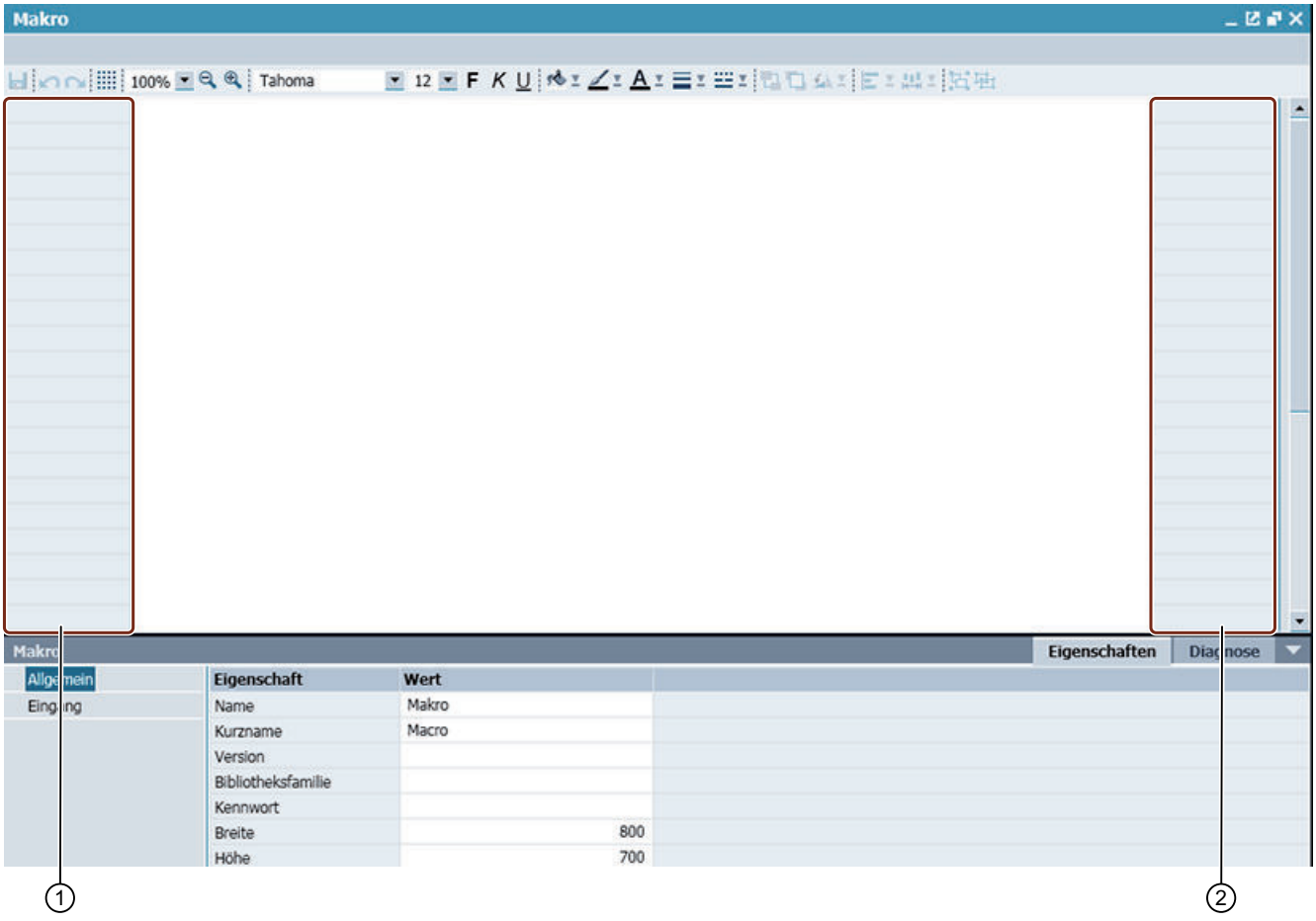
In einer Makrokomponente können Sie immer wiederkehrende Funktionen eines Simulationsmodells zusammenfassen.
Weitere Informationen hierzu finden Sie im Abschnitt: Task-Card "Makros" (Seite 236).

Um ein neues Makro zu erstellen, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Task-Card "Makros".
2. Führen Sie einen Doppelklick auf das Symbol "

" Im Bereich "Eigene Makros" aus.
   Eine neue Makrokomponente wird in diesem Bereich angelegt.
3. Vergeben Sie einen Namen für die neue Makrokomponente oder übernehmen Sie die Voreinstellung.
   Der Makroeditor wird automatisch geöffnet, wenn Sie den vorgegebenen oder geänderten Namen mit der Enter-Taste übernehmen.

Ein Makrodiagramm unterscheidet sich von Diagrammen nur durch die beiden Randleisten links und rechts:



① Randleiste für Eingänge

② Randleiste für Ausgänge

Den Zellen der Randleisten werden die Ein- und Ausgänge des Makros zugeordnet:

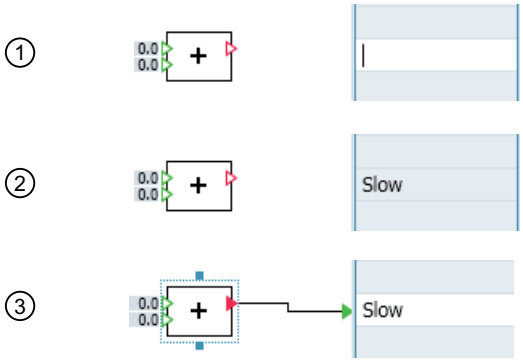
- Eingänge in der linken Randleiste
- Ausgänge in der rechten Randleiste

Wie bei Diagrammen können Sie alle Komponenten der Bibliothek und statische Grafiken verwenden, um ein bestimmtes Verhalten in der Makrokomponente nachzubilden. Von den Konnektorkomponenten können Sie nur den globalen und den topologischen Konnektor verwenden.
Weitere Informationen hierzu finden Sie im Abschnitt: Topologische Anschlüsse von Makrokomponenten (Seite 247).
Controls und Animationen stehen in Makrokomponenten nicht zur Verfügung.
Makrokomponenten können nicht hierarchisch aufgebaut werden, d. h. Sie können in ein Makrodiagramm keine Makrokomponenten einfügen.

Falls Sie bereits ein Diagramm erstellt haben und von diesem Diagramm eine Makrokomponente bilden wollen, kopieren Sie die entsprechenden Teile des Diagramms und fügen Sie diese in ein Makrodiagramm ein. Es werden nur die verwendbaren Komponenten in das Makrodiagramm übernommen.

Um Anschlüsse in die Makrokomponente einzufügen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in eine Zelle der Randleiste.
   Die Zelle wird zur Eingabe eines Namens für den Anschluss geöffnet, wie in der Abbildung unten unter ① gezeigt.
2. Geben Sie einen Namen ein.
3. Bestätigen Sie die Eingabe mit der Return-Taste.
4. Verbinden Sie den Makroanschluss mit dem Anschluss an einer Komponente, indem Sie z. B. den Komponentenanschluss auf den Makroanschluss ziehen, wie in der Abbildung unten unter ② gezeigt.
   Die Verbindung wird dann durch eine Verbindungslinie angezeigt (③).



5. Verfahren Sie so mit allen Anschlüssen.
6. Speichern und schließen Sie das Makrodiagramm.

Alternativ können Sie einen Anschluss der Makrokomponente erzeugen, indem Sie die Verbindung am Anschluss der Komponente durch einen Klick mit der Maus aufnehmen, diese auf die gewünschte Zelle der Randleiste ziehen und die Verbindung dort mit einem Mausklick abschließen. In die Randleiste wird der Name des Komponentenanschlusses als Voreinstellung übernommen.

Hinweis

Jeder Anschluss muss einen eindeutigen Namen besitzen. Der Name darf nur Buchstaben, Ziffern und den Unterstrich enthalten und muss mit einem Buchstaben beginnen. Es wird zwischen Groß- und Kleinschreibung unterschieden.

Makrokomponenten, die nicht fehlerfrei sind, werden durch das Symbol "" gekennzeichnet.

Fehlerhafte Makrokomponenten können im Editor geöffnet und bearbeitet, aber nicht auf Diagrammen verwendet werden. Im Eigenschaftsfenster des Editors finden Sie in der Registerkarte "Diagnose" eine Beschreibung der vorhandenen Fehler.

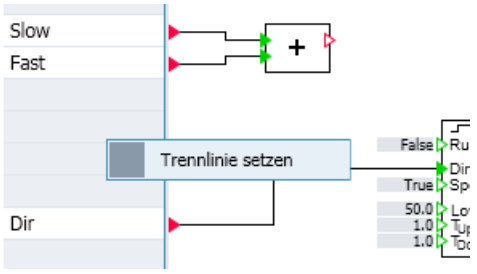


3.4.2 Trennlinien zwischen Anschlüssen von Makrokomponenten einfügen

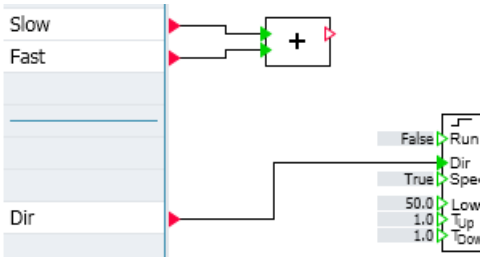
Anschlüsse von Makrokomponenten werden aufeinanderfolgend in den Randleisten gesetzt. Um die Anschlüsse optisch voneinander abzugrenzen, fügen Sie Trennlinien ein.

Gehen Sie hierzu folgendermaßen vor:

1. Öffnen Sie eine Makrokomponente im Makroeditor.
2. Wählen Sie in einer der beiden Randleisten den Kontextmenübefehl "Trennlinie setzen" zwischen zwei Anschlüssen:

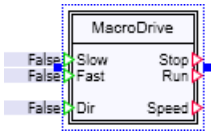


Die Trennlinie wird jetzt in der Zelle als Linie dargestellt:



Darstellung im Symbol der Makrokomponente

Auch im Symbol der Makrokomponente sind jetzt die Anschlüsse voneinander abgesetzt. Optisch wird dies mit einem blauen Quadrat angezeigt. In der folgenden Abbildung sehen Sie, dass auch die Ausgänge durch eine Trennlinie abgesetzt worden sind:

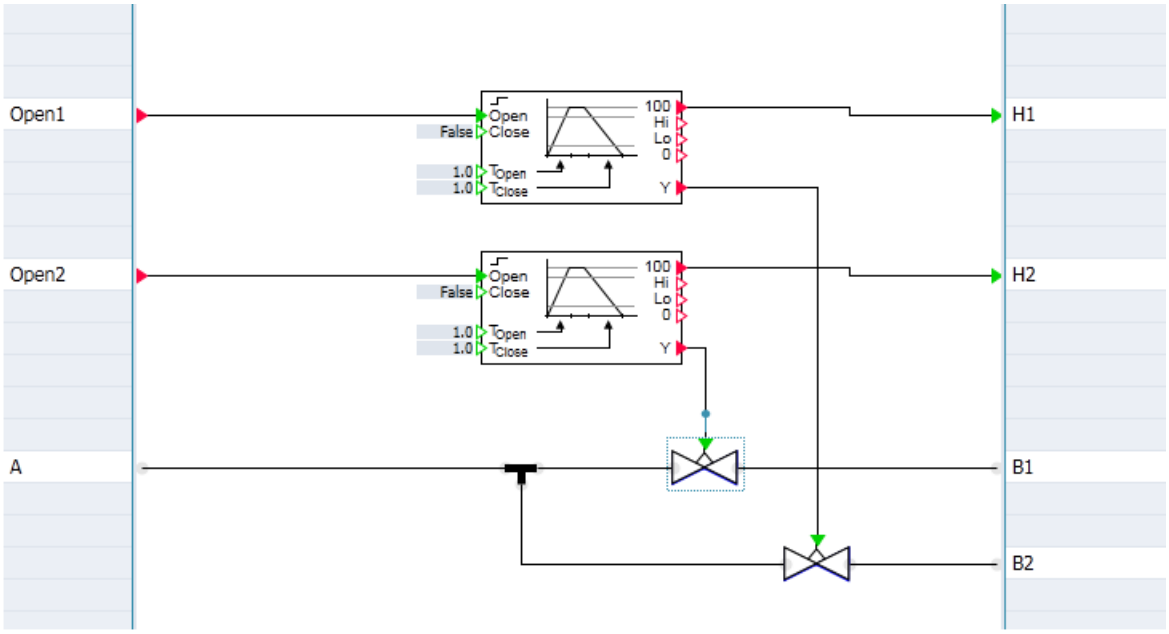


Löschen von Trennlinien

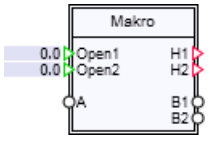
Löschen Sie Trennlinien mit dem entsprechenden Kontextmenübefehl der Zelle.

3.4.3 Topologische Anschlüsse von Makrokomponenten

Komponenten der beiden Bibliotheken FLOWNET und CONTEC können in Makrokomponenten verwendet werden. Offene topologische Anschlüsse aus diesen Komponenten werden auf die topologischen Anschlüsse in den Randleisten der Makrokomponente gelegt.



Wenn solche Makrokomponenten auf Diagrammen verwendet werden, können die topologischen Anschlüsse der Makrokomponente mit topologischen Anschlüssen anderer Komponenten oder anderer Makrokomponenten verbunden werden. Die in den Makrokomponenten definierten topologischen Strukturen sind dann nur Teilstrukturen der gesamten topologischen Struktur eines Fluss- oder Transportnetzes.



3.4.4 Eingänge von Makrokomponenten vorbelegen

Eingänge von Makrokomponenten können individuell vorbelegt werden. Gehen Sie hierzu folgendermaßen vor:

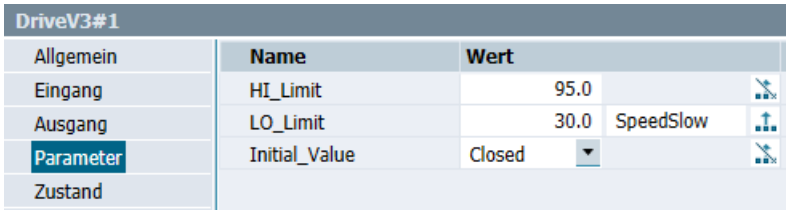
1. Wählen Sie im Eigenschaftsfenster des Makrokomponenteneditors den Eintrag "Eingang" aus.
2. Tragen Sie in der Spalte "Wert/Signal" die gewünschten Werte ein.



3.4.5 Parameter von Makrokomponenten definieren

Für Makrokomponenten können Sie Parameter festlegen. Dafür müssen diese Parameter in der Makrokomponente bereits vorhanden sein, indem sie in den dort verwendeten Komponenten enthalten sind.

Im Eigenschaftsfenster des Makrokomponenteneditors werden Parameter von Komponenten mit dem Symbol " " angezeigt.



Bearbeiten von Parametern

- Klicken Sie auf das Symbol " ", um diesen Parameter als einen Parameter der Makrokomponente festzulegen. Der Parameter wird damit von der Komponentenebene auf die Makrokomponentenebene gehoben.
  Das Symbol wechselt die Darstellung auf " ".
- Tragen Sie in das Textfeld links vom Symbol " " einen Namen für den Parameter ein (Beispiel im Screenshot oben: "SpeedSlow").
  Unter diesem Namen wird der Parameter in der Makrokomponente angezeigt.
- Tragen Sie in das Textfeld unter der Spalte "Wert" eine geeignete Voreinstellung ein (Beispiel im Screenshot oben: "30.0").

Wenn die Makrokomponente auf einem Diagramm verwendet wird, sehen Sie im Eigenschaftsfenster die Parameter mit ihren Voreinstellungen.

3.4.6 Eigenschaften von Makrokomponenten

Folgende Eigenschaften finden Sie im Eigenschaftsfenster des Makrokomponenteneditors:

- **Name**
  Der Name einer Makrokomponente entspricht dem Dateinamen, unter dem eine Makrokomponente im Dateisystem (Arbeitsbereich oder beliebiges Verzeichnis im Dateisystem) abgespeichert wird. Mit diesem Namen wird die Makrokomponente in der Task-Card "Makros" angezeigt.

  Hinweis

  Wenn Makrokomponenten aus einer älteren SIMIT-Version übernommen wurden, kann der Dateiname zunächst auch vom Makronamen abweichen.

- **Kurzname**
  Der Kurzname wird auf der Instanz einer Makrokomponente im Kopfbereich dargestellt.
- **Version**
  Die Version einer Makrokomponente kann frei vergeben werden. Sie wird im Bereich "Info" in der Task-Card "Makros" und im Tooltip der Makrokomponente auf dem Diagramm angezeigt.
- **Bibliotheksfamilie**
  Die Bibliotheksfamilie einer Makrokomponente kann frei vergeben werden. Sie wird im Bereich "Info" in der Task-Card "Makros" und im Tooltip der Makrokomponente auf dem Diagramm angezeigt.
- **Kennwort**
  Makrokomponenten können mit einem Kennwort geschützt werden. Ist eine Makrokomponente mit einem Kennwort geschützt, dann kann sie zwar auf Diagrammen verwendet werden, sie kann aber ohne Eingabe des korrekten Kennworts nicht geöffnet werden. Das zuletzt eingegebene Passwort zum Öffnen kennwortgeschützter Makros bleibt gespeichert, bis das Projekt wieder geschlossen wird.

  Hinweis

  Bewahren Sie die Kennworte Ihrer Makrokomponenten sicher auf, da Sie sonst Ihre mit einem Kennwort geschützte Makrokomponente nicht mehr öffnen können.

- **Breite**
  Die Breite des Symbols.
- **Höhe**
  Die Höhe des Symbols.

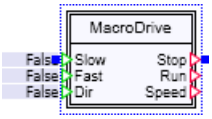
3.4.7 Suchen und Ersetzen in Makrokomponenten

Wählen Sie im Kontextmenü der Makrokomponente den Befehl "Suchen & Ersetzen" für den Austausch von Komponenten innerhalb einer Makrokomponente.

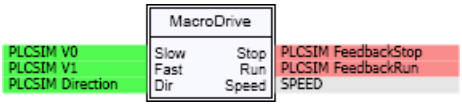
Damit wird der Editor "Suchen & Ersetzen" mit voreingestelltem Fokus auf die Makrokomponente geöffnet.

3.4.8 Makrokomponenten verwenden

Eine Makrokomponente können Sie wie jede Komponente per "Drag & Drop" in ein Diagramm einfügen. Sie wird auf dem Diagramm durch ein Rechteck mit Doppelrahmen als Symbol dargestellt. Im Kopf des Symbols wird der Name der Makrokomponente angezeigt. Die Ein- und Ausgänge sind in der im Makrodiagramm festgelegten Reihenfolge links und rechts an der Komponente angeordnet.



Auf einem Diagramm kann eine Makrokomponente wie jede andere Komponente verwendet werden. Sie kann mit Komponenten und Controls oder auch mit anderen Makrokomponenten verbunden werden.



Durch Doppelklick auf die Makrokomponente im Diagramm öffnen Sie das Makrodiagramm im Lesemodus, d. h. Sie können keine Änderungen am Makrodiagramm vornehmen.

3.5 Migration von Projekten aus älteren SIMIT-Versionen

Projekte aus SIMIT V7.x, SIMIT V8.x und SIMIT V9.x können unverändert in SIMIT V10.0 geöffnet bzw. dearchiviert werden.

Hinweis

Das Dearchivieren oder Öffnen eines SIMIT V7-, SIMIT V8- oder SIMIT V9-Projekts in SIMIT V10.0 kann beim ersten Mal sehr lange dauern.

Projekte aus älteren SIMIT-Versionen können mit SIMIT V10.0 nicht verwendet werden. Diese Projekte müssen zuerst mit einer SIMIT V8-Version geöffnet und migriert werden.

Automatische Modellerstellung 4

In SIMIT stehen verschiedene Automatismen zur Verfügung, um Diagramme zu erzeugen und zu editieren. Die Automatismen basieren auf Dateien, die ein von SIMIT lesbares Format besitzen. Diese Dateien enthalten die notwendigen Informationen zum Erzeugen oder Verändern von Diagrammen. Diese Informationen werden mit dem Import einer Datei automatisch in Diagramme umgesetzt.

Folgende Automatismen stehen zur Verfügung:

- Vorlagen
  Vorlagen sind Muster von Diagrammen, in denen Platzhalter für Parameter, Vorbelegungen etc. definiert sind. Mit dem Import einer Tabelle, in der Werte für die Platzhalter gesetzt sind, werden automatisch die Diagramme aus den Vorlagen erzeugt. Aus PCS 7 exportierte IEA-Dateien, CMT-Dateien sowie Excel-Tabellen beziehen sich auf Vorlagen.
  Weitere Informationen hierzu finden Sie im Abschnitt: Vorlagen (Seite 252).
- Tabellenimport
  Import von Dateien in Tabellenformaten. Die Tabellen enthalten die Ersetzungen für die Platzhalter.
  Weitere Informationen hierzu finden Sie im Abschnitt: Tabellenimport (Seite 261).
- IEA-Import
  Import einer IEA-Datei aus PCS 7. Für jeden CFC-Plan, der auf einer Musterlösung oder einem Messstellentyp basiert, wird ein passendes Diagramm erzeugt.
  Weitere Informationen hierzu finden Sie im Abschnitt: IEA-Import (Seite 264).
- CMT-Import
  Import von Einzelsteuereinheitstypen aus PCS 7.
  Weitere Informationen hierzu finden Sie im Abschnitt: CMT-Import (Seite 266).
- Automatisierter Import
  Import von Diagrammen aus einem ZIP-Archiv oder einer XML-Datei.
  Weitere Informationen hierzu finden Sie im Abschnitt: Automatisierter Import (Seite 270).
- Generierung der Geräteebene
  Dieser Automatismus benutzt keine externe Datei, sondern Informationen, die in den Parametern der im Simulationsmodell verwendeten Komponenten enthalten sind. So können für bestimmte Komponententypen automatisch Vorlagen instanziiert werden, z. B. um zu prozesstechnischen Geräten die passende Simulation der Ansteuerungslogik zu erstellen.
  Weitere Informationen hierzu finden Sie im Abschnitt: Generierung der Simulation von Antrieben und Sensoren (Seite 833).
- Massenbearbeitung von Parametern
  Export und Import von Parametern und Eingangsvorbelegungen aus und in bestehende Diagramme.
  Weitere Informationen hierzu finden Sie im Abschnitt: Daten exportieren (Seite 280).

Die Automatismen sind unter dem Menüpunkt "Automatische Modellerstellung" zusammengefasst. Wenn sie von dort aufgerufen werden, entstehen die neuen Diagramme auf oberster Projektebene bzw. bei Angabe einer Ordnerhierarchie relativ zum Systemordner "Diagramme".

Wenn die Automatismen aus dem Kontextmenü eines Diagramm-Unterordners aufgerufen werden, werden alle neuen Diagramme relativ zu diesem ausgewählten Ordner angelegt. Da der Parameterimport keine neuen Diagramme erzeugt, ist er im Kontextmenü nicht enthalten.

Siehe auch

Aufbau der Steuerdatei "procedure.cfg" für Import aus ZIP-Datei (Seite 278)

4.1 Vorlagen

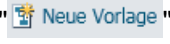
In Vorlagen werden immer wiederkehrende Funktionen eines Simulationsmodells, z. B. Teile eines Diagramms so aufbereitet, dass Sie diese als Vorlage für Diagramme beim Erstellen von Projekten verwenden können. In Vorlagen können Sie alle Elemente verwenden, die Sie auch in Diagrammen verwenden:

- Komponenten
- Makrokomponenten
- Controls
- Grafiken

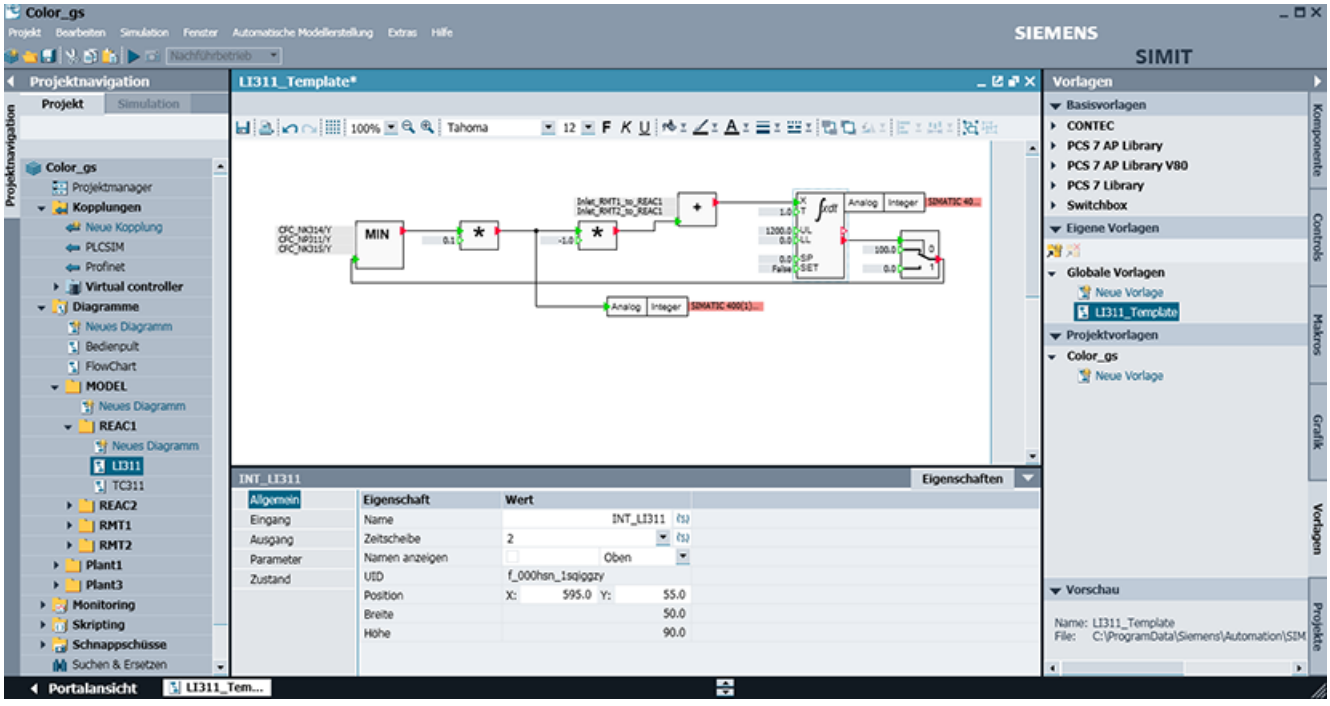
Im Unterschied zu Diagrammen werden in Vorlagen Platzhalter für verschiedene Elemente verwendet, die in den Komponenten der Vorlage enthalten sind. Beim Instanziieren einer Vorlage wird ein Diagramm erzeugt, in dem die Platzhalter durch Werte, Signalnamen etc. ersetzt worden sind. Es gibt folgende Möglichkeiten, Vorlagen zu instanziieren:

- über Drag & Drop der Vorlage in das Projekt
- über den Import von Dateien
- über das Simulationsmodell.

Zum Erstellen von Vorlagen gehen Sie folgendermaßen vor:

1. Öffnen Sie die Task-Card "Vorlagen" im Projektmanager.
2. Öffnen Sie den Vorlageneditor zum Erstellen einer neuen Vorlage mit einem Doppelklick auf "

".
3. Öffnen Sie eine Vorlage zum Bearbeiten mit einem Doppelklick auf die Vorlage in der Task-Card.
4. Bearbeiten Sie die Vorlage mit den Komponenten, Controls, Makrokomponenten, Grafiken, Vorlagen und Projekten, die Ihnen mit den Ressourcen des Vorlageneditors zur Verfügung stehen.

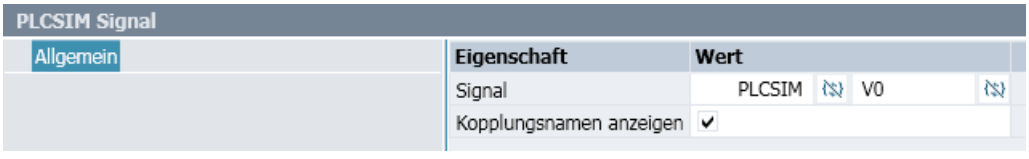


Vorlagen im Bereich "Basisvorlagen" können im Vorlageneditor nur zur Ansicht geöffnet werden. Bearbeiten ist nicht möglich. Als Hinweis darauf wird die Vorlage im Editor mit einer weißen Titelzeile geöffnet. Wenn Sie eine Basisvorlage bearbeiten möchten, kopieren Sie diese in den Bereich "Eigene Vorlagen".

Der Unterschied zwischen Vorlagen und Diagrammen ist, dass in Vorlagen Platzhalter benutzt werden.

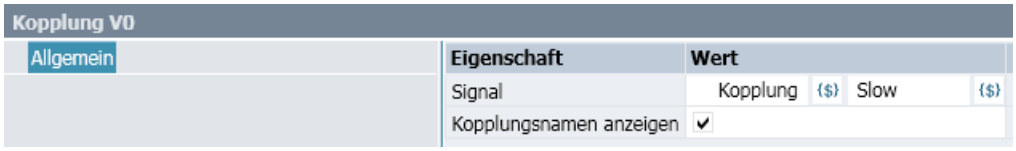
Beispiel:

1. Erstellen Sie eine neue Vorlage im Bereich "Globale Vorlagen".
2. Öffnen Sie das Beispielprojekt "Aufzug" in der Task-Card "Projekte".
3. Öffnen Sie das Diagramm "Hauptantrieb".
4. Kopieren Sie den gesamten Inhalt des Diagramms in die geöffnete Vorlage.
   In den Eigenschaften des Ausgangskonnektors PLCSIM V0 sind nun die Eingabefelder für Quelle (Kopplung) und Name des Signals um das Symbol "" zum Setzen von Platzhaltern erweitert.



5. Klicken Sie auf die beiden Symbole, um das Signal auf Platzhalter umzuschalten.

6. Geben Sie als Platzhalter "Kopplung" und "Slow" ein.



7. Setzen Sie entsprechende Platzhalter für die anderen Eingangs- und Ausgangssignale und für den globalen Konnektor.
   Benutzen Sie dabei immer den gleichen Platzhalter ("Kopplung") für die Ein- und Ausgangssignale.

Mit dem Platzhaltersymbol können Sie sowohl alle Eingänge und Parameter von Komponenten und Makrokomponenten, wie auch Signale in Controls und Animationen und die Namen von Komponenten als Platzhalter definieren.

Für einige Controls können Sie auch den Typ, die Voreinstellung und die Skalierung als Platzhalter definieren. Der folgenden Tabelle können Sie entnehmen, welche Eigenschaften für Controls als Platzhalter möglich sind.

Tabelle 4-1 Platzhalter von Controls

Control	Name	Zeitscheibe	Typ	Voreinstellung	Skalierung	Anschluss	Ziel
Taster	X	X	X				
Taster mit Bild	X	X	X				
Schalter	X	X	X	X			
Schalter mit Bild	X	X	X	X			
Stufenschalter	X	X		X			
Stufenschalter mit Bild	X	X		X			
Digitaleingabe	X	X		X			
Schieber	X	X		X	X	X	
Balkenanzeige	X	X			X	X	
Binäranzeige	X	X				X	
Analoganzeige	X	X				X	
Digitalanzeige	X	X				X	
Aktion	X						X

Der Typ und die Voreinstellung von Controls sind in der Benutzeroberfläche zum Teil als sprachabhängige Aufzählungen realisiert. Wenn Sie diese Eigenschaften als Platzhalter definieren, müssen Sie bei der Ersetzung die Position des entsprechenden Begriffs in der Aufzählung angeben.

Tabelle 4-2 Parameterwert für Typ Öffner/Schließer

Typ	Wert
Öffner	0
Schließer	1

Tabelle 4-3 Parameterwert für Voreinstellung Aus/Ein

Voreinstellung	Wert
Aus	0
Ein	1

4.1.1 Suchen und Ersetzen in Vorlagen

Wählen Sie im Kontextmenü der Vorlage den Befehl "Suchen & Ersetzen", um Komponenten innerhalb einer Vorlage austauschen.

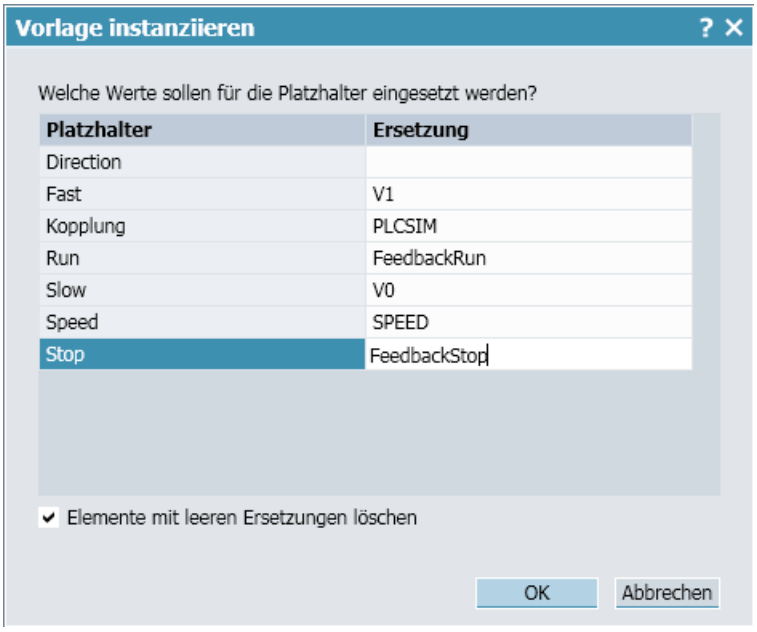
Damit wird der Editor "Suchen & Ersetzen" mit voreingestelltem Fokus auf die Vorlage geöffnet.

4.1.2 Vorlagen instanziieren mit Eingabe von Ersetzungen

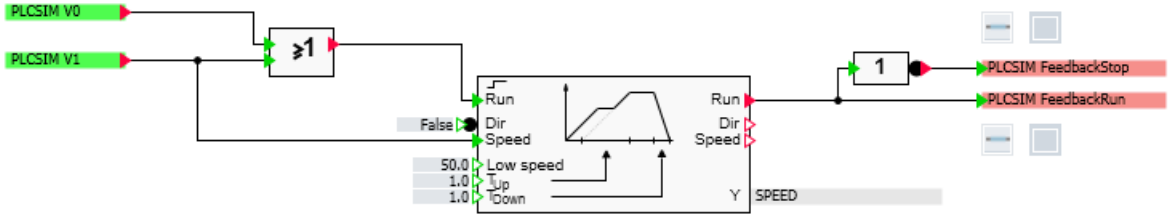
Vorlagen werden dadurch instanziiert, dass Platzhalter durch geeignete Werte ersetzt werden. Komponenten, deren Name nicht als Platzhalter definiert ist, erhalten den Namen, den sie auf der Vorlage erhalten haben, ggf. ergänzt um eine fortlaufende Nummer, wodurch eindeutige Komponentennamen entstehen.

Vorlagen können Sie dadurch instanziieren, dass Sie sie im Projektmanager per "Drag & Drop" aus der Task-Card in einen Projektordner ziehen. Öffnen Sie den Projektmanager und die Task-Card Vorlagen. Wenn Sie nun die oben erstellte Vorlage in einen Projektordner ziehen, öffnet sich der Dialog gemäß der folgenden Abbildung. Es wird eine zweispaltige Liste mit den in der Vorlage enthaltenen Platzhaltern angezeigt. In die zweite Spalte tragen Sie nun die Ersetzung für den jeweiligen Platzhalter ein.

Wenn die Option "Elemente mit leeren Ersetzungen löschen" gesetzt ist, werden Elemente, bei denen für einen ihrer Platzhalter keine Ersetzung in der Tabelle enthalten ist, gelöscht. Beispielsweise werden globale Konnektoren und Peripheriekonnektoren gelöscht, wenn kein Signal angegeben ist. In der folgenden Abbildung trifft dies beispielsweise für den Platzhalter Direction zu. Sie können somit Ihre Vorlagen so anlegen, dass Sie mehrere Anwendungsfälle abdecken und die Vorlage für einen bestimmten Anwendungsfall instanziieren, indem Platzhalter für die anderen Anwendungsfälle nicht ersetzt werden.



Nach dem Bestätigen des Dialogs sehen Sie ein neues Diagramm mit dem Namen der Vorlage im Projektmanager. Im geöffneten Diagramm sehen Sie die instanziierte Simulation, die bis auf das fehlende Ausgangssignal *Direction* identisch mit der Simulation des Hauptantriebs ist, wie in der folgenden Abbildung. Der Ausgangskonnektor wurde, da ohne Ersetzung, nicht instanziiert. Hätten Sie im Instanziierungsdialog die "Elemente mit leeren Ersetzungen löschen" abgewählt, wäre für dieses Signal ein Konnektor ohne Signalwert, d. h. ein leerer Konnektor, in der Instanz erzeugt worden.



4.1.3 Anlegen von Tabellen aus der Vorlage heraus

Wenn Sie zu einer Vorlage eine passende Tabelle erstellen möchten, mit deren Hilfe diese Vorlage dann mehrfach instanziiert werden kann, bietet Ihnen SIMIT die Möglichkeit, eine Tabelle anzulegen, die bereits eine zur Vorlage passende Titelzeile besitzt. Rufen Sie dazu im Kontextmenü der Vorlage den Befehl "Export nach Excel" auf.

Je nachdem, ob auf Ihrem Rechner Microsoft Excel installiert ist oder nicht, unterscheidet sich der weitere Ablauf.

- **Microsoft Excel vorhanden**
  Es wird direkt in Excel eine neue Mappe geöffnet, die im ersten Blatt in der ersten Zeile die für die ausgewählte Vorlage benötigten Einträge vorgibt. Ergänzen Sie diese Tabelle und speichern Sie sie an einem beliebigen Ort, um sie dann beim Tabellen-Import verwenden zu können.
- **Microsoft Excel nicht vorhanden**
  Es wird eine Excel-Datei im *.xlsx-Format erzeugt, die im ersten Blatt in der ersten Zeile die für die ausgewählte Vorlage benötigten Einträge vorgibt. Speichern Sie diese Datei an einem beliebigen Ort, um sie dann später – gegebenenfalls auf einem anderen Rechner – mit Excel zu bearbeiten und zu ergänzen.

Die erzeugten Tabellen enthalten in den ersten drei Spalten die Begriffe HIERARCHY, TEMPLATE und CHART, die folgenden Spalten enthalten die Namen der in der Vorlage verwendeten Platzhalter. Der Name des Templates ist in der zweiten Zeile bereits eingetragen; alle anderen Angaben müssen Sie selbst ergänzen.

	A	B	C	D	E
1	HIERARCHY	TEMPLATE	CHART	GATEWAY	NAME
2		Rail-S4			
3					
4					

Hinweis

Achten Sie darauf, dass beim Import dieser Tabelle das Profil auf "Platzhalter in der ersten Zeile definiert" eingestellt ist.

4.1.4 Angabe einer Ordnerhierarchie für Vorlagen

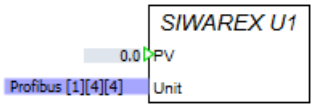
Sie können in Ihren Tabellen angeben, in welchem Ordner eine Vorlage gesucht werden soll. Geben Sie dazu für den Platzhalter TEMPLATE die Ordnerhierarchie vollständig an, wie sie auch in der Task-Card "Vorlagen" zu sehen ist. Als Trennzeichen ist der umgekehrte Schrägstrich "\" zu verwenden. Wenn Sie sich auf eine Vorlage beziehen möchten, die in keinem Unterordner liegt, stellen Sie dem Namen der Vorlage einen umgekehrten Schrägstrich voran.

Falls die Angabe der Vorlage kein Trennzeichen enthält, wird wie bisher in allen Ordnern und Unterordnern nach einer Vorlage dieses Namens gesucht. Stellen Sie in diesem Fall sicher, dass es keine zwei Vorlagen mit gleichem Namen in unterschiedlichen Unterordnern gibt. Grundsätzlich werden die drei Paletten der Task-Card "Vorlagen" in folgender Reihenfolge durchsucht:

1. Projektvorlagen
2. Eigene Vorlagen
3. Basisvorlagen

4.1.5 Adressierung eines Moduls über E/A-Adresse

Die Identität eines Moduls am PROFIBUS oder PROFINET wird durch die Slave- bzw. Device-Nummer und die Slot-Nummer bestimmt. Dementsprechend sind diese beiden Angaben in den Unit-Konnektor einzutragen, wenn er zur Identifikation eines Gerätes, wie beispielsweise einer Waage vom Typ SIWAREX-U benutzt wird.



Beim Einsatz solcher Geräte auf Vorlagen ist diese Information nicht immer verfügbar. Stattdessen kann die Anfangsadresse des Moduls bekannt sein. Daher ist es auf Vorlagen jetzt auch erlaubt, das Modul über seine Anfangsadresse zu identifizieren, also beispielsweise statt "[1][4][4]" "EW512" oder "AW512" anzugeben.

Beim Starten der Simulation ist die Kenntnis der Adresse mit Slave- bzw. Device-Nummer und Slot-Nummer erforderlich. Die Angabe einer Anfangsadresse wird daher während der Instanziierung einer Vorlage automatisch in die Adresse mit Slave- bzw. Device-Nummer und Slot-Nummer umgesetzt. Falls während der Instanziierung keine Kopplung mit der passenden Adresse existiert, können Sie diese automatische Umwandlung auch noch nachträglich über den Befehl "Extras > Kopplungssignale zuweisen (Seite 959)" vornehmen.

4.1.6 Indirekte Adressierung

In den Input- und Output-Konnektoren kann auf Vorlagen eine zusätzliche Angabe stehen, die den Zugriff auf Signale erlaubt, die relativ zu einer bekannten Adresse liegen. Dieser Ausdruck hat die Form [$+`n`:`RW`], wobei die Symbole Folgendes bedeuten:

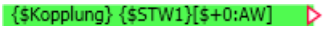
- `n` : der zu addierende Offset in Byte
- `R` : die Ein- oder Ausgangskennung, also E, I, A oder Q
- `W` : die Datenbreite, also B, W oder D

Wenn also beispielsweise die Adresse eines Steuerwortes EW560 bekannt ist, kann das zugehörige Zustandswort AW560 wie folgt angegeben werden:

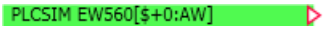
EW560[$+0:AW]

Oder, falls das Steuerwort EW560 in der Kopplung den symbolischen Namen STW1 hat, kann es auch in der Form STW1[$+0:AW] angegeben werden.

Wenn zum Zeitpunkt, zu dem die Vorlagen instanziiert werden, die Kopplung bereits existiert, wird versucht, die Ersetzung mit dem entsprechenden Signal vorzunehmen. Falls nicht, bleibt dieser Ausdruck auch nach Ersetzung des Signals, auf das er sich bezieht, erhalten. Existiert beispielsweise in einer PLCSIM-Kopplung das Steuerwort STW1 mit seiner Adresse EW560, aber kein zugehöriges Zustandswort, dann wird der Konnektor



wie folgt ersetzt:



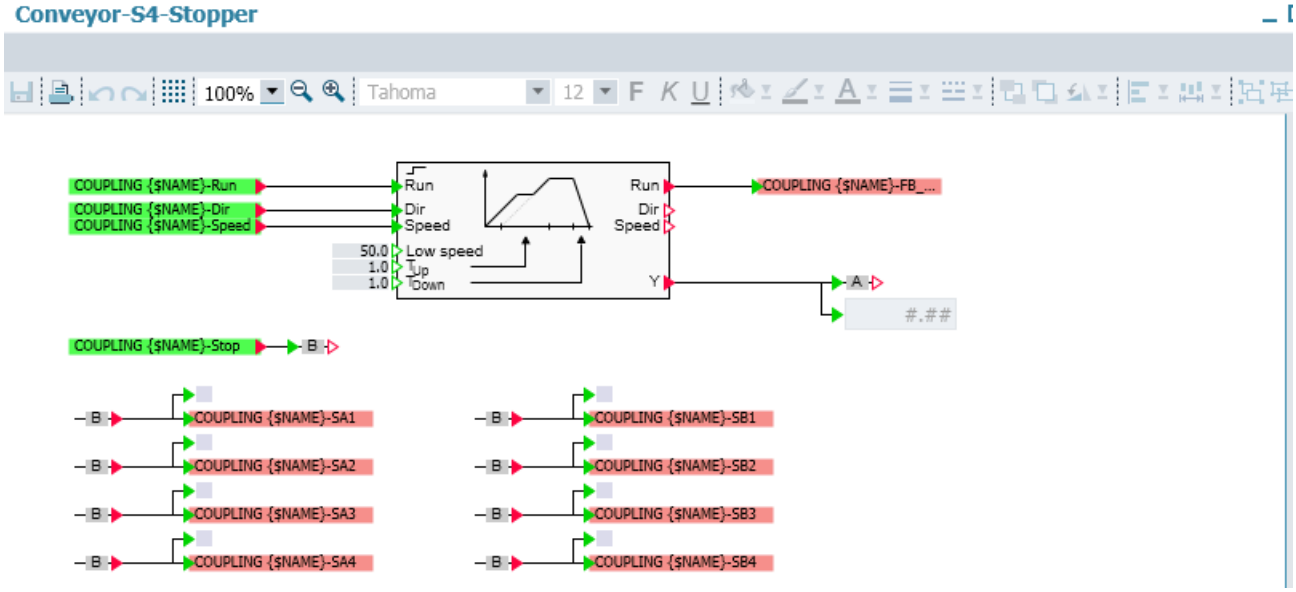
Sie können die vollständige Ersetzung aber jederzeit auch noch nachträglich anstoßen, indem Sie die Funktion "Extras > Kopplungssignale zuweisen (Seite 959)" aufrufen.

Hinweis

Die indirekte Adressierung wird nur vom Virtual Controller und von den folgenden Kopplungen unterstützt: PROFIBUS DP, PROFINET IO, PRODAVE und PLCSIM.

4.1.7 Öffnen von Basisvorlagen im Editor

Vorlagen in der Palette "Basisvorlagen" können Sie jetzt direkt über das Kontextmenü im Vorlageneditor öffnen. Allerdings werden die so geöffneten Basisvorlagen nur zur Ansicht geöffnet, können also nicht editiert werden. Als Hinweis darauf wird die Vorlage im Editor mit einer weißen Titelzeile geöffnet.



Wenn Sie eine Basisvorlage editieren wollen, kopieren Sie diese zunächst in die Palette "Eigene Vorlagen".

4.2 Instanziierung von Vorlagen aus Dateien oder dem Simulationsmodell

4.2.1 Vorlagen aus Dateien instanziieren

Voraussetzung

- SIMIT-Projekt ist geöffnet

Vorgehen

Um Vorlagen aus Dateien zu instanziieren, gehen Sie folgendermaßen vor:

1. Wählen Sie "Automatische Modellerstellung > Vorlagen instanziieren".
   Das Dialogfeld "Instanziiere Vorlagen" wird geöffnet.
2. Wählen Sie die Option "Import-Datei".
3. Geben Sie den Speicherort der Import-Datei ein.
4. Um Tabellen zu importieren, gehen Sie folgendenmaßen vor:
   - Um eine Vorlage entsprechend der Tabelle mehrfach in einem Diagramm zu instanziieren, wählen Sie die vorzunehmende Gruppierung im Diagramm, horizontal oder vertikal.
   - Um die Instanzen *horizontal* aneinanderzureihen, geben Sie die *maximale Breite* des Diagramms an.
   - Um die Instanzen *vertikal* aneinanderzureihen, geben Sie die *maximale Höhe* des Diagramms an.
5. Um IEA-Dateien zu imporieren, gehen Sie folgendenmaßen vor:
   - Geben Sie den Namen der Kopplung an, die für die E/A-Signale verwendet werden soll.
   - Geben Sie eine zur Import-Datei passende Vorlage ein.
   - Wählen Sie das Trennzeichen.
6. Um CMT-Dateien zu importieren, gehen Sie folgendenmaßen vor:
   - Geben Sie den Namen der Kopplung an, die für die E/A-Signale verwendet werden soll.
   - Geben Sie den Ordner an, in dem nach den in der CMT-Datei referenzierten SIMIT-Vorlagen gesucht wird.
7. Deaktivieren Sie unter "Vorschau>>" bei Bedarf einzelne Diagrammordner, Diagramme oder Ersetzungen.
8. Um den Import zu starten, klicken Sie auf "Importieren".

Ergebnis

Die Vorlagen wurden aus einer Datei importiert und instanziiert. Die erzeugten Diagramme werden im Ordner "Diagramme" abgelegt.

Siehe auch

Tabellenimport (Seite 261)

IEA-Import (Seite 264)

CMT-Import (Seite 266)

Dialogfeld "Instanziiere Vorlagen" (Seite 974)

4.2.2 Tabellenimport

Vorlagen können mit Hilfe von Tabellen instanziiert werden. In diesen Tabellen stehen die Platzhalter in der ersten Zeile. In den weiteren Zeilen stehen die Ersetzungen für die Platzhalter und Informationen über die zu verwendenden Vorlagen und die zu erzeugenden Instanzen.

In allen Tabellen wird mit jeder Zeile das Instanziieren einer Vorlage definiert.

Folgende Tabellenformate werden unterstützt:

- ***.xls**
  Das Microsoft Excel-Format Office 97 – 2003. Nur das erste Blatt der Mappe wird berücksichtigt. Dieses Dateiformat ist nur verfügbar, falls Microsoft Excel auf Ihrem Rechner installiert ist.
- ***.xlsx**
  Microsoft Excel-Format. Nur das erste Blatt der Mappe wird berücksichtigt. Dieses Dateiformat ist auch verfügbar, falls Microsoft Excel nicht auf Ihrem Rechner installiert ist.
- ***.txt**
  Tabulatorgetrennte Liste im Textformat

Hinweis

Wenn Sie noch Dateien im Format *.csv von Projekten mit früheren Versionen von SIMIT verwenden möchten, müssen Sie diese in das tabulatorgetrennte *.txt-Format oder in das *.xls- oder *.xlsx-Format konvertieren.

Hinweis

Wenn Sie eine Datei im Format *.xls oder *.xlsx importieren, die noch in Excel geöffnet ist, wird nicht der aktuelle Bearbeitungsstand, sondern der letzte gespeicherte Stand dieser Datei importiert.

Import nach SIMIT

Es gibt folgende Möglichkeiten, einen Tabellenimport nach SIMIT durchzuführen:

- Wählen Sie in der Portalansicht "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie in der Projektansicht den Menübefehl "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie im Kontextmenü eines Diagrammordners "Automatische Modellerstellung > Vorlagen instanziieren".

Platzhalter in der ersten Zeile

Mit Spaltenüberschriften wird die Bedeutung der Spalte festgelegt.

	A	B	C	D	E	F
1	HIERARCHY	TEMPLATE	CHART	P1	P2	P3
2	H1	T1	C1	1.0	E0.0	
3	H2	T2	C2	3.0		A1.0

Die ersten drei Spalten sind in ihrer Bedeutung definiert und ihre Überschriften sind fest vorgeben:

- Spalte A (HIERARCHY) enthält die im Projekt anzulegende Ordnerhierarchie.
- Spalte B (TEMPLATE) enthält die zu verwendende Vorlage.
- Spalte C (CHART) enthält den Namen des zu instanziierenden Diagramms.

Ab Spalte D kann die Tabelle beliebig viele weitere Spalten mit Ersetzungen für die Platzhalter enthalten. Der Platzhalter steht in jeder Spalte als Überschrift. In diesem Tabellenprofil gibt es keine Beschränkungen in der Anzahl der Platzhalter.

Hinweis

Vorlagen (*TEMPLATE*) geben Sie mit der vollständigen Ordnerhierarchie an. Als Trennzeichen ist der umgekehrte Schrägstrich "\" zu verwenden. Um sich auf eine Vorlage zu beziehen, die in keinem Unterordner liegt, stellen Sie dem Namen der Vorlage einen umgekehrten Schrägstrich voran.

Wenn eine Vorlage ohne Trennzeichen angegeben ist, wird in allen Ordnern und Unterordnern nach einer Vorlage dieses Namens gesucht. Deshalb darf es keine zwei Vorlagen mit gleichem Namen in unterschiedlichen Unterordnern geben. Die drei Paletten der Task-Card "Vorlagen" werden in folgender Reihenfolge durchsucht:

1. Projektvorlagen
2. Eigene Vorlagen
3. Basisvorlagen

Beispiel für das Instanziieren mit einer Tabelle

Ein Beispiel für eine Tabelle finden Sie auf der SIMIT Software-CD im Ordner "_Beispiele \SMD". Zur Instanziierung der Vorlagen nach Tabelle wählen Sie den Menübefehl "Automatische Modellerstellung > Vorlagen instanziieren". Das Dialogfeld "Instanziiere Vorlagen" wird geöffnet.

Wählen Sie als Tabelle die Datei "Beispiel.txt". Um eine Vorlage entsprechend der Tabelle mehrfach in einem Diagramm zu instanziieren, ist die Angabe der vorzunehmenden Gruppierung und des zur Verfügung stehenden Platzes auf dem Diagramm erforderlich. Um die Instanzen *horizontal* aneinanderzureihen, geben Sie die *maximale Breite* des Diagramms an. Um Instanzen *vertikal* aneinanderzureihen, geben Sie die *maximale Höhe* des Diagramms an.

Wenn Sie nun die Vorschau mit der Schaltfläche "Vorschau>>" öffnen, dann sehen Sie im linken Bereich der Vorschau, welche Ordnerhierarchie und welche Diagramme instanziiert werden. Für jedes Diagramm können Sie in der rechten Hälfte der Vorschau die Liste der Ersetzungen laut Tabelle sehen und kontrollieren, bevor tatsächlich Diagramme instanziiert werden. Zum Öffnen der Liste selektieren Sie das entsprechende Diagramm. Sie können jetzt entscheiden, ob Sie alle Informationen aus der Tabelle übernehmen oder aber gezielt einzelne Ersetzungen abwählen möchten.



Wenn ein Diagramm in der linken Baumansicht abgewählt ist, wird es nicht instanziiert. Sie können über die Selektion eines Ordners auch mehrere darin enthaltene Diagramme an- und abwählen. Sie erkennen an der Darstellung des Optionskästchens, ob ein Diagramm oder Ordner angewählt ist (☑) oder abgewählt ist (☐) oder ob in einem der unterlagerten Ordner mindestens ein Optionskästchen abgewählt ist (■).

Zusätzlich können Sie in der rechten Tabelle einzelne Ersetzungen an- und abwählen. Wenn Sie eine Ersetzung abwählen, hat das die gleiche Auswirkung, als wäre diese Ersetzung in der Tabelle nicht vorhanden.

4.2.3 IEA-Import

Wenn Sie PCS 7 einsetzen und mit Messstellentypen oder Musterlösungen arbeiten, erzeugen Sie für jeden CFC-Plan, der auf einer Musterlösung oder einem Messstellentyp basiert, einen passenden Plan in SIMIT erzeugen.

Mit SIMIT werden verschiedene Vorlagen ausgeliefert, die mit Messstelltypen der Bibliotheken von PCS 7 korrespondieren. Sie finden passende Vorlagen in den Verzeichnissen der Basisvorlagen:

- "PCS 7 Library" enthält Vorlagen kompatibel zur Standard-Bibliothek von PCS 7.
- "PCS 7 AP Library V71" enthält Vorlagen passend zur AP Library von PCS 7 V7.1.
- "PCS 7 AP Library V80" enthält Vorlagen passend zur AP Library von PCS 7 V8.0.
- "PCS 7 AP Library V81" enthält Vorlagen passend zur AP Library von PCS 7 V8.1.
- "PCS 7 AP Library V82" enthält Vorlagen passend zur AP Library von PCS 7 V8.2.
- "PCS 7 AP Library V90" enthält Vorlagen passend zur AP Library von PCS 7 V9.0.

Wenn Sie eines der Standard-Templates aus einer PCS 7-Bibliothek in Ihrem Projekt verwendet haben, können Sie diese Vorlagen nutzen und die passende Simulation dafür erstellen. Die Vorgehensweise ist im Folgenden am Beispiel eines Motorantriebes (MOTOR) erläutert. Zwei Beispiel-IEA-Dateien aus einem PCS 7-Projekt finden Sie auf der SIMIT-Software-CD im Ordner "_Beispiele\SMD".

Import nach SIMIT

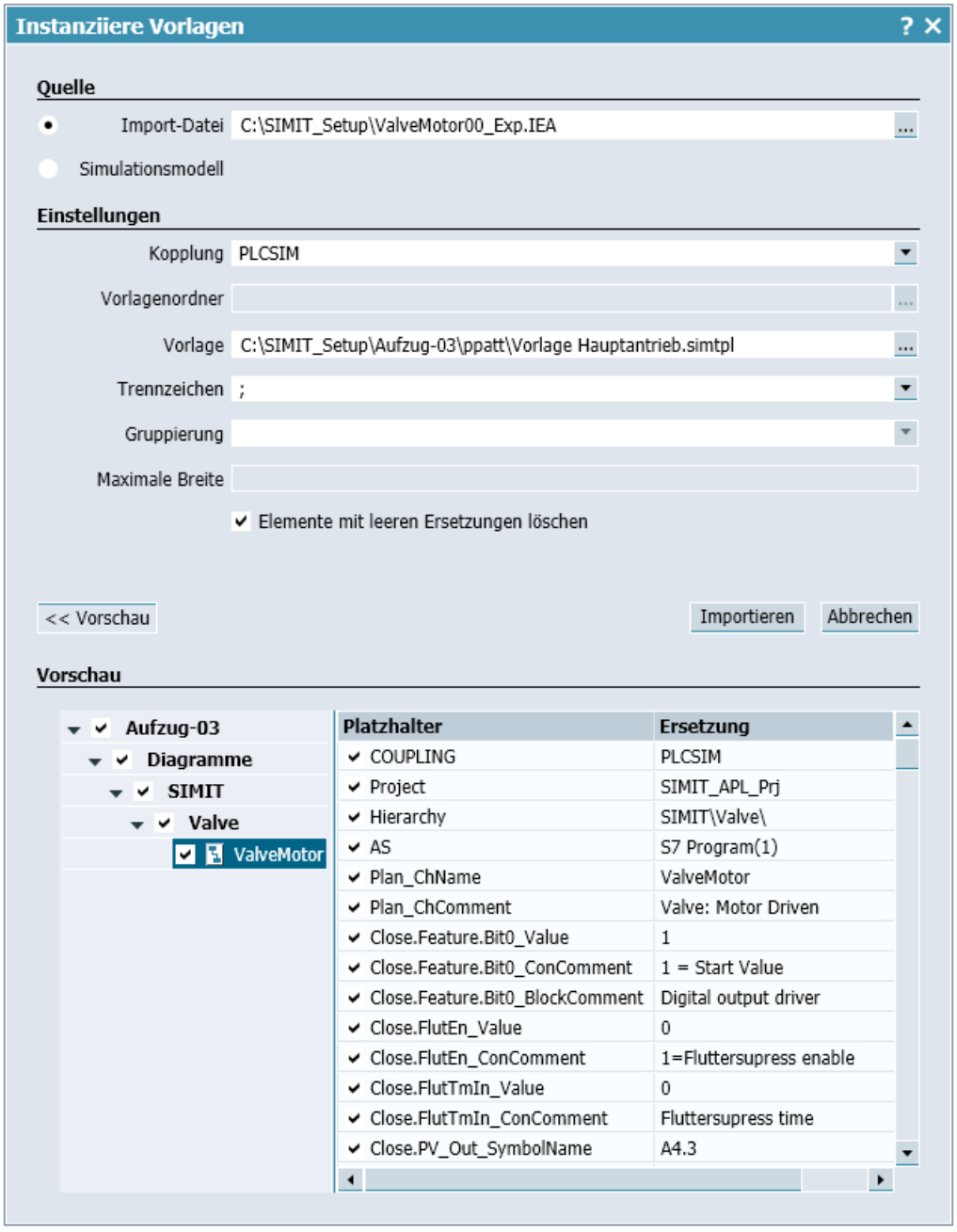
Es gibt folgende Möglichkeiten, einen IEA-Import nach SIMIT durchzuführen:

- Wählen Sie in der Portalansicht "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie in der Projektansicht den Menübefehl "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie im Kontextmenü eines Diagramm-Ordners "Automatische Modellerstellung > Vorlagen instanziieren".

Das Dialogfeld "Instanziiere Vorlagen" wird geöffnet.

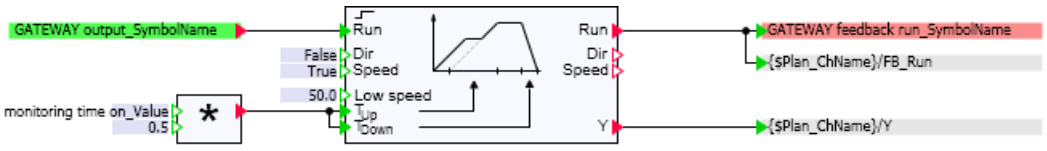
Geben Sie den Namen der Kopplung an, die für die E/A-Signale verwendet werden soll.

In der Vorschau sehen Sie, dass je CFC-Plan ein Diagramm instanziiert wird. Dieses Diagramm hat denselben Namen wie der CFC-Plan und wird in einer zum PCS 7-Projekt identischen Ordnerhierarchie abgelegt. Sie können somit entscheiden, ob Sie alle Informationen aus der IEA-Datei übernehmen wollen oder aber gezielt einzelne Ersetzungen abwählen.

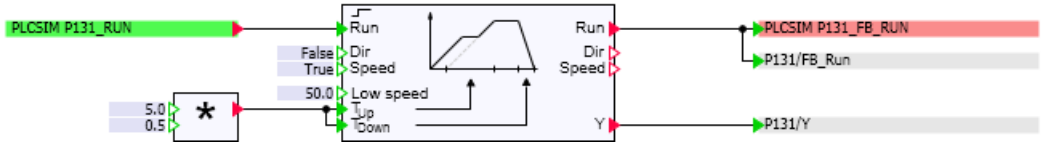


In der folgenden Abbildung sehen Sie unter (a) die Vorlage und unter (b) die daraus erzeugte Instanz.

(a)



(b)



Importieren Sie anschließend die zweite zur Vorlage VALVE passende Beispieldatei "VALVE00_Exp.IEA". Sie erhalten dann weitere Diagramme im SIMIT-Projekt in der schon beim ersten Import angelegten Ordnerhierarchie.

Wenn Sie modifizierte oder eigene Messstellentypen in Ihrem PCS 7-Projekt einsetzen, können Sie natürlich auch diese IEA-Dateien mit entsprechend passenden SIMIT-Vorlagen verwenden. Erzeugen Sie passende Vorlagen beispielsweise in "Eigene Vorlagen" oder kopieren Sie die geeigneten Basisvorlagen dorthin und modifizieren Sie die Kopien.

4.2.4 CMT-Import

CMT steht für "Control Module Type" (Einzelsteuereinheitstyp) in PCS 7. Mit einem Einzelsteuereinheitstyp werden CFC-Planvorlagen erstellt und mit variablen Bestandteilen versehen. Diese CFC-Planvorlagen werden dann durch Instanziierung vervielfältigt und individualisiert. Dazu erstellt PCS 7 eine exportierbare XML-Datei, die sowohl alle Einzelsteuereinheitstypen als auch alle im Projekt enthaltenen Instanzen, deren Variablenersetzungen und Varianten enthält.

Export aus PCS 7

Diese Funktionalität wird in PCS 7 zum Massendaten-Engineering benutzt, oder um Projektierungsinformationen mit anderen Projektierungstools, wie z. B. COMOS, auszutauschen.

Hinweis

Die für PCS 7 notwendige Funktion "XML-Transfer" ist nicht Teil des Lieferumfangs von SIMIT und kann über SIOS (https://support.industry.siemens.com/cs/ww/de/view/63481413) bezogen werden:

1. Kategorie "PCS 7 Software Updates" wählen.
2. PCS 7-Version wählen.
3. "Engineering System Collection" Vx.y herunterladen.
   Die "Engineering System Collection" enthält mehrere Komponenten, die Sie über den Installationsassistenten auswählen. Für die ausschließliche Nutzung mit SIMIT benötigen Sie nur die Komponente "XML-Transfer".

Nach der Installation kann der Export direkt aus dem PCS 7-Projekt im SIMATIC Manager (aus der technologischen Sicht) mit dem Kontextmenü "XML exportieren" erzeugt werden.

Die exportierte XML-Datei kann nun in SIMIT importiert werden, um die instanziierten Einzelsteuereinheitstypen in der Simulation zu erzeugen und zu testen.

Import nach SIMIT

Es gibt folgende Möglichkeiten, einen CMT-Import nach SIMIT durchzuführen:

- Wählen Sie in der Portalansicht "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie in der Projektansicht den Menübefehl "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie im Kontextmenü eines Diagramm-Ordners "Automatische Modellerstellung > Vorlagen instanziieren".

Das Dialogfeld "Instanziiere Vorlagen" wird geöffnet.

Nach dem Instanziieren der Vorlagen sind anders als in PCS 7 die Diagramme nicht mehr als "Vorlageninstanzen" identifizierbar. Deshalb ist es nicht möglich, nachträglich Varianten umzuschalten oder individuelle Anpassungen beizubehalten. Ein Abgleich ist nicht möglich.

Wenn sich das Projekt in PCS 7 ändert, muss die XML-Datei erneut exportiert und neu in SIMIT importiert werden. Bei diesem Import können dann alle bestehenden Instanzen durch neue Diagramme ersetzt werden. Individuelle Anpassungen, die nach dem letzten CMT-Import auf den bereits instanziierten Diagrammen vorgenommen wurden, gehen dabei verloren.

Vorlagenerstellung

Um Variablen und Platzhalter in den Vorlagen zu definieren, gibt es folgende Möglichkeiten im Vorlageneditor:

- Dem Wert eines Eigenschaftsfeldes kann eine Variable zugewiesen werden. Dazu muss rechts daneben das Symbol "{$}" aktiviert werden. {$} bedeutet, dass der gesamte Feldinhalt als Variablenname interpretiert wird.
- Wenn mehrere Variablen oder feste und variable Bestandteile miteinander kombiniert werden sollen, kann in das jeweilige Feld ein Variablenname auch mittels der Notation {$Variablenname} eingetragen werden.

Damit können Ersetzungen aus der XML-Datei in der Instanz einer Vorlage spezifiziert werden.

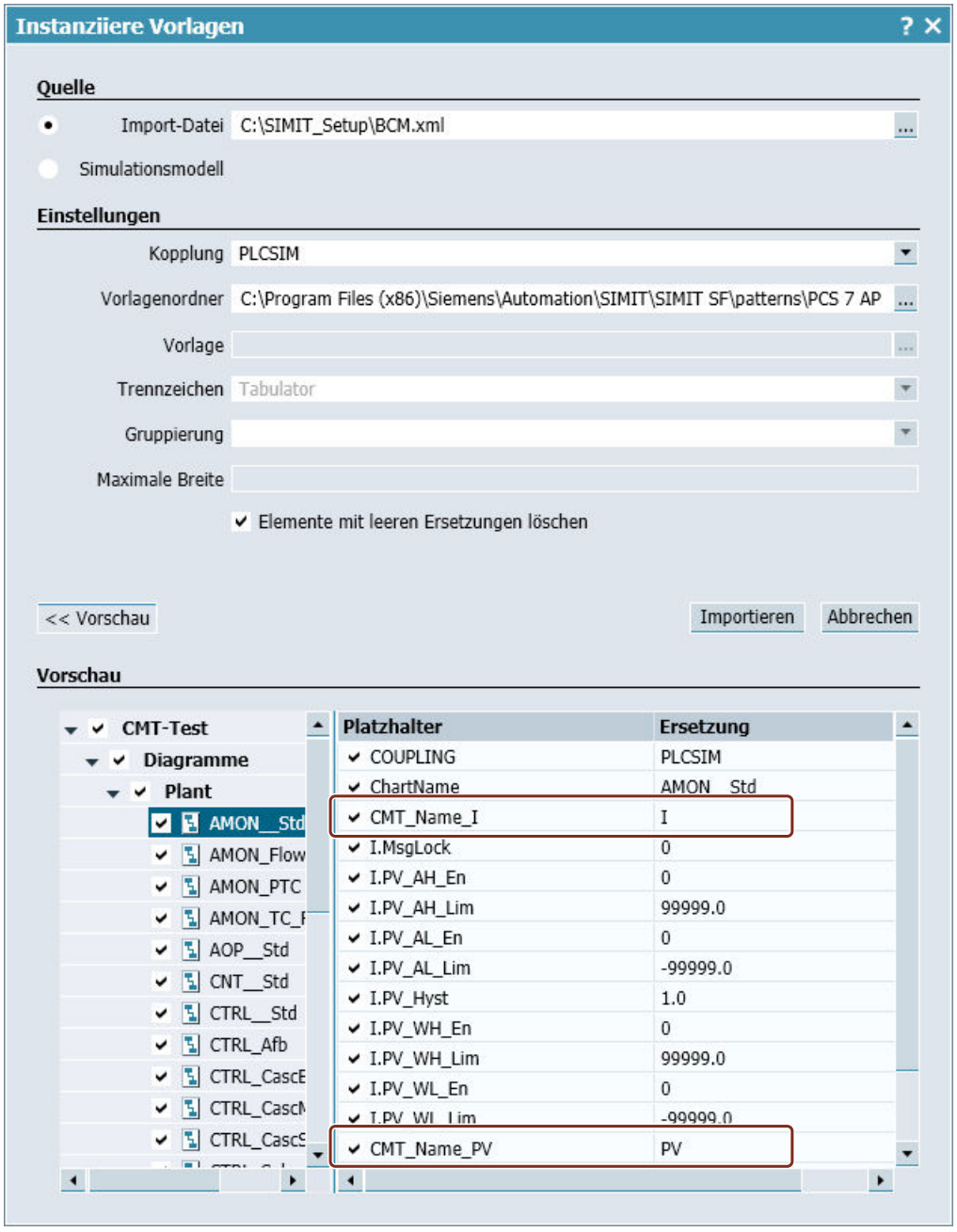
Varianten und Mehrfachverbindungen in CMTs

Das Abbilden von Varianten eines CMT wird in SIMIT folgendermaßen realisiert:

- Das Optionskästchen "Elemente mit leeren Ersetzungen löschen" ist aktiv.
- Die Variable "CMT_Name_x" ist nur dann in der Liste der Ersetzungen vorhanden, wenn diese Variante auch im CMT vorhanden ist.
  Dies gilt für alle von diesem CMT abgeleiteten Variablen und Parameter.

  Hinweis

  Das "x" bei "CMT_Name_x" ist als CMT-Name zu sehen und ist in der folgenden Abbildung "I" oder "PV".



Wenn in SIMIT eine Komponente in das Simulationsmodell eingebaut werden soll, die von einem optionalen CMT in PCS 7 abhängig ist, muss sich z. B. der Name der Komponente auf diese Variable beziehen.

Wenn der CMT in der XML-Datei fehlt, fehlen dessen Variablen, Parameter und Signale ebenfalls und in SIMIT entfallen dadurch automatisch alle Komponenten, die eine dieser Ersetzungen beinhalten. Dadurch ist das Varianten-Konzept der CMTs auch in SIMIT abbildbar.

Mehrfachverschaltungen auf CMTs in PCS 7 können in SIMIT mit Hilfe von Konnektoren realisiert werden. Da diese Varianten in PCS 7 nur alternativ verwendet werden können, kann

hier in SIMIT der Mechanismus der eben beschriebenen Varianten genutzt werden, um mehrere Ausgänge von Komponenten in der Vorlage auf einen Eingang zu verschalten.

Dazu werden Komponenten den Varianten zugeordnet, und deren Ausgänge mit Konnektoren auf den Eingang einer weiteren Komponente gelegt. Fällt jetzt in der Instanz einer der beiden CMTs weg, dann fehlt die entsprechende Komponente in SIMIT und der Konnektor bleibt unangeschlossen auf einem Diagramm liegen. Derartige Konnektoren sind für SIMIT nicht vorhanden und somit entsteht keine Doppelzuweisung, die sonst zu einer Konsistenzfehlermeldung führen würde.

4.2.5 Vorlagen aus dem Simulationsmodell instanziieren

Einleitung

Die Instanziierung aus dem Simulationsmodell nutzt Parameter der im Simulationsmodell verwendeten Komponenten. Für bestimmte Komponententypen können automatisch Vorlagen instanziiert werden, z. B. um zu prozesstechnischen Geräten die passende Simulation der Ansteuerungslogik zu erstellen.

Voraussetzung

- SIMIT-Projekt ist geöffnet.
- Die Parameter "TEMPLATE" und "HIERARCHY" sind in den Komponenten im Projekt vorhanden.

Vorgehen

Um Vorlagen aus dem Simulationsmodell zu instanziieren, gehen Sie folgendermaßen vor:

1. Wählen Sie "Automatische Modellerstellung > Vorlagen instanziieren".
   Das Dialogfeld "Instanziiere Vorlagen" wird geöffnet.
2. Wählen Sie die Option "Simulationsmodell".
3. Geben Sie den Namen der Kopplung an, die für die E/A-Signale verwendet werden soll.
4. Geben Sie den Vorlagenordner an, in dem nach den referenzierten SIMIT-Vorlagen gesucht wird.
5. Um den Import zu starten, klicken Sie auf "Importieren".

Ergebnis

Die Vorlagen wurden instanziiert.

Siehe auch

Dialogfeld "Instanziiere Vorlagen" (Seite 974)

4.3 Automatisierter Import

4.3.1 Diagramme aus ZIP- oder XML-Dateien importieren

Einleitung

Die Funktion "Automatisierter Import" unterstützt den Import von Diagrammen aus folgenden Quellen:

- ZIP
  Importiert Diagramme mit darin referenzierten Komponenten, auch wenn die Komponenten nicht in der SIMIT-Taskcard enthalten sind.
  Anwendungen wie COMOS können solche Import-Dateien erzeugen. Eine Import-Datei ist ein ZIP-Archiv.
- XML
  Importiert Diagramme aus einer XML-Datei. Beim Import können sowohl vorhandene Vorlagen für Diagramme instanziiert als auch individuell gestaltete Diagramme ohne Rückgriff auf Vorlagen erzeugt werden.
  Die XML-Datei muss dabei eine Beschreibung von Diagrammen auf Basis einer XML-Syntax enthalten.

Hinweis

Um die XML-Datei auf syntaktische Korrektheit zu überprüfen, führen Sie vor dem Import eine Validation mit geeigneten Werkzeugen durch.

Es gibt drei Möglichkeiten, einen automatisierten Import nach SIMIT durchzuführen:

- Wählen Sie in der Portalansicht "Automatische Modellerstellung > Automatisierter Import".
- Wählen Sie in der Projektansicht den Menübefehl "Automatische Modellerstellung > Automatisierter Import".
- Wählen Sie im Kontextmenü eines Diagramm-Ordners "Automatische Modellerstellung > Automatisierter Import".

Voraussetzung

- Projekt ist geöffnet.
- Eine für den automatisierten Import geeignete Datei liegt vor.

Vorgehen

Um Diagramme aus ZIP- oder XML-Dateien zu importieren, gehen Sie folgendermaßen vor:

1. Starten Sie den automatisierten Import über einen der oben genannten Befehle.
   Der Dialog "Automatisierter Import" wird geöffnet.
2. Um Diagramme aus einer ZIP-Datei zu importieren:
   - Wählen Sie als Dateiformat "*.ZIP" aus.
   - Wählen Sie gewünschte Datei aus.
3. Um Diagramme aus einer XML-Datei zu importieren:
   - Wählen Sie als Dateiformat "*.XML" aus.
   - Wählen Sie gewünschte Datei aus.
     Der Dialog "Generischer Import" wird geöffnet.
   - Wählen Sie die gewünschten Daten aus.
4. Klicken Sie auf "Importieren".

Ergebnis

Die Daten werden importiert.

Siehe auch

Syntax der XML-Datei (Seite 271)

Beispiele für XML-Dateien (Seite 276)

Aufbau der Steuerdatei "procedure.cfg" für Import aus ZIP-Datei (Seite 278)

4.3.2 Syntax der XML-Datei

Die zu importierende Datei muss als Textdatei im XML-Format zur Verfügung gestellt werden und folgender Document Type Definition entsprechen:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<!DOCTYPE GENERIC [
<!ELEMENT GENERIC (FOLDER | DIAGRAM | TEMPLATE | BUILDUP)*>
<!ATTLIST GENERIC
    VERSION CDATA #REQUIRED>
<!ELEMENT FOLDER (FOLDER | DIAGRAM | TEMPLATE | BUILDUP)*>
<!ATTLIST FOLDER
    NAME CDATA #REQUIRED>
<!ELEMENT BUILDUP (TEMPLATE+)>
<!ATTLIST BUILDUP
    INSTANCE CDATA #REQUIRED
    ALIGNMENT (HOR|VER) "HOR"
    WIDTH CDATA #IMPLIED
    HEIGHT CDATA #IMPLIED>
<!ELEMENT TEMPLATE (SUBST*)>
```

```
<!ATTLIST TEMPLATE
    NAME CDATA #REQUIRED
    INSTANCE CDATA #IMPLIED>
<!ELEMENT SUBST EMPTY>
<!ATTLIST SUBST
    NAME CDATA #REQUIRED
    VALUE CDATA #REQUIRED>
<!ELEMENT DIAGRAM (COMP)*>
<!ATTLIST DIAGRAM
    NAME CDATA #REQUIRED
    WIDTH CDATA #IMPLIED
    HEIGHT CDATA #IMPLIED>
<!ELEMENT COMP (POS+, TRANSFORM?, PROP*, DEFAULT*, PORT*)>
<!ATTLIST COMP
    UID CDATA #REQUIRED
    NAME CDATA #REQUIRED
    REF CDATA #IMPLIED
    CYCLE CDATA #IMPLIED>
<!ELEMENT TRANSFORM EMPTY>
<!ATTLIST TRANSFORM
    SCALEX CDATA #IMPLIED
    SCALEY CDATA #IMPLIED
    ROTATION CDATA #IMPLIED>
<!ELEMENT POS EMPTY>
<!ATTLIST POS
    X CDATA #REQUIRED
    Y CDATA #REQUIRED>
<!ELEMENT PROP EMPTY>
<!ATTLIST PROP
    NAME CDATA #REQUIRED
    VALUE CDATA #REQUIRED>
<!ELEMENT PORT (POS?, CONNECTION*)>
<!ATTLIST PORT
    NAME CDATA #REQUIRED>
<!ELEMENT DEFAULT EMPTY>
<!ATTLIST DEFAULT
    NAME CDATA #REQUIRED
    VALUE CDATA #REQUIRED>
<!ELEMENT CONNECTION EMPTY>
<!ATTLIST CONNECTION
    TYPE (LINE|IMPLICIT) "LINE"
    SOURCE CDATA #REQUIRED
    NAME CDATA #REQUIRED>
]>
```

Sie finden diese Definition in der Datei "Generic.dtd" im Ordner "_Beispiele/XML" auf Ihrer Installations-CD von SIMIT.

Eine gültige XML-Datei beginnt mit dem Element GENERIC. Als Attribut ist diesem Element die Versionsnummer 8.0 anzugeben.

Element	Attribut	Beschreibung
GENERIC		*Ordner*
	VERSION	In der hier beschriebenen Form dieses Dokuments muss als Version "8.0" angegeben werden.

Die generische Import-Schnittstelle erlaubt auch das Anlegen einer beliebig geschachtelten Ordnerhierarchie. Die in der Import-Datei angegebene Ordnerhierarchie bezieht sich relativ auf den Ordner, dessen Kontextmenü Sie zur Ausführung des Imports ausgewählt haben. Bei Verwendung des Menüs "Automatische Modellerstellung" ist das der Systemordner "Diagramme". Enthält die Import-Datei keine FOLDER-Elemente, werden die Vorlagen direkt im selektierten Ordner angelegt.

Element	Attribut	Beschreibung
FOLDER		*Generische Import-Schnittstelle*
	NAME	Der Name eines anzulegenden Ordners. Eine tiefere Ordnerhierarchie ergibt sich aus der Verschachtelung mehrerer FOLDER-Tags.

Sie können wahlweise einzelne Vorlagen instanziieren (TEMPLATE) oder mehrere Vorlagen in einem Diagramm zusammenfassen (BUILDUP). Im Template ist als Attribut der Name der Vorlage (NAME) und der Name des anzulegenden Diagramms anzugeben (INSTANCE):

Element	Attribut	Beschreibung
TEMPLATE		*Vorlage*
	NAME	Name der Vorlage
	INSTANCE	Name des anzulegenden Diagramms

Wenn Sie mehrere Vorlagen zu einem Diagramm zusammenfassen wollen, müssen Sie die Vorlagen-Elemente durch ein Element BUILDUP umschließen. In diesem Fall ist der Name des anzulegenden Diagramms im Element BUILDUP und nicht bei jedem einzelnen Vorlagen-Element anzugeben:

Element	Attribut	Beschreibung
BUILDUP		*Gruppierte Vorlagen*
	INSTANCE	Name des anzulegenden Diagramms
	ALIGNMENT	HOR: Ausrichtung horizontal (nebeneinander) VER: Ausrichtung vertikal (untereinander)
	WIDTH	Maximale Breite des Diagramms in Pixeln (bei horizontaler Ausrichtung)
	HEIGHT	Maximale Höhe des Diagramms in Pixeln (bei vertikaler Ausrichtung)

Innerhalb eines Vorlagen-Elementes können beliebig viele Ersetzungen für Platzhalter mit dem Element SUBST spezifiziert werden

Element	Attribut	Beschreibung
SUBST		*Ersetzung*
	NAME	Name des Platzhalters
	VALUE	Zu ersetzender Wert

Diagramme können auch ohne die Verwendung von Vorlagen erzeugt werden. Verwenden Sie dazu das Element DIAGRAM:

Element	Attribut	Beschreibung
DIAGRAM		*Diagramm*
	NAME	Der Name des anzulegenden Diagramms
	WIDTH	Die Breite des anzulegenden Diagramms in Pixeln
	HEIGHT	Die Höhe des anzulegenden Diagramms in Pixeln

Komponenten auf einem Diagramm werden mit dem Element COMP beschrieben:

Element	Attribut	Beschreibung
COMP		*Komponente*
	UID	Die eindeutige UID des Komponententyps
	NAME	Der Instanzname der Komponente
	REF	Ein frei wählbarer Referenzname der Komponente
	CYCLE	Die Zeitscheibe, der diese Komponente zugeordnet werden soll (1 .. 8).

Der Typ der Komponente wird durch die eindeutige UID identifiziert. Zur Instanziierung des Komponententyps muss der Komponententyp in einer Bibliothek, d. h. in der Task-Card "Komponenten" vorhanden sein.

Geben Sie weiter an, welchen Namen (NAME) die Komponente auf dem Diagramm erhalten soll. Wenn dieser Name zum Zeitpunkt des Imports im Projekt schon existiert, wird er von SIMIT automatisch durch Anhängen einer fortlaufenden Nummer eindeutig gemacht.

Ein Referenzname (REF) muss nur angegeben werden, wenn der Name der Komponente innerhalb der Import-Datei nicht eindeutig ist und eine Verbindung zu dieser Komponente spezifiziert werden soll. Dieser Fall kann beispielsweise bei Konnektoren auftreten, die mehrfach mit gleichem Namen auftreten.

Die Zeitscheibe, der diese Komponente zugeordnet werden soll, kann mit dem Element CYCLE angegeben werden. Wenn dieses Attribut fehlt, wird die Zeitscheibe 2 eingetragen.

Verknüpfungen zu einer existierenden Komponente werden mit dem Element LINK beschrieben:

Element	Attribut	Beschreibung
LINK		*Komponente*
	UID	Die eindeutige UID des Komponententyps
	NAME	Der Instanzname der Komponente

Die Position einer Komponente oder einer Verknüpfung wird durch das Element POS und seine Attribute X und Y spezifiziert. Dabei wird die linke obere Ecke der Komponente

angegeben. Der Nullpunkt des Diagramms befindet sich in der linken oberen Ecke des Diagramms. Positionen müssen positiv sein und innerhalb der Dimensionen des Diagramms liegen:

Element	Attribut	Beschreibung
POS		*Position*
	X	Die X-Position der Komponente in Pixeln
	Y	Die Y-Position der Komponente in Pixeln

Eine Komponente kann mit dem Element TRANSFORM auch skaliert und gedreht werden:

Element	Attribut	Beschreibung
TRANSFORM		*Transformation (Skalierung, Drehung)*
	SCALEX	Skalierung in X-Richtung (Default: 1).
	SCALEY	Skalierung in Y-Richtung (Default: 1).
	ROTATION	Angabe des Drehwinkels in Grad, um den die Komponente gegen den Uhrzeigersinn gedreht werden soll. Der Drehmittelpunkt ist die geometrische Mitte der Komponente.

Die Parameter einer Komponente oder einer Verknüpfung können mit dem Element PROP und seinen Attributen NAME und VALUE spezifiziert werden:

Element	Attribut	Beschreibung
PROP		*Parameter*
	NAME	Name des Parameters
	VALUE	Wert des Parameters

Anschlüsse von Komponente können miteinander verbunden werden. Zum einen geschieht das automatisch, wenn sie auf dem Diagramm übereinander liegen und den gleichen Verbindungstyp besitzen, anderenfalls muss die Verbindung in der Import-Datei spezifiziert werden. Spezifizieren Sie dazu zunächst den Namen des Anschlusses mit dem Element PORT:

Element	Attribut	Beschreibung
PORT		*Anschluss der Komponente*
	NAME	Name des Anschlusses

Die Verbindung wird dann durch das Element CONNECTION beschrieben:

Element	Attribut	Beschreibung
CONNECTION		*Verbindung*
	TYPE	LINE: Verbindung mit Signallinie IMPLICIT: Implizite Verbindung
	SOURCE	Der Name der Komponente, mit der eine Verbindung hergestellt werden soll. Wenn die Komponente das Attribut REF besitzt, muss hier dessen Eintrag benutzt werden.
	NAME	Der Name des Anschlusses, mit dem die Verbindung hergestellt werden soll

Die Verbindung von gerichteten Signalen ist immer vom Eingang her zu definieren, da der Eingang nur mit genau einem Ausgang einer anderen Komponente verschaltet werden kann. Der in der Verbindungsdefinition anzugebende Anschluss ist daher immer der Ausgang einer Komponente. Da topologische Verbindungen richtungslos sind, spielt es bei dieser Art von Verbindung keine Rolle, welcher der beiden zu verbindenden Anschlüsse als Anschluss in der Verbindungsdefinition angegeben wird.

Im Attribut SOURCE steht der Name der zu verbindenden Komponente bzw. Kopplung. Entsprechend steht im Attribut NAME der Name des Anschlusses bzw. Signals. Bei Verbindungen vom Typ IMPLICIT muss im Attribut SOURCE der Name der Komponente angegeben sein, Referenznamen (REF) sind hier bei impliziten Verbindungen nicht zulässig.

Vorbelegungen eines Anschlusses werden mit dem Element DEFAULT und seinen Attributen NAME und VALUE angegeben.

Element	Attribut	Beschreibung
DEFAULT		*Vorbelegung*
	NAME	Name des Anschlusses
	VALUE	Vorbelegung

Sind keine Vorbelegungen angegeben, dann gilt für die Anschlüsse, dass die Werte, wie sie im Komponententyp festgelegt sind.

Entsprechend den üblichen XML-Konventionen sind in der Import-Datei auch Kommentare erlaubt:

```
<!-- Kommentar -->
```

Siehe auch

Diagramme aus ZIP- oder XML-Dateien importieren (Seite 270)

4.3.3 Beispiele für XML-Dateien

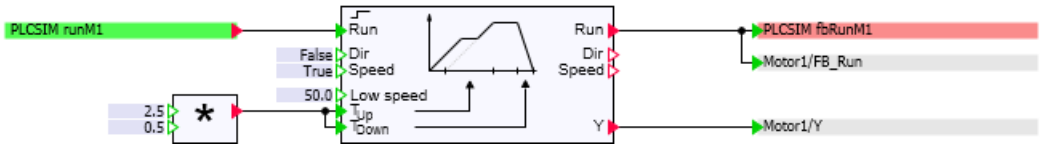
4.3.3.1 Beispiel für eine einfache Vorlageninstanziierung

Sie finden dieses Beispiel im Verzeichnis "_Beispiele/XML" auf der Installations-CD von SIMIT.

XML-Datei:

```
<GENERIC VERSION="7.1">
    <TEMPLATE NAME="MOTOR" INSTANCE="Motor1">
        <SUBST NAME="GATEWAY" VALUE="PLCSIM"/>
        <SUBST NAME="Plan_ChName" VALUE="Motor1"/>
        <SUBST NAME="feedback run_SymbolName" VALUE="fbRunM1"/>
        <SUBST NAME="monitoring time on_Value" VALUE="2.5"/>
        <SUBST NAME="output_SymbolName" VALUE="runM1"/>
    </TEMPLATE>
</GENERIC>
```

Diagramm:



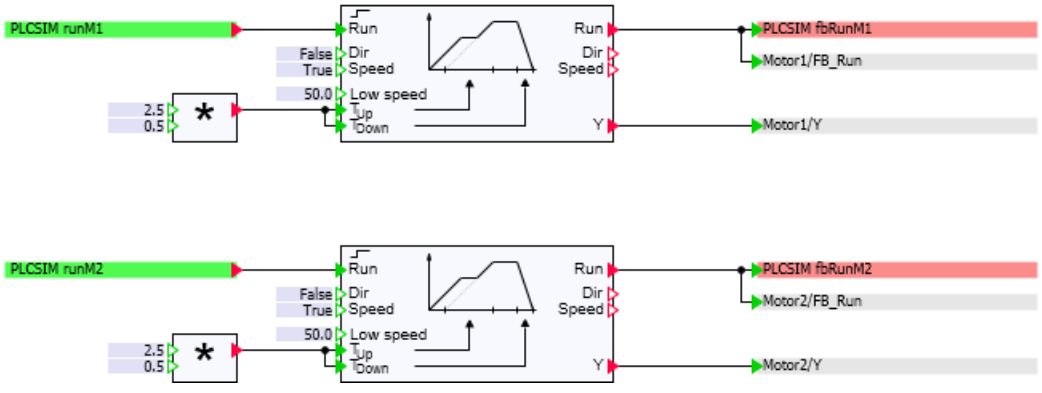
4.3.3.2 Beispiel für eine gruppierte Vorlageninstanziierung

Sie finden dieses Beispiel im Verzeichnis "_Beispiele/XML" auf der Installations-CD von SIMIT.

XML-Datei:

```
<GENERIC VERSION="7.1">
    <BUILDUP  INSTANCE="Motor1" DIR="VER" HEIGHT="1500">
        <TEMPLATE NAME="MOTOR">
            <SUBST NAME="GATEWAY" VALUE="PLCSIM"/>
            <SUBST NAME="Plan_ChName" VALUE="Motor1"/>
            <SUBST NAME="feedback run_SymbolName" VALUE="fbRunM1"/>
            <SUBST NAME="monitoring time on_Value" VALUE="2.5"/>
            <SUBST NAME="output_SymbolName" VALUE="runM1"/>
        </TEMPLATE>
        <TEMPLATE NAME="MOTOR">
            <SUBST NAME="GATEWAY" VALUE="PLCSIM"/>
            <SUBST NAME="Plan_ChName" VALUE="Motor2"/>
            <SUBST NAME="feedback run_SymbolName" VALUE="fbRunM2"/>
            <SUBST NAME="monitoring time on_Value" VALUE="2.5"/>
            <SUBST NAME="output_SymbolName" VALUE="runM2"/>
        </TEMPLATE>
    </BUILDUP>
</GENERIC>
```

Diagramm:



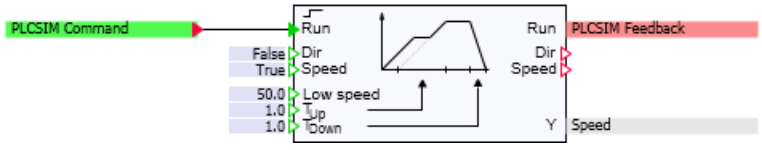
4.3.3.3 Beispiel für eine Diagrammerstellung

Sie finden dieses Beispiel im Verzeichnis "_Beispiele/XML" auf der Installations-CD von SIMIT.

XML-Datei:

```
<GENERIC VERSION="7.1">
    <FOLDER NAME="Motors">
        <DIAGRAM NAME="Diag1">
            <COMP ID="f_000hsn_20cyz1u8" NAME="Drive1">
                <POS X="230" Y="100"/>
                <PORT NAME="Run">
                    <CONNECTION SOURCE="PLCSIM/Command" NAME="Y"/>
                </PORT>
            </COMP>
            <COMP ID="f_000hsn_1zisln8r" NAME="PLCSIM/Feedback">
                <POS X="400" Y="110"/>
                <TRANSFORM SCALEX="2"/>
            </COMP>
            <COMP ID="f_000hsn_1zislnd3" NAME="PLCSIM/Command">
                <POS X="50" Y="110"/>
                <TRANSFORM SCALEX="2"/>
            </COMP>
            <COMP ID="f_000hsn_21b4dv0t" NAME="Speed">
                <POS X="400" Y="170"/>
            </COMP>
        </DIAGRAM>
    </FOLDER>
</GENERIC>
```

Diagramm:



4.3.4 Aufbau der Steuerdatei "procedure.cfg" für Import aus ZIP-Datei

Einleitung

Im ZIP-Archiv muss auf oberster Ebene die Steuerdatei "procedure.cfg" enthalten sein. Über diese Steuerdatei definieren Sie, wo die importierten Komponenten in SIMIT abgelegt werden.

Aufbau der Steuerdatei "procedure.cfg"

[RETRIEVE_COMPONENTS]

- source=<ZIP-Datei>
  Erforderlich. Legt die ZIP-Datei fest, in der die Komponenten enthalten sind. Unterstützt werden nur unverschlüsselte ZIP-Dateien.
- area=<user | project>
  Optional. Legt fest, ob die Komponenten nach dem Entpacken in der Taskcard "Komponenten" in "Eigene Komponenten" oder "Projektkomponenten" abgelegt werden. Wenn Sie diese Angabe weglassen, werden die Komponenten unter "Projektkomponenten" abgelegt.
- destination=<Ordnername>
  Optional. Legt den Ordner unterhalb von "Eigene Komponenten" oder "Projektkomponenten" fest, in dem die Komponenten gespeichert werden. Wenn Sie diese Angabe weglassen, werden die Komponenten im obersten Ordner gespeichert.

[COPY_COMPONENTS]

- source=<Ordnername>
  Erforderlich. Legt den Ordner im ZIP-Archiv fest, in dem die Komponenten im Format "*.simcmp" enthalten sind. Unterordner werden berücksichtigt.
- area=<user | project>
  Optional. Legt fest, ob die Komponenten in der Taskcard "Komponenten" in "Eigene Komponenten" oder "Projektkomponenten" importiert werden. Wenn Sie diese Angabe weglassen, wird in "Projektkomponenten" importiert.
- destination=<Ordnername>
  Optional. Legt den Ordner unterhalb von "Eigene Komponenten" oder "Projektkomponenten" fest, in den importiert wird. Wenn Sie diese Angabe weglassen, wird in den obersten Ordner importiert.

[GENERIC_IMPORT]

- file=<XML-Datei>
  Erforderlich. XML-Datei, welche die Importinformationen enthält.
- destination=<Ordnername>
  Optional. Legt den Ordner unter "Projektnavigation > Diagramme" fest, in den importiert wird. Wenn Sie diese Angabe weglassen, wird direkt in "Diagramme" importiert.
- RemoveEmptyElementsWithEmptyReplacement=<True | False>
  Optional. Entfernt Komponenten, die leere Ersetzungen beinhalten. Voreinstellung ist "False".

Siehe auch

Automatische Modellerstellung > Automatisierter Import (Seite 958)

Diagramme aus ZIP- oder XML-Dateien importieren (Seite 270)

4.4 Massenbearbeitung

4.4.1 Daten exportieren

Einleitung

Mit der Massenbearbeitung bearbeiten Sie Parameter eines Projekts in Tabellenform, ohne einzelne Pläne und Komponenten zu öffnen. Parameter und Eingangsvorbelegungen exportieren Sie in eine Tabellenverarbeitungsdatei (*.xlsx), bearbeiten die Werte und importieren anschließend den Inhalt der Tabellenverarbeitungsdatei wieder in das Projekt. Für strukturelle Änderungen am Simulationsmodell verwenden Sie die Mechanismen zur automatischen Modellerstellung.

Voraussetzung

- SIMIT-Projekt ist geöffnet.
- Projektansicht wird angezeigt.

Vorgehen

Um Daten zu exportieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation im Kontextmenü von "Diagramme" den Befehl "Massenbearbeitung Export".
   Der Befehl steht bei den Elementen Projekt, Diagramme und allen Elementen unterhalb von Diagrammen zur Verfügung.
   Wenn Sie den Export auf Elementen unterhalb der Ebene "Diagramme" ausführen, werden selektiv nur die Parameter und Eingangsvorbelegungen der Komponenten ab dieser gewählten Ebene exportiert.
2. Geben Sie einen Namen für die Export-Datei ein.

Ergebnis

Die Export-Datei wird erzeugt und im angegebenen Verzeichnis gespeichert.

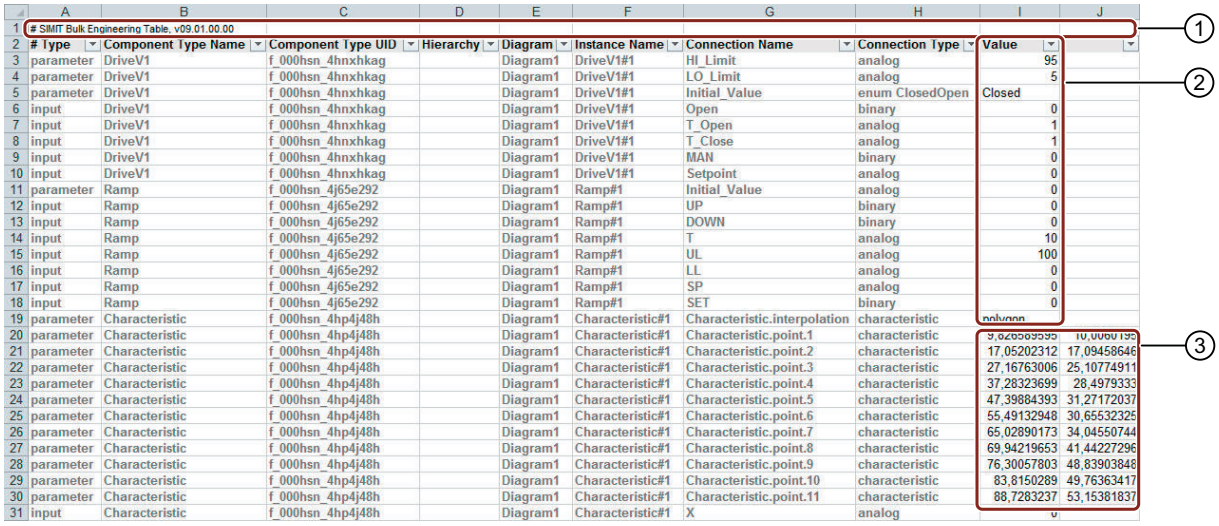
Siehe auch

Automatische Modellerstellung (Seite 251)

4.4.2 Inhalt der Exportdatei

Aufbau der Export-Datei

Die folgende Abbildung zeigt den Aufbau der Export-Datei:



	A	B	C	D	E	F	G	H	I	J
1	# SIMIT Bulk Engineering Table, v09.01.00.00									
2	# Type	Component Type Name	Component Type UID	Hierarchy	Diagram	Instance Name	Connection Name	Connection Type	Value	
3	parameter	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	HI_Limit	analog	95	
4	parameter	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	LO_Limit	analog	5	
5	parameter	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	Initial_Value	enum ClosedOpen	Closed	
6	input	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	Open	binary	0	
7	input	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	T_Open	analog	1	
8	input	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	T_Close	analog	1	
9	input	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	MAN	binary	0	
10	input	DriveV1	f_000hsn_4hnxhkag		Diagram1	DriveV1#1	Setpoint	analog	0	
11	parameter	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	Initial_Value	analog	0	
12	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	UP	binary	0	
13	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	DOWN	binary	0	
14	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	T	analog	10	
15	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	UL	analog	100	
16	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	LL	analog	0	
17	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	SP	analog	0	
18	input	Ramp	f_000hsn_4j65e292		Diagram1	Ramp#1	SET	binary	0	
19	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.interpolation	characteristic	polygon	
20	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.1	characteristic	3,826869595	10,0060196
21	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.2	characteristic	17,05202312	17,09458646
22	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.3	characteristic	27,16763006	25,10774911
23	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.4	characteristic	37,28323699	28,4979333
24	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.5	characteristic	47,39884393	31,27172037
25	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.6	characteristic	55,49132948	30,65532325
26	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.7	characteristic	65,02890173	34,04550744
27	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.8	characteristic	69,94219653	41,44227296
28	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.9	characteristic	76,30057803	48,83903848
29	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.10	characteristic	83,8150289	49,76363417
30	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.11	characteristic	88,7283237	53,15381837
31	input	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	X	analog	0	

① SIMIT-Kennung für den Import.
Verändern Sie diese Zeile nicht.

② Parameterwerte und Vorbelegungen von offenen Eingängen

③ Parameterpaare von Kennlinien

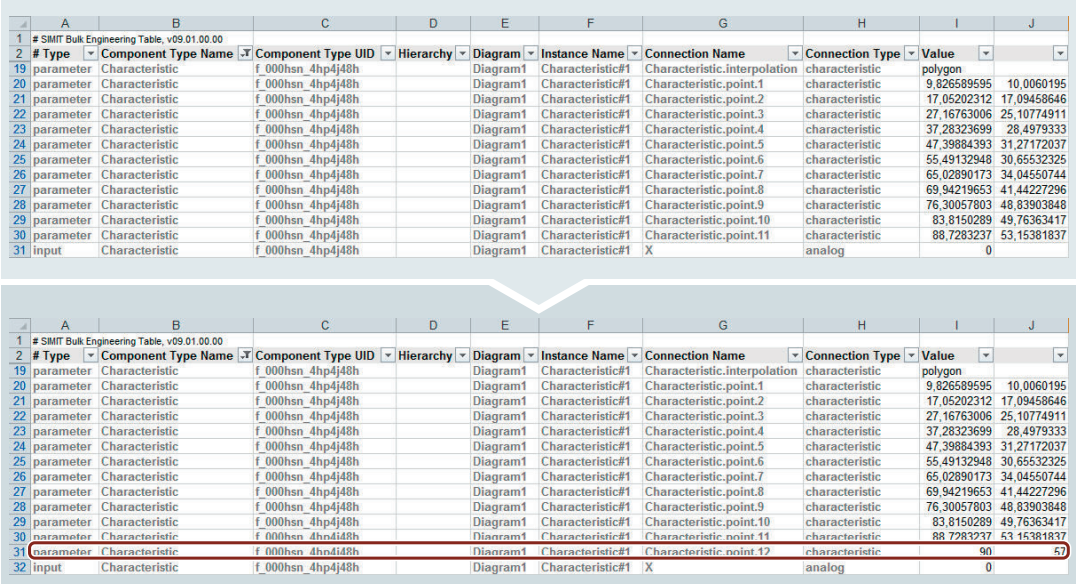
Regeln zum Arbeiten in der Export-Datei

Beachten Sie folgende Regeln:

- Ändern Sie ausschließlich Werte in der Spalte "Value".
- Beachten Sie beim Ändern von Werten den zugehörigen Datentyp in Spalte "Connection_Type".
- Ändern Sie weder Anzahl noch Reihenfolge der Spalten.
- Fügen Sie keine Zeilen hinzu. Ausnahme: "Hinzufügen von Parameterpaaren".

Hinzufügen von Parameterpaaren

In der Export-Datei können Sie Parameterpaare von Kennlinien ergänzen. Unterstützt wird das Fortsetzen einer Reihe von Parameterpaaren:



	A	B	C	D	E	F	G	H	I	J
1	# SIMIT Bulk Engineering Table, v09.01.00.00									
2	# Type	Component Type Name	Component Type UID	Hierarchy	Diagram	Instance Name	Connection Name	Connection Type	Value	
19	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.interpolation	characteristic	polygon	
20	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.1	characteristic	9,826589595	10,0060195
21	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.2	characteristic	17,05202312	17,09458646
22	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.3	characteristic	27,16763006	25,10774911
23	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.4	characteristic	37,28323699	28,4979333
24	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.5	characteristic	47,39884393	31,27172037
25	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.6	characteristic	55,49132948	30,65532325
26	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.7	characteristic	65,02890173	34,04550744
27	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.8	characteristic	69,94219653	41,44227296
28	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.9	characteristic	76,30057803	48,83903848
29	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.10	characteristic	83,8150289	49,76363417
30	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.11	characteristic	88,7283237	53,15381837
31	input	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	X	analog	0	

	A	B	C	D	E	F	G	H	I	J
1	# SIMIT Bulk Engineering Table, v09.01.00.00									
2	# Type	Component Type Name	Component Type UID	Hierarchy	Diagram	Instance Name	Connection Name	Connection Type	Value	
19	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.interpolation	characteristic	polygon	
20	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.1	characteristic	9,826589595	10,0060195
21	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.2	characteristic	17,05202312	17,09458646
22	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.3	characteristic	27,16763006	25,10774911
23	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.4	characteristic	37,28323699	28,4979333
24	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.5	characteristic	47,39884393	31,27172037
25	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.6	characteristic	55,49132948	30,65532325
26	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.7	characteristic	65,02890173	34,04550744
27	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.8	characteristic	69,94219653	41,44227296
28	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.9	characteristic	76,30057803	48,83903848
29	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.10	characteristic	83,8150289	49,76363417
30	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.11	characteristic	88,7283237	53,15381837
31	parameter	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	Characteristic.point.12	characteristic	90	57
32	input	Characteristic	f_000hsn_4hp4j48h		Diagram1	Characteristic#1	X	analog	0	

4.4.3 Daten importieren

Einleitung

Mit der Funktion "Massenbearbeitung Import" importieren Sie folgende Daten:

- Mit der Funktion "Massenbearbeitung Export" exportierte und bearbeitete Daten
- Geänderte Daten aus dem letzten Simulationslauf. Die Änderungen werden beim Beenden der Simulation in einer Datei gespeichert.

Voraussetzung

- SIMIT-Projekt ist geöffnet.
- Eine für den Massenimport geeignete Datei liegt vor.

Vorgehen

Um Daten zu importieren, gehen Sie folgendermaßen vor:

1. Wählen Sie im Menü "Automatische Modellerstellung" den Befehl "Massenbearbeitung Import".
   Der Dialog "Massenbearbeitung Import" wird geöffnet.
2. Wenn Sie Daten aus dem letzten Simulationslauf importieren wollen:
   - Aktivieren Sie die Option "Änderungen der letzten Simulation übernehmen".
3. Wenn Sie mit der Funktion "Massenbearbeitung Export" exportierte Daten importieren wollen:
   - Deaktivieren Sie die Option "Änderungen der letzten Simulation übernehmen".
   - Wählen Sie die Datei, aus der Sie die geänderte Werte importieren wollen.
4. Öffnen Sie bei Bedarf die Vorschau und wählen Sie die Elemente aus, die Sie importieren wollen.
5. Starten Sie den Import.

Ergebnis

Geänderte Werte werden in das SIMIT-Projekt importiert. Wenn ein Parameter außerhalb eines definierten zulässigen Bereichs liegt, wird der jeweilige maximale oder minimale Wert importiert. Fehlerhafte Werte werden nicht importiert.

Siehe auch

Dialogfeld "Massenbearbeitung Import" (Seite 976)

Diagnose & Visualisierung 5

5.1 Trend and Messaging Editor

5.1.1 Funktionen des Trend and Messaging Editors

Der Trend and Messaging Editor unterstützt die Analyse der Abläufe und Signalwerte in der Simulation mit folgenden Funktionen:

- **Neues Kurvenbild**
  Mit einem Kurvenbild werden Signalverläufe grafisch dargestellt und können auch ausgedruckt werden. Weitere Informationen hierzu finden Sie im Kapitel: Kurvenbilder (Seite 290).
- **Meldungen**
  Mit dem Meldesystem werden Meldungen zu Ereignissen in der Simulation erzeugt. Die Meldungen können im Meldungseditor ausgewertet werden. Weitere Informationen hierzu finden Sie im Kapitel: Meldesystem (Seite 285).
- **Archiv**
  Im Archiv können Sie Signale eines Simulationslaufs aufzeichnen. Weitere Informationen hierzu finden Sie im Kapitel: Archiv (Seite 288).

Die Funktionen befinden sich in der Projektnavigation im Ordner "Monitoring".

5.1.2 Meldesystem

5.1.2.1 Meldungsklassen

Meldungen dienen dazu, über das Auftreten bestimmter Ereignisse in der Simulation zu informieren. Sie können nach dem auslösenden Ereignis wie folgt klassifiziert werden:

- **Meldungen der Meldekomponente**
  Hier werden Meldungen von einer speziellen Meldekomponente erzeugt. Diese Komponente steht in der Standardbibliothek zur Verfügung und kann auf Diagrammen mit Binärsignalen verknüpft werden. Meldekategorie und -text können in jeder Meldekomponente frei parametriert werden.
- **Meldungen sonstiger Komponenten**
  Diese Meldungen sind in den Komponententypen definiert und haben eine feste, im Komponententyp hinterlegte Meldekategorie und Meldetext.
- **Systemmeldungen**
  Systemmeldungen werden von SIMIT selbst erzeugt und informieren Sie im Fehlerfall über unerwartete Ereignisse wie beispielsweise Probleme beim Verbindungsaufbau einer Kopplung. Systemmeldungen von SIMIT sind immer von der Kategorie *SYSTEM*.

Alle Meldungen, die nach dem Start der Simulation erzeugt werden, werden in SIMIT gepuffert und stehen im Meldungseditor zur Verfügung. Die vorhandenen Meldungen werden mit dem erneuten Starten der Simulation verworfen.

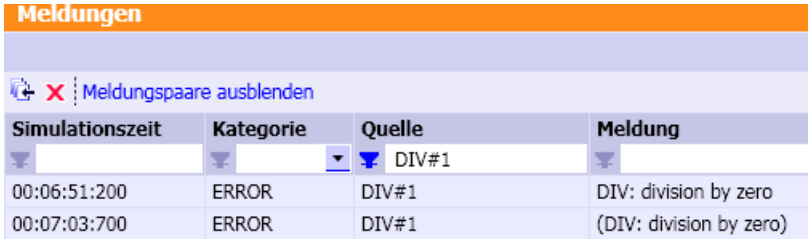
5.1.2.2 Meldungseditor

Öffnen Sie den Meldungseditor in der Projektnavigation unter "Monitoring > Meldungen" mit einem Doppelklick. Die Meldungen werden mit folgenden Informationen aufgelistet:

- Simulationszeit im Format "hh:mm:ss:msec"
- Kategorie
- Name der Komponente, die die Meldung ausgelöst hat (Quelle)
- Meldetext (Meldung)

Der Meldungseditor ist sowohl bei laufender wie auch bei gestoppter Simulation verfügbar. Bei laufender Simulation werden die Meldungen hinzugefügt, sobald sie erzeugt worden sind. Nach dem Stoppen der Simulation werden alle im vorhergehenden Simulationslauf erzeugten Meldungen aufgelistet.

Die angezeigten Meldungen können nach verschiedenen Kriterien gefiltert werden. In der folgenden Abbildung ist z. B. ein Filter auf Meldungen mit der Quelle *DIV#1* eingestellt:



5.1.2.3 Meldungen bearbeiten

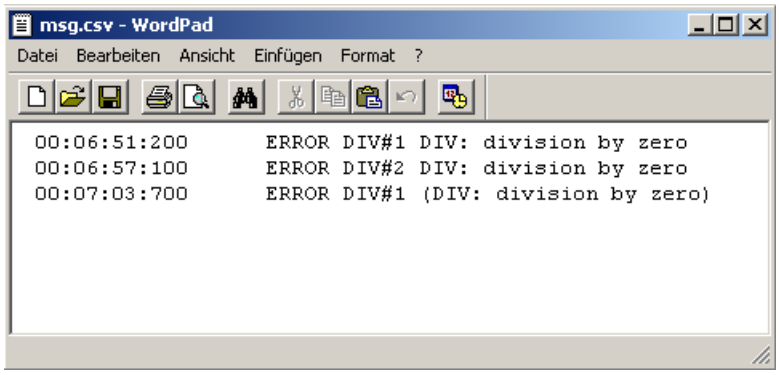
Meldungen exportieren

Exportieren Sie die Meldungen, indem Sie auf das Symbol "" im Meldungseditor klicken. Der Export erfolgt in eine Textdatei (*.txt). Die Textdatei enthält 4 Spalten:

- Simulationszeit
- Kategorie
- Name der auslösenden Komponente
- Meldetext.

Die Spalten sind mit dem Tabulator-Zeichen getrennt.

Mit der Textdatei archivieren Sie die Meldungen oder bearbeiten sie mit einem Texteditor.



Meldungen löschen

Löschen Sie alle im Meldungseditor vorhandenen Meldungen, indem Sie auf das Symbol "x" klicken.

Meldungen ausblenden

Eine Meldekomponente erzeugt Meldungen, die von dem mit ihr verbundenen Binärsignal getriggert werden. Beim Wechsel des Signals von null nach eins wird eine Meldung als "Kommend"-Meldung erzeugt, beim Wechsel von eins nach null wird eine "Gehend"-Meldung erzeugt. "Gehend"-Meldungen werden mit eingeklammertem Meldetext dargestellt.

Ein Meldungspaar besteht aus einer "Kommend"-Meldung und einer späteren "Gehend"-Meldung zum selben Ereignis. Über die Schaltfläche "Meldungspaare ausblenden" können Sie Meldungspaare aus- und einblenden. Wenn Sie Meldungspaare ausblenden, bleiben nur solche Meldungen sichtbar, die gekommen, aber noch nicht wieder gegangen sind.

5.1.2.4 Message – die Meldekomponente

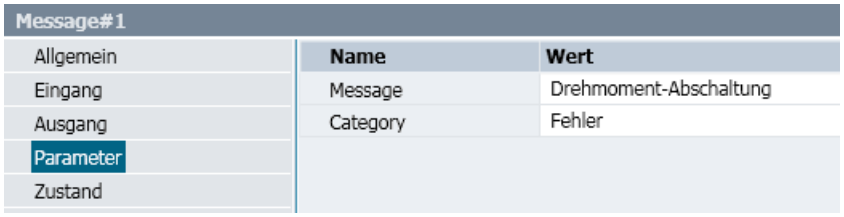
Symbol



Funktion

Der Komponententyp *Message* ist in der Standardbibliothek im Verzeichnis *Misc* zu finden. Eine Komponente dieses Typs erzeugt eine Meldung, wenn sich der binäre Wert an ihrem Eingang ändert. Der von der Komponente ausgegebene Meldetext wird bei der "Gehend"-Meldung in runde Klammern gesetzt.

Die Kategorie und der Meldetext werden als Parameter im Eigenschaftsfenster der Komponente eingetragen.



Hinweis

Meldungen der Kategorie SYSTEM sind Systemmeldungen von SIMIT, Meldungen der Kategorie ERROR sind Meldungen von Komponenten aus der Basisbibliothek. Vermeiden Sie bei der Wahl von Namen für die Kategorie die Namen SYSTEM und ERROR, damit Sie beim Filtern über Kategorien ausschließlich Meldungen der von Ihnen gesetzten Kategorien angezeigt bekommen.

5.1.2.5 Komponentenspezifische Meldungen

Einige Komponenten der Basisbibliothek erzeugen Meldungen, die auf unzulässige Operationen hinweisen (z. B. Division durch 0 oder unzulässige Parametrierung).

Eine Komponente des Typs SQRT gibt z. B. die Meldung "SQRT: invalid argument" aus, wenn der Eingangswert negativ geworden ist.

Meldungen von Komponenten der Basisbibliothek sind immer der Meldekategorie ERROR zugeordnet, und der Meldetext ist in der Form "Komponententyp: Fehlermeldung" gehalten. Alle Meldungen sind als "Kommend"- und "Gehend"-Meldung angelegt.

5.1.2.6 Meldungen in der Statusleiste

Die aktuelle Meldung wird unter Angabe der zugehörigen Simulationszeit und des Meldetextes rechts in der Statusleiste in der Projektansicht von SIMIT angezeigt:



5.1.2.7 Begrenzungen des Meldesystems

Das Meldesystem kann bis zu 1000 Meldungen pro Sekunde verarbeiten. Wird dieser Wert überschritten, gehen Meldungen verloren.
Die folgende Meldung wird dann angezeigt:

00:00:02:600	SYSTEM	ControlSystem Warning	TME : Message system overload

Maximal können 4096 Meldungen gespeichert werden. Wird diese Grenze erreicht, wird die folgende Meldung angezeigt:

00:00:09:900	SYSTEM	TME	Der Meldepuffer ist voll

5.1.3 Archiv

Das Archiv zeichnet die Signale eines Simulationslaufs auf, d. h. vom Starten bis zum Beenden einer Simulation. Mit dem nächsten Simulationslauf, d. h. mit dem nächsten Starten der Simulation, wird das Archiv gelöscht, und die zu archivierenden Signale werden erneut bis zum Beenden der Simulation aufgezeichnet.

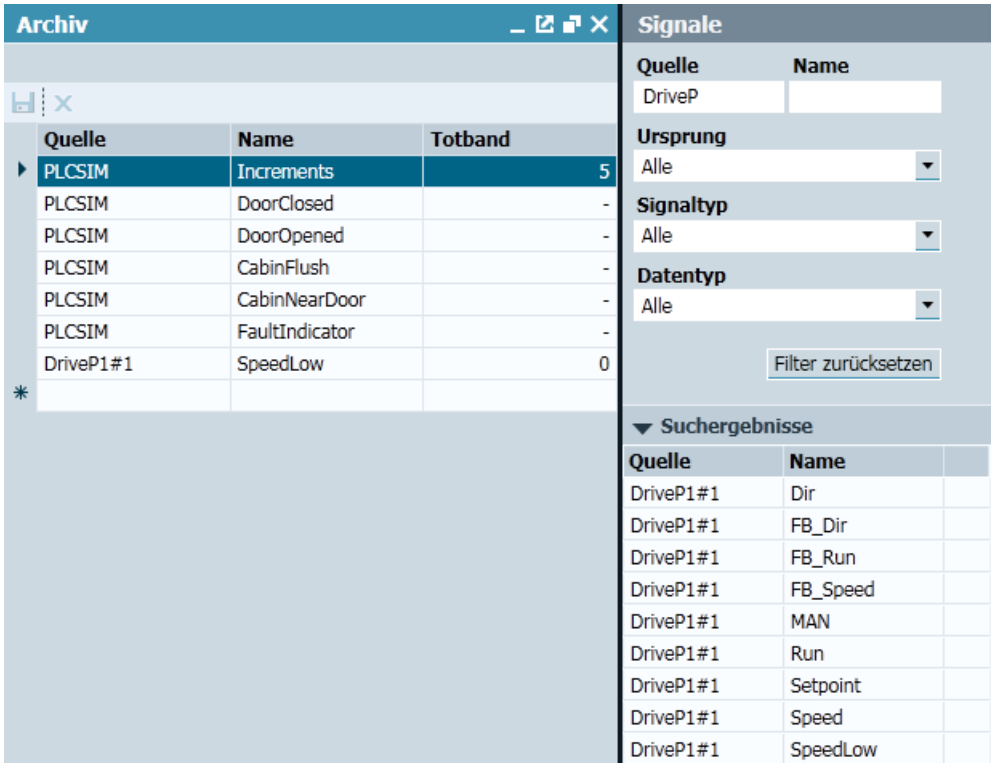
Folgende Signale werden im Archiv aufgezeichnet:

- Eingangs- und Ausgangssignale von Komponenten
- Zustände von Komponenten
- Eingangs- und Ausgangssignale der Kopplungen

Archivierte Signale können in Form von Kurvenverläufen dargestellt werden.

Das Archiv konfigurieren

Öffnen Sie das Archiv mit einem Doppelklick in der Projektnavigation unter "Monitoring > Archiv". Der Archiveditor wird geöffnet und die Signale werden angezeigt, die aktuell archiviert werden. Tragen Sie weitere Signale von Hand ein oder ziehen Sie sie aus der Task-Card "Signale".



Im Archiveditor wird ein Totband eingetragen, um die Datenmenge beim Aufzeichnen der Signale zu begrenzen. Das Totband gibt an, um wieviel sich ein Signalwert ändern muss, damit der Wert aufgezeichnet wird. Ein Signalwert wird nur aufgezeichnet, wenn sich der Wert gegenüber dem zuletzt aufgezeichneten Wert um mehr als das Totband unterscheidet. Das Totband kann nur für analoge oder ganzzahlige (Integer) Signale angegeben werden. Der Wert eines binären Signals wird bei jedem Signalwechsel aufgezeichnet.

Hinweis

Wenn Sie für das Totband "0" eintragen, dann wird für Analog- oder ganzzahlige Signale, die sich kontinuierlich ändern, in jedem Zyklus ein Wert aufgezeichnet.

Im Archiv können maximal 128 Signale eingetragen werden. Das Überschreiten dieser Grenze wird mit einer Fehlermeldung angezeigt.

Für jedes Signal wird Speicherplatz reserviert. Damit wird bei einem Grundzyklus von 100 ms und einem Totband von "0" ein Signal ca. 2,5 Stunden lang aufgezeichnet. Seltenere Signalwechsel führen zu einer längeren Aufzeichnung.

5.1.4 Kurvenbilder

5.1.4.1 Signaldarstellung mit Kurvenbildern

Kurvenbilder bieten die Möglichkeit, den zeitlichen Verlauf von Signalwerten grafisch darzustellen. Sie können folgende Signale in Kurvenbildern darstellen:

- Eingangs- und Ausgangssignale von Komponenten
- Zustände von Komponenten
- Eingangs- und Ausgangssignale der Kopplungen

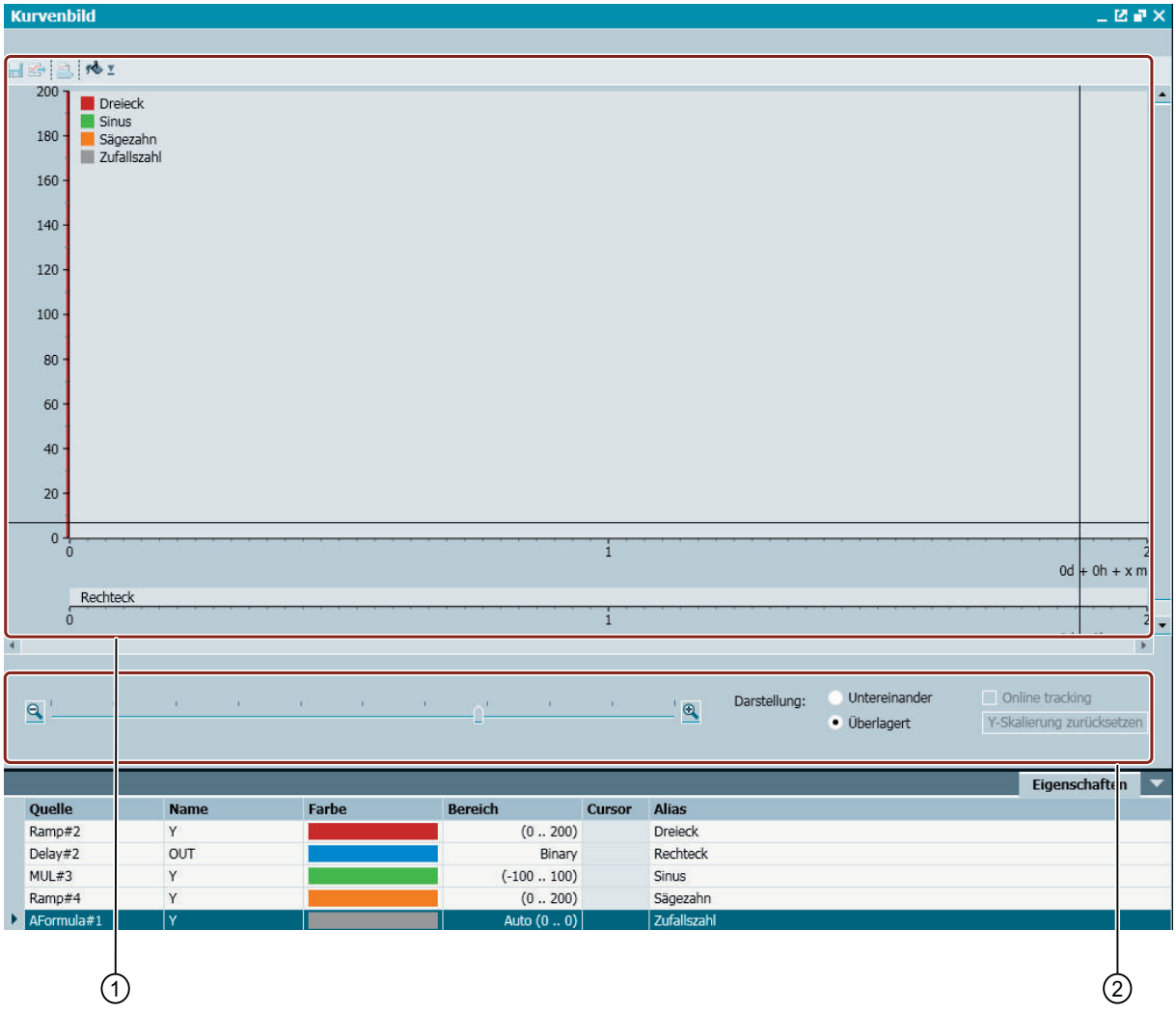
Die Signale können sowohl archiviert als auch nicht archiviert sein. Archivierte Signale können mit ihrem vollständigen, im Archiv aufgezeichneten Verlauf dargestellt werden.

Für nicht archivierte Signale wird der Verlauf nur so lange dargestellt, wie das Kurvenbild geöffnet ist. Für jedes nicht archivierte Signal wird im Kurvenbild Speicherplatz reserviert. Damit kann bei einem Grundzyklus von 100 ms ein Signal ca. 35 min lang aufgezeichnet werden, wenn sein Wert in jedem Zyklus aufgezeichnet wird. Seltenere Signalwechsel führen zu einer längeren Aufzeichnung.

5.1.4.2 Kurvenbilder anlegen und konfigurieren

Kurvenbilder werden in der Projektnavigation im Ordner "Monitoring" abgelegt. Legen Sie ein neues Kurvenbild an, indem Sie auf den Eintrag "Neues Kurvenbild" doppelklicken. In Ihrem Projekt können Sie beliebig viele Kurvenbilder anlegen. Sie können Kurvenbilder auch anlegen und konfigurieren, wenn Sie eine Simulation gestartet haben.

Kurvenbilder werden im Kurvenbildeditor im Arbeitsbereich geöffnet. Im oberen Bereich des Kurvenbildfensters werden die Kurvenverläufe der Signale dargestellt, im unteren Bereich des Fensters nehmen Sie Einstellungen für das Kurvenbild vor.



① Kurvenverläufe
② Einstellungen

Signale von Kurvenbildern konfigurieren

Im Eigenschaftsfenster des Kurvenbildeditors listen Sie die Signale auf, die das Kurvenbild darstellen soll. Sie können die Signale mit Quelle und Name von Hand in die Tabelle eintragen oder die Signale aus der Task-Card "Signale" in die Tabelle ziehen.

Eigenschaften

Quelle	Name	Farbe	Bereich	Cursor	Alias
Ramp#2	Y		(0 .. 200)		Dreieck
Delay#2	OUT		Binary		Rechteck
MUL#3	Y		(-100 .. 100)		Sinus
Ramp#4	Y		(0 .. 200)		Sägezahn
AFormula#1	Y		Auto (0 .. 0)		Zufallszahl

Ein Kurvenbild kann maximal 16 Signale enthalten.

Signale in Kurvenbildern parametrieren

Für jedes Signal können Sie folgende Parameter einstellen:

- Die Farbe, mit der der Signalverlauf dargestellt werden soll
- Den Wertebereich des Signals
- Einen Ersatznamen für das Signal

Analog- und Integersignale werden in einem bestimmten Wertebereich dargestellt. Voreingestellt ist ein automatischer Modus "Auto (0..0)", bei dem der Wertebereich entsprechend den tatsächlich auftretenden Werten ständig angepasst wird. Mit diesem Modus müssen Sie keinen individuellen Wertebereich angeben, und alle Signalverläufe werden in der jeweils maximalen Auflösung angezeigt.

Hinweis

Im automatischen Modus wird durch die ständig wechselnde Skalierung der Vergleich mehrerer unterschiedlicher Signalverläufe unter Umständen schwieriger.

Um feste Wertebereiche einzutragen, können Sie den Anfangs- und Endwert auch direkt mit zwei Punkten getrennt eintragen, z. B.: "0..2000".

Wenn Sie im Kurvenbild die Signale mit einem anderen als dem mit Quelle und Name gegebenen Signalnamen bezeichnen wollen, können Sie einen Ersatznamen (Alias) vergeben. Dieser Ersatzname wird für die Bezeichnung der Kurve verwendet.

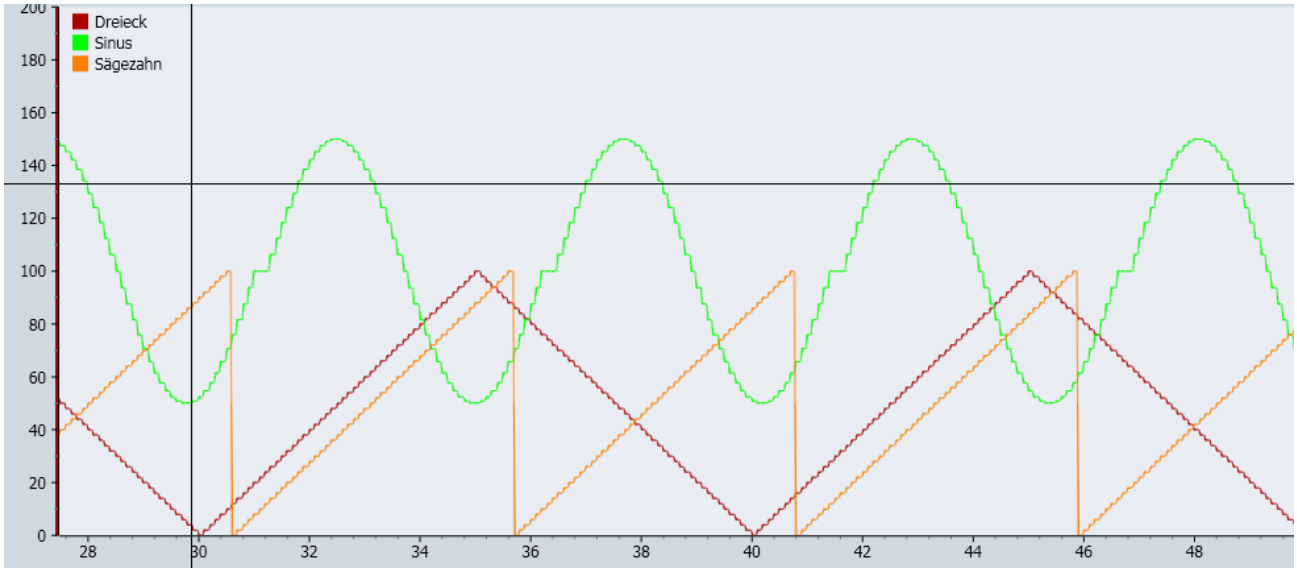
Hinweis

In einem Projekt können Sie beliebig viele Kurvenbilder anlegen, die Summe der Signale in allen geöffneten Kurvenbildern darf aber 128 nicht übersteigen.

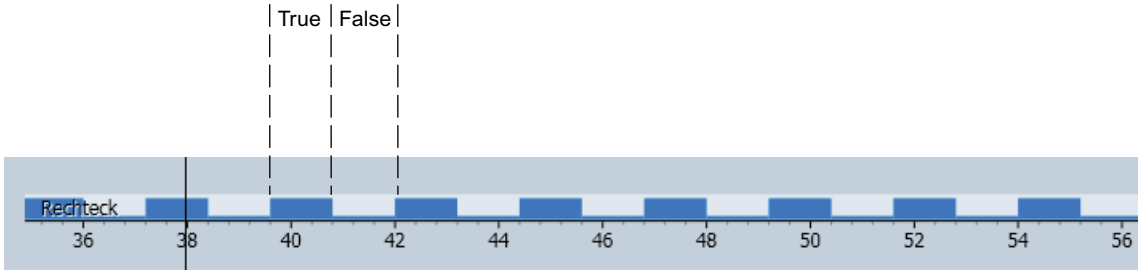
5.1.4.3 Kurvenverläufe anzeigen

Darstellung von Kurvenbildern nach Datentypen

Die Signalverläufe von Analog- und Integersignalen werden als Kurvendiagramme angezeigt. Dabei ist der Abszisse (X-Achse) die Simulationszeit zugeordnet, der Ordinate (Y-Achse) die Signalwerte.



Binärsignale werden unterhalb der Kurven der Analog- und Integersignale dargestellt. Da Binärsignale nur die beiden Werte null (False) und eins (True) annehmen, wird der Verlauf eines Binärsignals als Balkendiagramm über der Simulationszeit dargestellt. Eine Linie repräsentiert dabei den Nullwert (False) und ein Balken den Wert eins (True), wie in der folgenden Abbildung dargestellt:



Darstellung von archivierten Signalen

Archivierte Signale werden mit den archivierten Signalwerten dargestellt. Da die Aufzeichnung von Signalwerten im Archiv mit dem Start der Simulation beginnt, sehen Sie den Kurvenverlauf von archivierten Signalen immer für alle Zeitpunkte vom letzten Start der Simulation an, wenn die Speicherbeschränkung nicht überschritten wurde.

Nicht archivierte Signale werden in einem Kurvenbild nur aufgezeichnet, solange das Kurvenbild geöffnet ist. Sie sehen den Verlauf dieser Signale nur ab dem Zeitpunkt des Öffnens des Kurvenbildes. Beenden Sie eine laufende Simulation, dann sehen Sie den Signalverlauf

von nicht archivierten Signalen nur in den bereits zur Laufzeit der Simulation geöffneten Kurvenbildern. Mit dem Schließen eines Kurvenbildes werden die aufgezeichneten Signale gelöscht, so dass nach erneutem Öffnen des Kurvenbildes die Signalverläufe von nicht archivierten Signalen nicht mehr angezeigt werden können.

Zur Laufzeit der Simulation wird in den Signaleigenschaften auch der jeweils aktuelle Signalwert angezeigt.

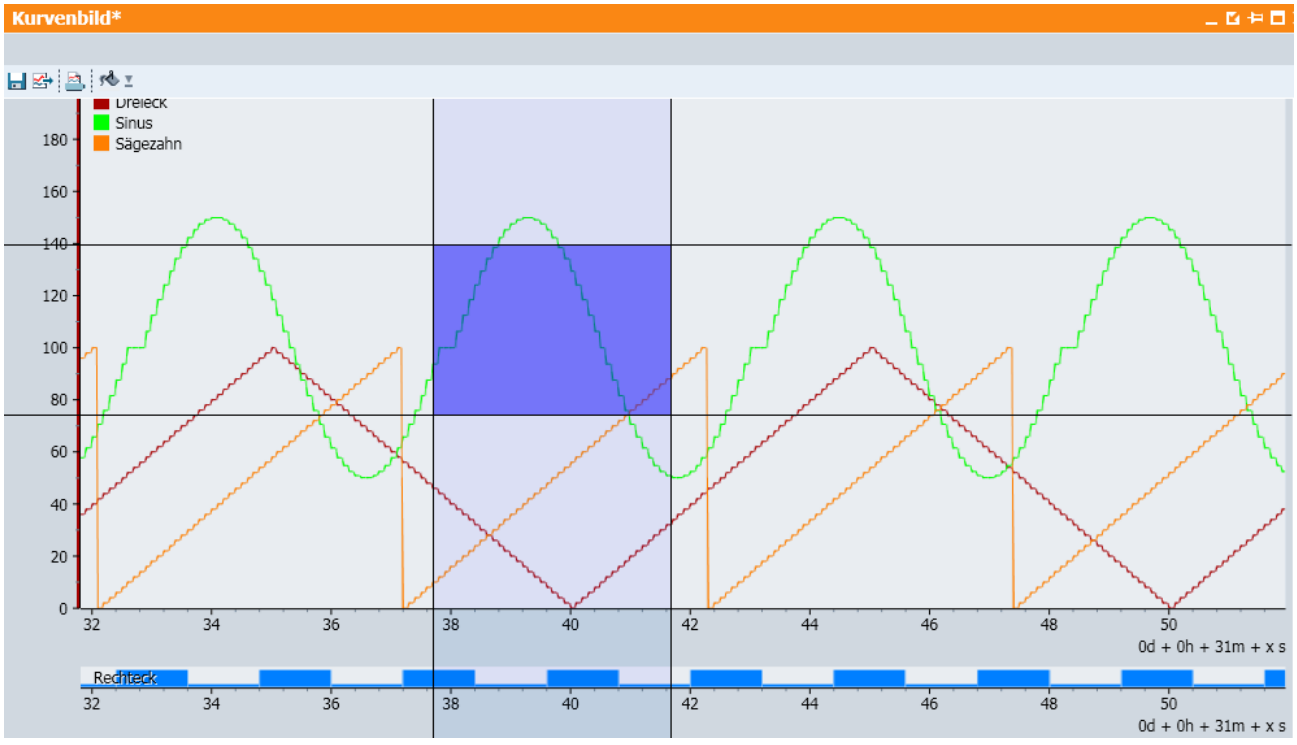
Eigenschaften

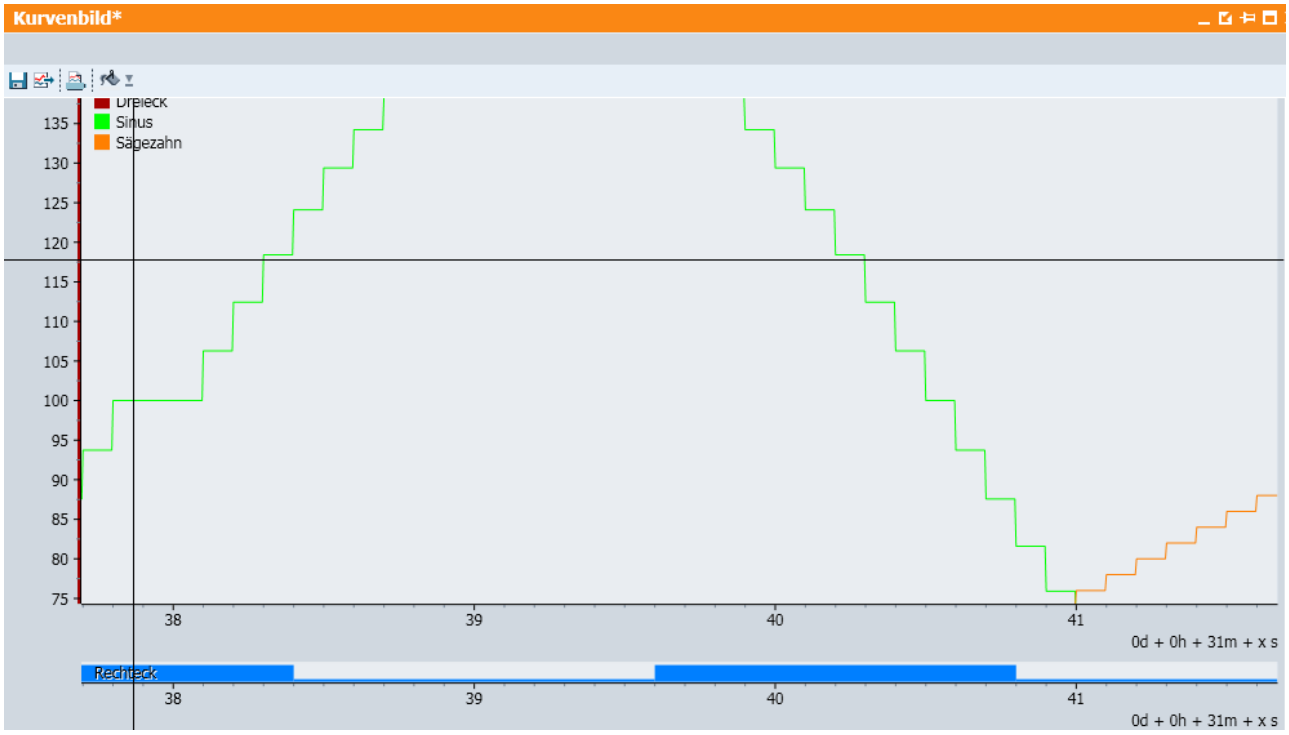
Quelle	Name	Farbe	Bereich	Totband	Aktuell	Alias	Archiviert
Ramp#2	Y		(0 .. 200)	0	4.0	Dreieck	☑
Delay#2	OUT		Binary	-	False	Rechteck	☑
MUL#3	Y		(-100 .. 100)	0	-45.2413526233011	Sinus	☑
Ramp#4	Y		(0 .. 200)	0	26.0	Sägezahn	☐
*							

Das Totband von archivierten Signalen wird im Kurvenbild bei der Darstellung der Signale entsprechend einberechnet. In den Eigenschaften der Signale werden zur Laufzeit der Simulation die Totbandwerte angezeigt. Für nicht archivierte Signale werden in Kurvenbildern alle in der Simulation berechneten Werte aufgezeichnet.

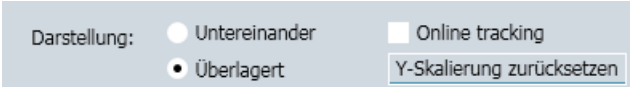
Ausschnitte des Kurvenverlaufs vergrößern

Wenn Sie die fortlaufende Aktualisierung im Kurvenbild ausschalten und den Mauszeiger in das Kurvendiagramm bewegen, wird ein Kreuz-Cursor eingeblendet. Um einen bestimmten Bereich zu vergrößern, markieren Sie mit gedrückter linker Maustaste den Bereich. Der markierte Bereich wird dann im Kurvendiagramm entsprechend vergrößert dargestellt.





Mit der Schaltfläche "Y-Skalierung zurücksetzen" setzen Sie die Y-Achse wieder auf die definierten Bereiche zurück.

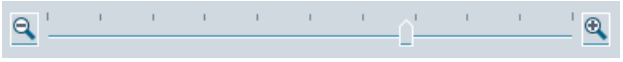


Weitere Informationen hierzu finden Sie im Kapitel: Einstellungen des Kurvenbildes (Seite 295).

Einstellungen des Kurvenbildes

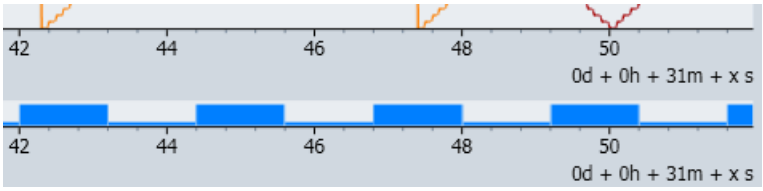
Zeitachse

Der Maßstab der Zeitachse ist über einen Schieber einstellbar:



Rechts unter der Zeitachse wird eingeblendet, bei welchem Zeitpunkt die Darstellung beginnt. Addieren Sie zu dieser Angabe die Tage (d), Stunden (h), Minuten (m) oder Sekunden (s) gemäß den entsprechenden Maßzahlen an der X-Achse.

Die folgende Abbildung zeigt z. B. die Zeitachse mit einem Ausschnitt von 31 Minuten 42 Sekunden bis 31 Minuten 50 Sekunden.



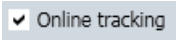
Hinweis

Aktualisierung der Zeitanzeige

Die Zeitachse bewegt sich von rechts nach links und ist mit den Sekunden beschriftet. Messpunkt für die Minutenzählung ist der linke Rand des Anzeigefensters. Sobald der Wert 60 [s] den Messpunkt passiert hat, wird der Minutenzähler um 1 erhöht und die Skalenwerte um 60 zurückgesetzt.

Fortlaufende Aktualisierung

Wenn Sie ein Kurvenbild öffnen und die Simulation starten, dann sehen Sie die Kurvenverläufe in ihrem ständig aktualisierten Verlauf. Die Kurvenverläufe werden den sich ändernden Signalen nachgeführt. Um ein statisches Bild der momentanen Kurvenverläufe zu bekommen, schalten Sie die Aktualisierung ab. Wählen Sie dazu die Option Aktualisierung ab:



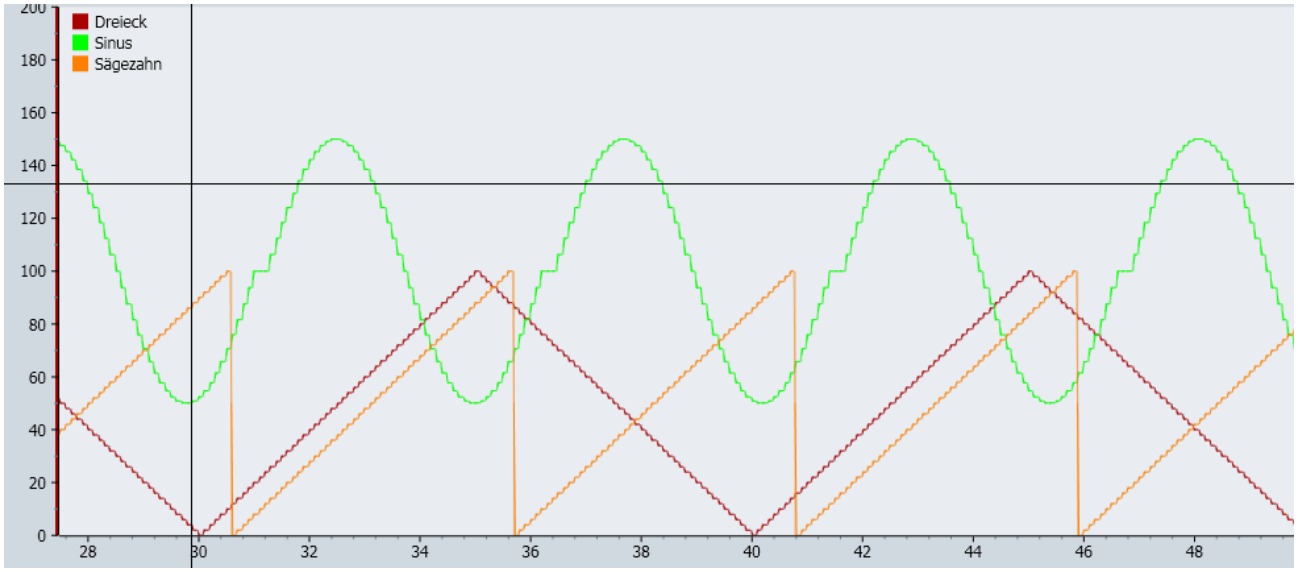
Falls keine Simulation gestartet ist, sehen Sie ein statisches Bild der aufgezeichneten Signale.

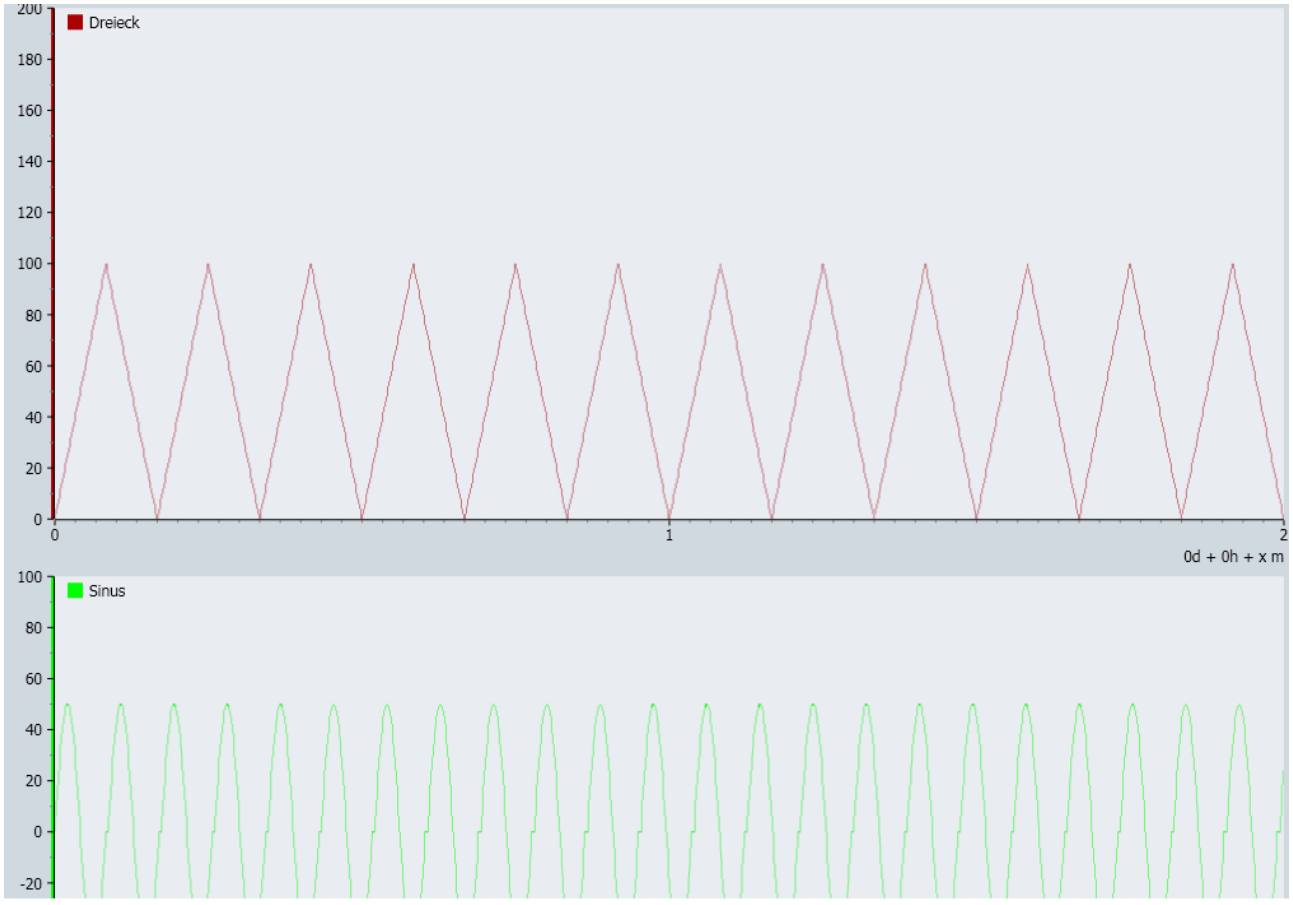
Darstellung der Kurven überlagert und untereinander

Alle Analog- und Integersignale eines Kurvenbildes können in einem gemeinsamen Kurvendiagramm *überlagert* darstellt werden, oder alle Signale werden *untereinander* in jeweils eigenen Diagrammen angezeigt.

In der überlagerten Darstellung kann nur für eines der Signale die passend beschriftete Y-Achse angezeigt werden. Durch Anklicken des Signalnamens können Sie die darzustellende Y-Achse festlegen. Die Y-Achse wird in der Farbe des Signals markiert.

Die folgenden beiden Abbildungen zeigen ein überlagert dargestelltes Kurvenbild (Abbildung oben) und ein untereinander dargestelltes Kurvenbild (Abbildung unten):





Kurven von Binärsignalen werden immer in einzelnen, untereinander angeordneten Kurvenverläufen mit einer gemeinsamen Zeitachse dargestellt.

5.1.4.4 Kurvenbildeditor

Funktionen des Kurvenbildeditors

Mit dem Kurvenbildeditor können Sie die folgenden Funktionen durchführen:

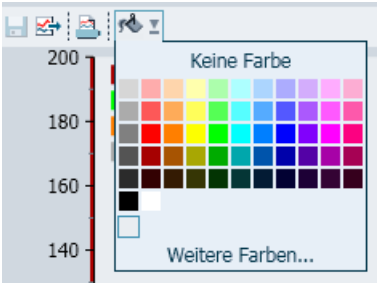
- Hintergrundfarbe der Kurvenbilder anpassen
- Kurvenbilder ausdrucken
- Kurvenbilder exportieren

Die entsprechenden Befehle finden Sie in der Symbolleiste des Kurvenbildeditors:



Hintergrundfarbe von Kurvenbildern

Die Hintergrundfarbe eines Kurvenbildes ist frei wählbar. Benutzen Sie dazu die Farbauswahl in der Werkzeugleiste des Kurvenbildeditors:



Drucken von Kurvenbildern

Um ein Kurvenbild auszudrucken, gehen Sie folgendermaßen vor:

- Klicken Sie auf das Symbol "" im Kurvenbildeditor.
  Das Dialogfeld "Drucken" wird geöffnet.
- Im Dialogfeld selektieren Sie eines oder mehrere Signale, die ausgedruckt werden sollen und wählen unter "Darstellung" einen Signalverlauf "untereinander" oder "überlagert" aus. Die weiteren Funktionen des Dialogfeldes finden Sie im Abschnitt: Dialogfeld "Drucken" (Seite 978).

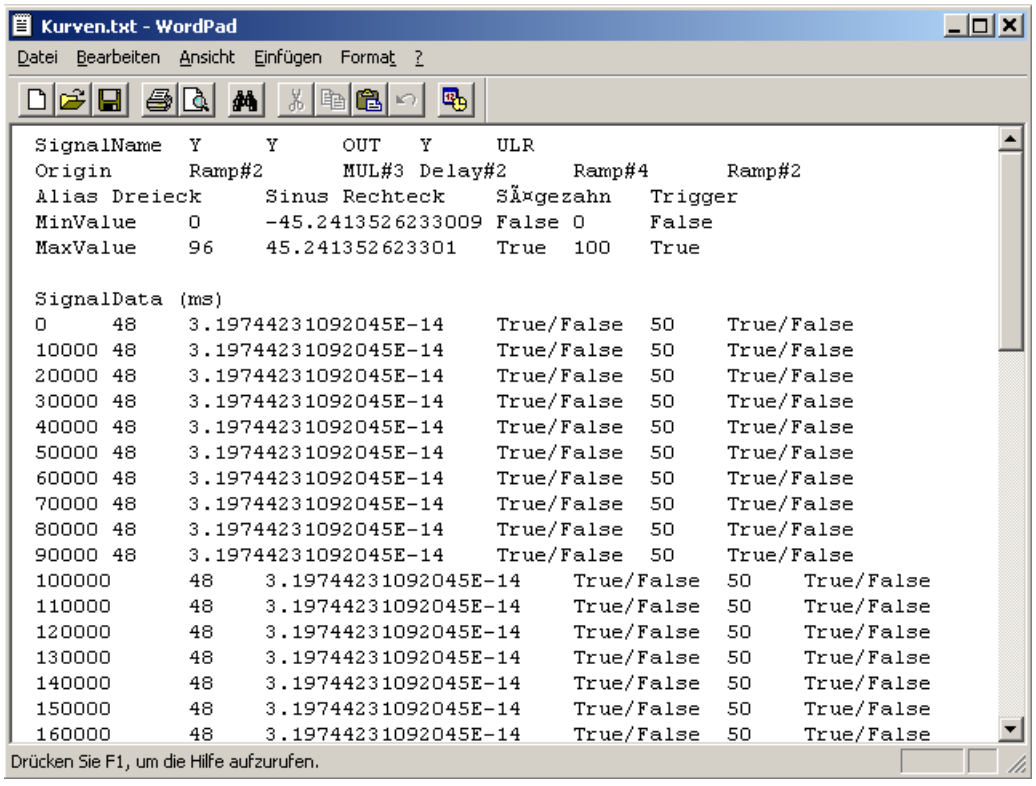
Export von Kurvenbildern

Um die in einem Kurvenbild aufgezeichneten Signale in eine Textdatei (*.txt) zu exportieren, gehen Sie folgendermaßen vor:

1. Klicken Sie auf das Symbol "" im Kurvenbildeditor.
   Das Dialogfeld "Exportieren" wird geöffnet.
2. Wählen Sie die Signale aus, die exportiert werden sollen.
3. Geben Sie unter "Datei" der Exportdatei einen Namen.

4. Klicken Sie auf die Schaltfläche "..." um einen Speicherort für die Exportdatei auszuwählen.
5. Wählen Sie unter "Auflösung" einen zeitlichen Abstand für die Signale aus. Mögliche Einstellungen sind 100 ms, 1 s, 10 s oder 60 s.

In der erzeugten Datei werden die Zeitpunkte und die Werte für jedes Signal in je einer Spalte angeordnet. Die einzelnen Spalten sind mit dem Tabulator-Zeichen getrennt. Wenn bei analogen oder ganzzahligen Signalen in einen Zeitraum mehrere verschiedene Signalwerte vorliegen, wird der arithmetische Mittelwert zwischen größtem und kleinstem vorkommenden Wert verwendet.



5.2 Suchen & Ersetzen

Sie finden die Funktion "Suchen & Ersetzen" in der Projektnavigation.

Mit einem Doppelklick wird die Funktion im Arbeitsbereich geöffnet.

Nutzen Sie diese Funktion, um nach Elementen in Ihrem Projekt zu suchen und Ersetzungen vorzunehmen.

In den Signaleigenschaften von Komponenten und Controls können Sie über das Symbol "" nach Referenzen eines Signals in Projektelementen suchen.

Weitere Informationen zur Suchen-Funktion finden Sie im Kapitel: Suchen (Seite 300).

Weitere Informationen zur Ersetzen-Funktion finden Sie im Kapitel: Ersetzen (Seite 303).

Hinweis

Suchen & Ersetzen in Texten

Mit den Tastenkombinationen <Strg+F> und <Strg+H> rufen Sie das Dialogfeld zum "Suchen" oder "Ersetzen" auf.

Siehe auch

Dialogfeld "Suchen & Ersetzen" (Seite 977)

5.2.1 Suchen

5.2.1.1 Suchen mit dem Editor Suchen & Ersetzen

Der Editor Suchen & Ersetzen öffnet sich im Arbeitsbereich. Für diesen Editor stehen die Task-Cards "Komponenten", "Makros" und "Signale" zur Verfügung.

Sie können nach folgenden Elementen suchen:

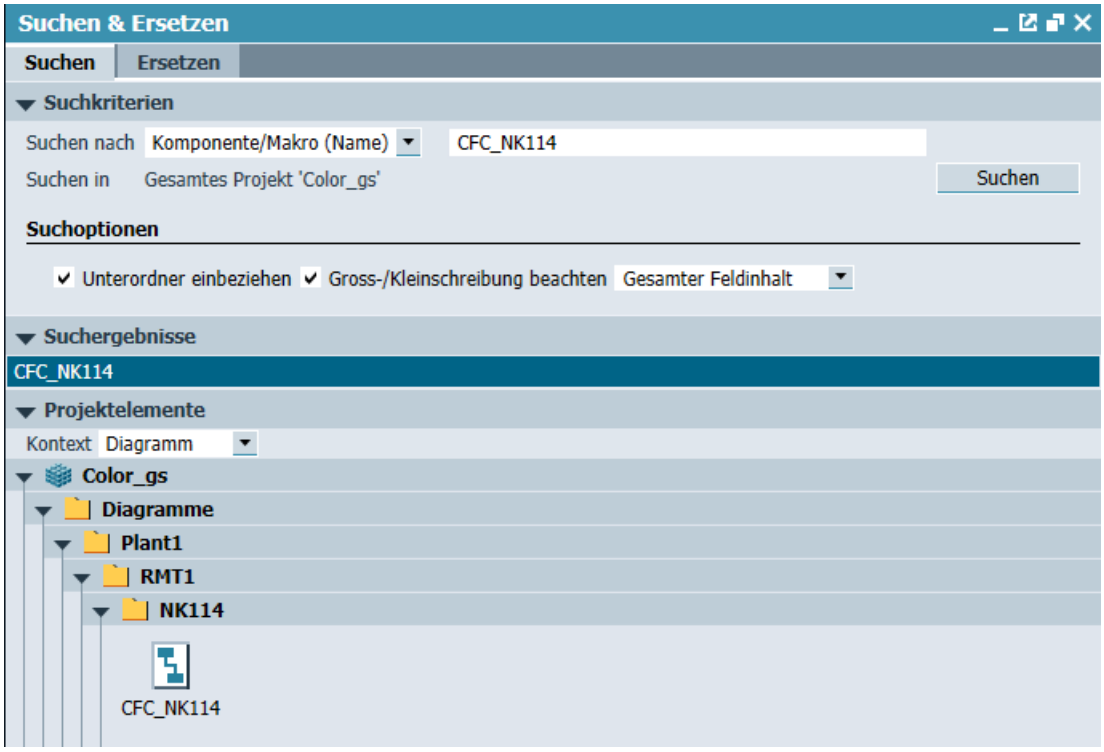
- Signale
- Konnektoren
- Namen von Komponenten und Makrokomponenten
- Komponenten und Makrokomponenten nach Typ
- Grafiktext

Hinweis

Die Suche funktioniert nur auf gespeicherten Daten, deshalb müssen Diagramme und Kopplungen vor der Suche gespeichert werden.

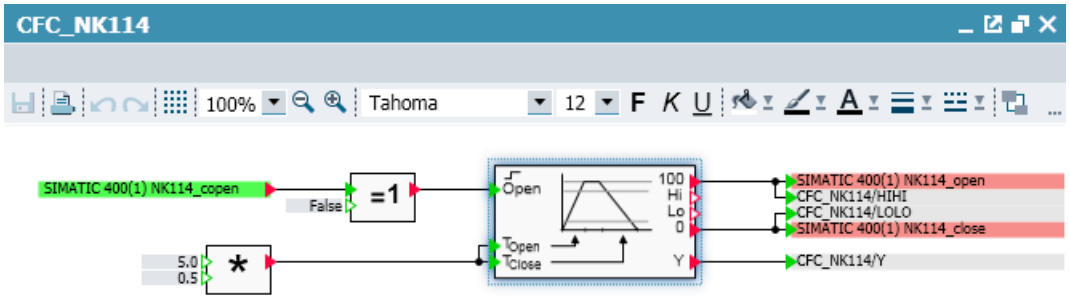
Wählen Sie ein Element aus und klicken Sie auf die Schaltfläche "Suchen", um die Suche zu starten.
Abhängig vom ausgewählten Element können Sie weitere Suchkriterien und Suchoptionen eingeben, z. B. einen Signalnamen bei der Suche nach einem Signal.

In der folgenden Abbildung wurde nach Komponenten und Makrokomponenten gesucht, welche im Namen "CFC_NK114" enthalten. Gesucht wurde im gesamten Projekt mit Unterordnern und unter Berücksichtigung von Groß- und Kleinschreibung. Unter "Suchergebnisse" sehen Sie die Ergebnisse: Die Komponente CFC_NK114.



Wählen Sie unter "Suchergebnisse" eine Komponente oder Makrokomponente aus, um unter "Projektelemente" die Diagramme und Kopplungen anzuzeigen, die die gesuchte Komponente oder Makrokomponente enthalten.

In der Abbildung oben wird bei der Auswahl des Suchergebnisses "CFC_NK114" unter "Projektelemente" das Diagramm "CFC_NK114" angezeigt. Wenn Sie das Diagramm "CFC_NK114" mit Doppelklick öffnen, sehen Sie, dass die gesuchte Komponente mit einem blauen Rahmen gekennzeichnet ist.



Wenn ein Suchergebnis genau einem Diagramm zugeordnet werden kann, können Sie das zugehörige Diagramm durch einen Doppelklick auf das Suchergebnis öffnen.

Für die Suche nach einem bestimmten Signal können Sie das zu suchende Signal einfach per "Drag & Drop" aus der Task-Card "Signale" in eines der beiden Eingabefelder ziehen.

Bei der Suche nach Komponententypen geben Sie den Namen und/oder die ID des Komponententyps an. Entsprechend können Sie bei der Suche nach einem bestimmten

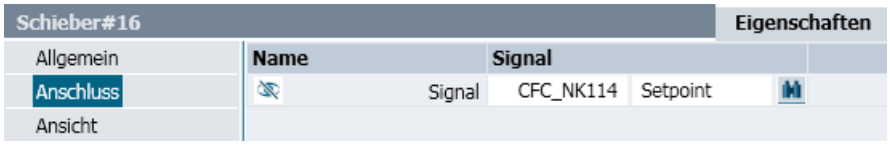
Komponententyp in Ihrem Projekt den Komponententyp einfach aus der Task-Card "Komponenten" in das Eingabefeld ziehen.

Für die Suche nach Komponenten und Konnektoren geben Sie deren Namen ein. Mit der Suchoption "Nur Ausgangskonnektoren" können Sie bei der Suche nach globalen Konnektoren alle Suchergebnisse von Eingangskonnektoren ausblenden und so direkt zum jeweiligen Ausgangskonnektor navigieren.

5.2.1.2 Suchen über die Eigenschaften von Signalen und Konnektoren

Signale und Konnektoren aus einem Diagramm können auch über das Eigenschaftsfenster gesucht werden.

Wenn Sie z. B. in einem Diagramm nach Eingangskonnektoren zum Ausgangskonnektor SPEED suchen wollen, öffnen Sie das Eigenschaftsfenster und klicken Sie auf die Schaltfläche "Suchen" ( ).

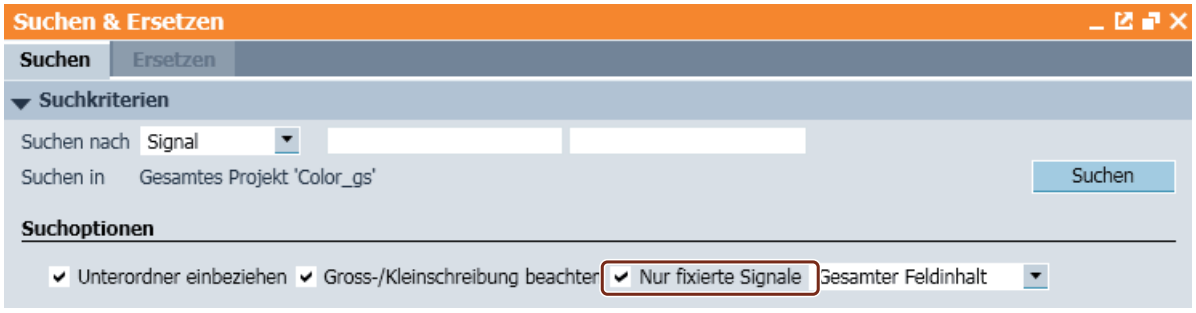


Der Editor Suchen & Ersetzen mit den Suchergebnissen wird geöffnet.

Anzeige der Suchergebnisse und Navigation zu den Suchergebnissen und sind nun wie im Abschnitt: Suchen mit dem Editor Suchen & Ersetzen (Seite 300).

5.2.1.3 Suchen von fixierten Signalen

Um nach fixierten Signalen im Projekt zu suchen, wählen Sie bei der Suche nach Signalen die Suchoption "Nur fixierte Signale" aus. Damit erhalten Sie jederzeit einen Überblick, welche Signale aktuell fixiert sind.



Da Signale nur fixiert werden können, wenn die Simulation läuft, steht Ihnen diese Suchoption erst nach dem Starten der Simulation zur Verfügung.

5.2.2 Ersetzen

5.2.2.1 Ersetzen mit dem Editor Suchen & Ersetzen

Zum Ersetzen von Suchergebnissen klicken Sie im Editor auf die Registerkarte "Ersetzen". Ersetzen können Sie

- Signale
- Konnektoren
- Namen von Komponenten und Makrokomponenten
- Komponenten und Makrokomponenten nach Typ
- Grafiktext

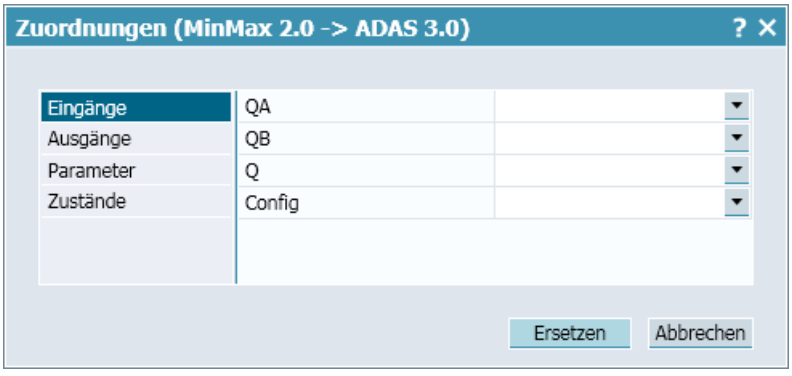
Hinweis

Das Ersetzen funktioniert nur auf gespeicherten Daten. Diagramme und Kopplungen müssen deshalb vor der Ersetzung gespeichert werden.

- Geben Sie die Suchkriterien und Optionen ein. Befüllen Sie zusätzlich die Textfelder zu "Ersetzen durch" und klicken Sie dann auf die Schaltfläche "Suchen".
- Wählen Sie in den Suchergebnissen aus, welche Treffer Sie ersetzen wollen. Entsprechend der eingestellten Suchoptionen werden Ihnen die Ergebnisse der Ersetzung für jeden Treffer angezeigt. Unter "Projektelemente" werden Ihnen wie beim Suchen die Diagramme angezeigt, in denen die gesuchten Signale enthalten sind.

Sie können auch Komponenten oder Makrokomponenten eines Typs durch eine Komponente oder Makrokomponente eines anderen Typs in Ihrem Projekt ersetzen. Die Typen werden mit ihrer eindeutigen Kennung UID übernommen.

- Klicken Sie auf die Schaltfläche "Ersetzen".
  Das Dialogfeld "Zuordnungen" öffnet sich.



- Das Dialogfeld zeigt die Zuordnungen der Ein- und Ausgänge der beiden Elemente an. Diese Zuordnung kann hier geändert werden.
- Klicken Sie auf die Schaltfläche "Ersetzen", um die Zuordnungen zu übernehmen und das Ersetzen durchzuführen.

5.2.2.2 Aktualisieren

Aktualisieren von Komponenten

Mit dem Suchkriterium "Aktualisieren" wird für alle Komponenten im definierten Suchbereich nach Komponententypen gesucht, die denselben Namen haben und derselben Bibliotheksfamilie angehören, die aber zu einem späteren Zeitpunkt erstellt (gespeichert) worden sind.
Damit bringen Sie die Komponenten in Ihrem Simulationsprojekt mit einem Arbeitsschritt auf den aktuellen Stand. Die Suche nach aktuellen Komponententypen erfolgt in allen Komponentenbibliotheken der Task-Card "Komponenten", in der Reihenfolge der Bereiche "Projektkomponenten", "Eigene Komponenten" und "Basiskomponenten". Der Komponententyp, der zuletzt erstellt worden ist, wird dann als aktueller Typ zum Austausch im Suchergebnis vorgeschlagen.

Das Aktualisieren erfolgt gleichermaßen für alle im definierten Suchbereich enthaltene Makrokomponenten, wobei die aktuellen Makrokomponenten in den Bibliotheken der Task-Card "Makros" gesucht werden. Falls bereits alle Komponenten und Makrokomponenten im gewählten Suchbereich aktuell sind, werden keine Treffer angezeigt.

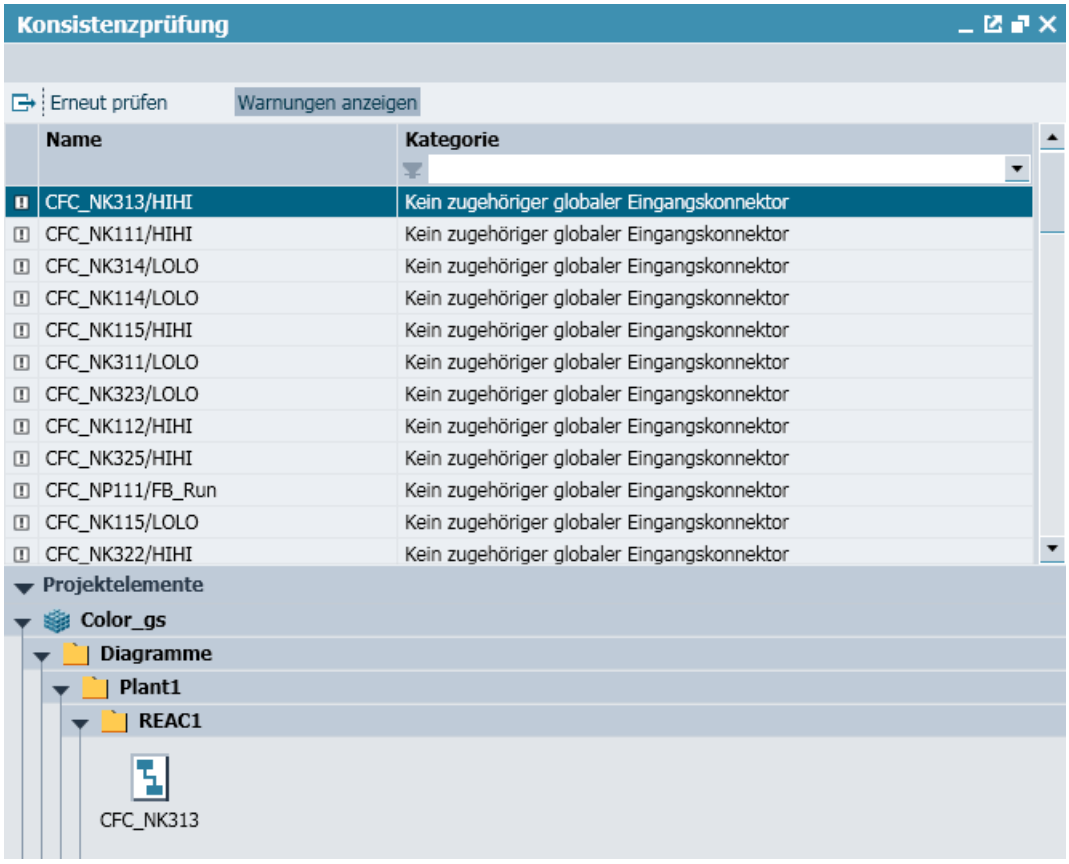
Die Alternative "Aktualisieren" steht Ihnen auch zur Verfügung, wenn Sie Komponenten typbezogen in Makrokomponenten aktualisieren wollen. Initiieren Sie dazu das Aktualisieren über das Kontextmenü der Makrokomponente.

5.3 Konsistenzprüfung

Die Konsistenzprüfung erfolgt immer automatisch, wenn die Simulation gestartet wird. Inkonsistente Projekte können nicht gestartet werden.

Starten Sie die Konsistenzprüfung manuell, indem Sie einen Doppelklick auf den Baumeintrag "Konsistenzprüfung" in der Projektnavigation durchführen. Für ein konsistentes Projekt wird in einem Dialog angezeigt, dass keine Inkonsistenzen vorliegen.

Wenn Sie beispielsweise im Projekt "Aufzug-03" den Konnektor "SPEED" im Diagramm Hauptantrieb in "Schachtzaehler" umbenennen, dann existieren zwei Ausgangskonnektoren gleichen Namens im Projekt. Das Projekt ist nicht mehr konsistent, und die Konsistenzprüfung öffnet sich nach dem Starten mit den gefundenen Inkonsistenzen im Arbeitsbereich (siehe Abbildung unten). Da die Konsistenzprüfung immer automatisch beim Starten einer Simulation angestoßen wird, öffnet sich die Konsistenzprüfung gleichermaßen im Arbeitsbereich, wenn Sie versuchen, dieses inkonsistente Projekt zu starten.



Inkonsistenzen sind durch das vorangestellte rote Symbol "", Warnungen durch das Symbol "" gekennzeichnet. Die Anzeige von Warnungen kann über Befehl zu- und abgeschaltet werden.

Für das Beispiel werden zwei Treffer angezeigt:

1. Für den Eingangskonnektor "SPEED" existiert kein Ausgangskonnektor.
2. Der Ausgangskonnektor "Schachtzaehler" existiert mehrfach.

Der erste Treffer ist nur eine Warnung und gibt einen Hinweis auf ein eventuell noch nicht fertiges Simulationsmodell. Eine Warnung verhindert den Start der Simulation nicht. Der zweite Treffer bezeichnet einen Fehler, der vor dem Starten der Simulation behoben werden muss.

Zur Behebung von Fehlern oder Warnungen selektieren Sie diese in der Trefferliste. Die Diagramme, in denen der Fehler oder die Warnung beseitigt werden soll, werden Ihnen dann in den Projektelementen angezeigt. Sie können dann die Diagramme zur Korrektur aus dieser Ansicht heraus unmittelbar öffnen. Nach Beseitigung der Inkonsistenzen können Sie eine erneute Prüfung über die Schaltfläche "Erneut prüfen" anstoßen.

Hinweis

Die Konsistenzprüfung funktioniert nur mit gespeicherten Daten. Speichern Sie deshalb vorher Ihre Daten, Pläne und Kopplungen.

Die Ergebnisse der Konsistenzprüfung können auch in eine tabulatorgetrennte Textdatei exportiert werden. Aktivieren Sie dazu die in der folgenden Abbildung gekennzeichnete Schaltfläche im Editor der Konsistenzprüfung.



5.4 Analysefunktionen

5.4.1 Grundlagen zur Analysefunktion

Einleitung

Die Analysefunktion von SIMIT erzeugt eine statistische Auswertung aus den Daten eines SIMIT-Projekts. Die statistische Auswertung wird als Analysedatei im Format "*.xslx" gespeichert. Berücksichtigt werden folgende Projektdaten:

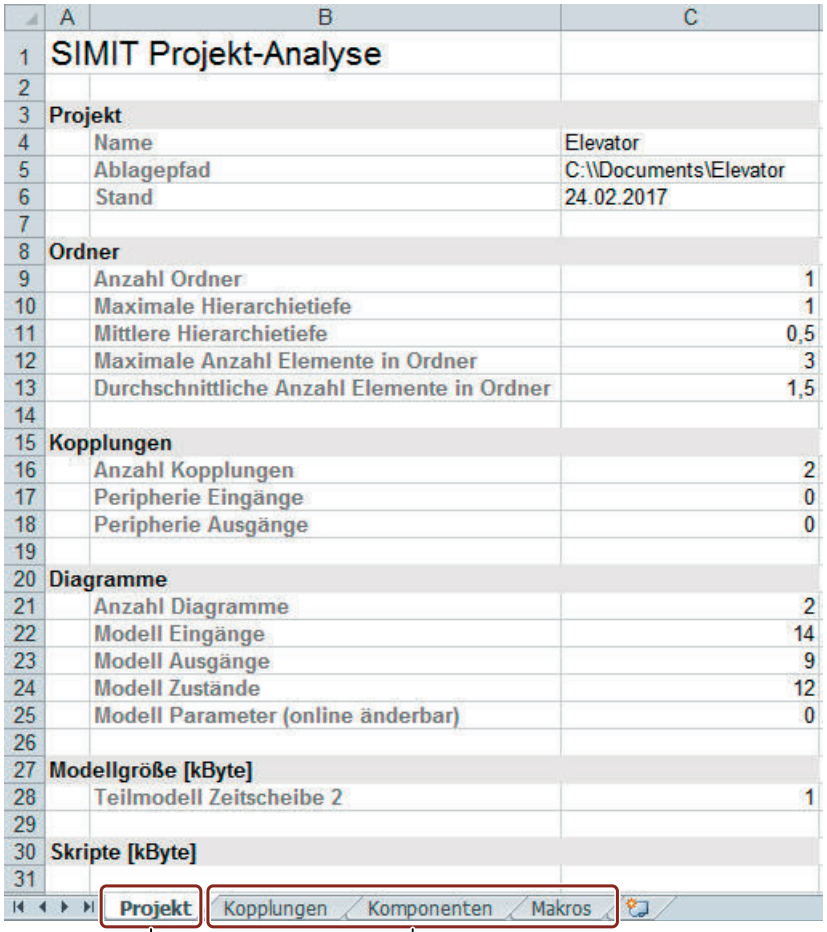
- Kopplungen
- Komponenten
- Makros

Zusätzlich werden Mengengerüste für folgende Objekte ermittelt:

- Ordner
- Kopplungen
- Diagramme
- Modellgröße
- Skripte

Aufbau der Analysedatei

Die folgende Abbildung zeigt den Aufbau der Analysedatei:



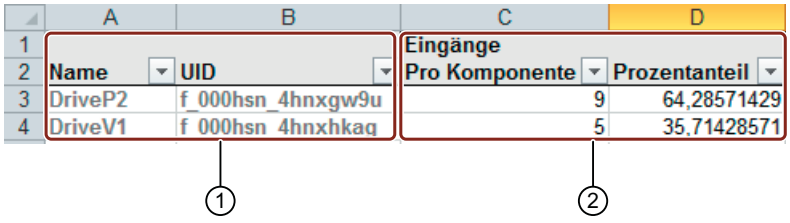
	A	B	C
1	SIMIT Projekt-Analyse		
2			
3	**Projekt**		
4		Name	Elevator
5		Ablagepfad	C:\\Documents\Elevator
6		Stand	24.02.2017
7			
8	**Ordner**		
9		Anzahl Ordner	1
10		Maximale Hierarchietiefe	1
11		Mittlere Hierarchietiefe	0,5
12		Maximale Anzahl Elemente in Ordner	3
13		Durchschnittliche Anzahl Elemente in Ordner	1,5
14			
15	**Kopplungen**		
16		Anzahl Kopplungen	2
17		Peripherie Eingänge	0
18		Peripherie Ausgänge	0
19			
20	**Diagramme**		
21		Anzahl Diagramme	2
22		Modell Eingänge	14
23		Modell Ausgänge	9
24		Modell Zustände	12
25		Modell Parameter (online änderbar)	0
26			
27	**Modellgröße [kByte]**		
28		Teilmodell Zeitscheibe 2	1
29			
30	**Skripte [kByte]**		
31			

① Mengengerüste

② Statistische Daten

Inhalt der statistischen Daten

Die folgende Abbildung zeigt die prinzipielle Struktur der statistischen Daten am Beispiel der "Komponenten":



	A	B	C	D
1			Eingänge	
2	Name	UID	Pro Komponente	Prozentanteil
3	DriveP2	f_000hsn_4hnxgw9u	9	64,28571429
4	DriveV1	f_000hsn_4hnxhkag	5	35,71428571

① Aufgelistet werden die im Projekt verwendeten Typen

② Der Prozentanteil wird bezogen auf die Gesamtanzahl berechnet.

5.4.2 Auswertung generieren

Voraussetzung

- Projekt ist in SIMIT geöffnet.
- Projektsicht wird angezeigt.

Vorgehen

1. Wählen Sie in der Projektnavigation im Kontextmenü des Projekts den Befehl "Analyse".
2. Geben Sie einen Namen für die Analyse-Datei ein.
   Wenn Excel auf dem PC installiert ist, wird die Analyse-Datei sofort in Excel geöffnet.

Ergebnis

Die Analyse-Datei wird erzeugt und im angegebenen Verzeichnis gespeichert.

Skripte

6

6.1 Funktionsweise von Skripten

Eine Simulation wird funktional in SIMIT durch die grafische Verschaltung und Parametrierung von Komponenten festgelegt. Beim Starten der Simulation wird aus allen Diagrammen eines Simulationsprojekts ein ablauffähiges Simulationsmodell generiert, das dann zyklisch bearbeitet wird. Bei laufender Simulation können Sie über die Benutzeroberfläche von SIMIT auf alle Ein- und Ausgangssignale sowie die Zustände und online änderbaren Parameter der Komponenten zugreifen.

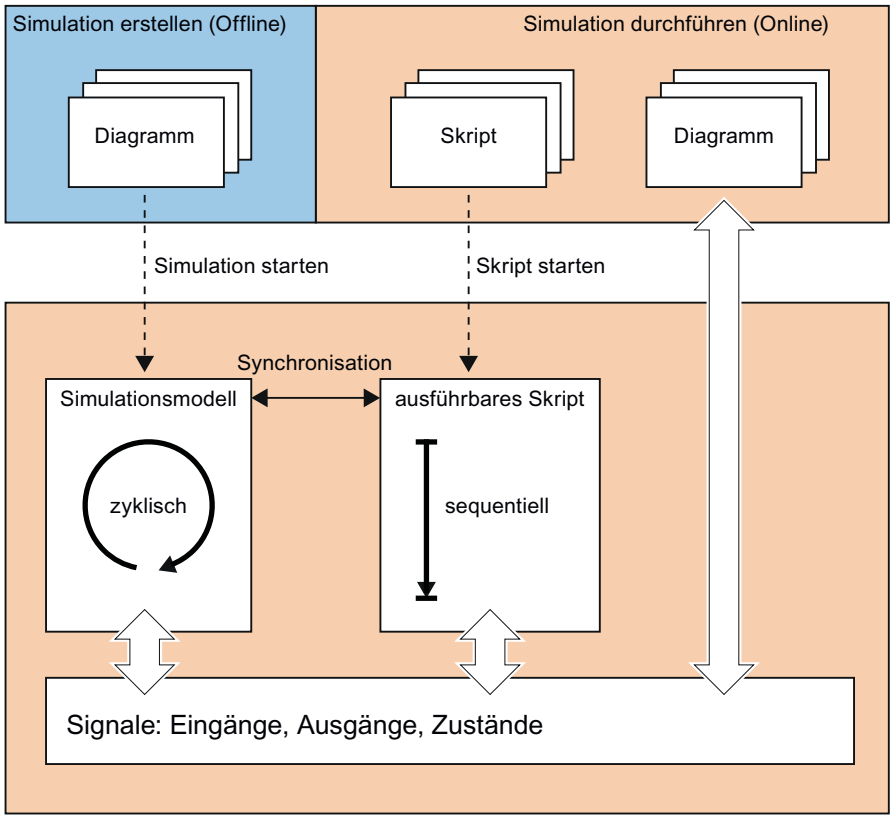
Das Funktionsmodul Automatic Control Interface von SIMIT bietet Ihnen eine zusätzliche Möglichkeit, mit Hilfe von Skripten bei laufender Simulation zu festgelegten Zeitpunkten auf Simulationsgrößen zuzugreifen und somit manuelle Eingriffe über die Benutzeroberfläche zu automatisieren bzw. Vorgänge in der Simulation zu überwachen und durch Ausgabe von Informationen auch zu protokollieren.

Ein Skript besteht aus einer Folge von Anweisungen, die sequentiell und ohne Sprünge von der ersten bis zur letzten Anweisung abgearbeitet werden. Das Skript und das zyklisch ausgeführte Simulationsmodell sind über Zeitscheiben synchronisiert. Innerhalb der Zeitscheibe wird das Skript vor dem Simulationsmodell dieser Zeitscheibe ausgeführt.

Sie können ein Skript anhalten, um auf bestimmte Ereignisse im Simulationsmodell oder Zeitpunkte zu warten. Darüber hinaus können Sie mit Anweisungen im Skript Einfluss auf die Steuerung des Simulationsmodells nehmen, beispielsweise Schnappschüsse erzeugen und laden.

Folgende Abbildung zeigt schematisch die Einbindung von Skripten in die Architektur von SIMIT:

Benutzeroberfläche



Ablaufsteuerung

6.2 Handhabung von Skripten

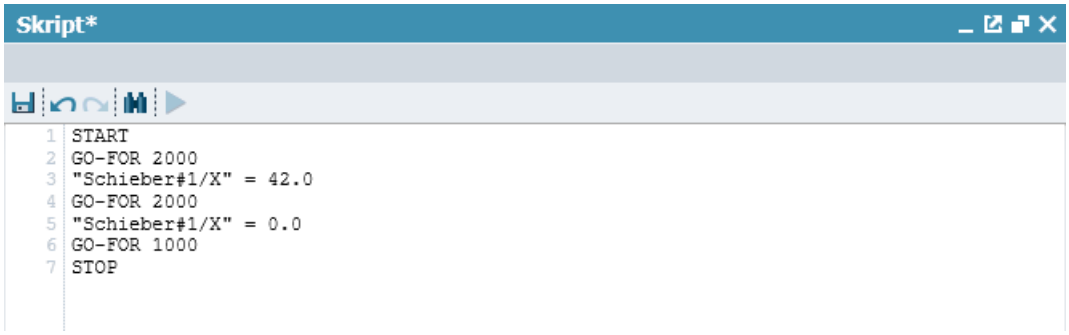
6.2.1 Erstellen eines Skripts

Skripte können Sie in Ihrem Simulationsprojekt sowohl erstellen als auch ausführen. Zum Editieren Ihrer Skripte steht Ihnen der Skripteditor zur Verfügung.

Skripte sind Bestandteile eines SIMIT-Projekts und werden im Projektbaum im Ordner "Skripting" verwaltet. Sie können Skripte immer erstellen und bearbeiten, wenn SIMIT geöffnet ist, also sowohl wenn die Simulation noch nicht gestartet ist (offline) als auch wenn sie gestartet ist (online). Mit einem Doppelklick auf den Eintrag "Neues Skript" im Ordner "Skripting" legen Sie ein neues Skript an.

Zum Bearbeiten eines Skripts öffnen Sie den Skripteditor über das Kontextmenü oder per Doppelklick auf das Skript im Navigationsfenster.

Der sich dann öffnende Editor enthält am linken Rand eine Spalte mit der Anzeige der Zeilennummer, damit Sie Fehlermeldungen leichter zuordnen können.



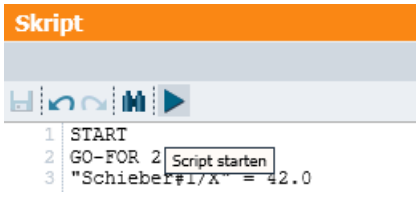
Im Eigenschaftsfenster des Skripteditors geben Sie an, in welcher Zeitscheibe das Skript ausgeführt werden soll. Wenn Sie weitere Skripte per Anweisung inkludieren, ist die Angabe der Zeitscheibe in den inkludierten Skripten irrelevant; entscheidend ist die im gestarteten Skript angegebene Zeitscheibe.



Wenn Sie die Option "Oberfläche blockieren" aktivieren, erscheint nach dem Starten des Skripts ein Dialog, der verhindert, dass über die Oberfläche in die laufende Simulation eingegriffen werden kann. Wenn Sie diesen Dialog schließen, wird das Skript beendet.

6.2.2 Ausführen eines Skriptes

Ein Skript kann nur bei laufender Simulation ausgeführt werden. Sie starten ein Skript entweder über sein Kontextmenü im Projektbaum oder über die Toolbar im Skripteditor.



Wenn ein Skript gestartet ist, wird dies durch das blinkende Symbol { } in der Statuszeile angezeigt. Zusätzlich ist das Symbol für die laufende Simulation eingeblendet.

Über das Kontextmenü des Ordners "Skripting" können Sie die Ausführung eines Skripts jederzeit beenden.

Sie können nicht mehrere Skripte gleichzeitig ausführen, d. h. Sie können immer nur ein Skript zur Ausführung bringen. Die Ausführung eines Skripts wird beendet, wenn alle Anweisungen im Skript ausgeführt sind, wenn es durch einen Befehl beendet wird oder wenn die Simulation an der Oberfläche geschlossen oder neu initialisiert wird.

Nach dem Starten eines Skripts wird die Syntax geprüft und das Skript wird in eine ausführbare Anweisungsliste übersetzt. Falls bei der Prüfung ein Fehler im Skript entdeckt wird, wird dieser gemeldet, und das Skript wird nicht gestartet.

Darüber hinaus können auch Probleme auftreten, die erst während der Abarbeitung des Skripts erkannt werden können. In diesen Fällen wird eine entsprechende Meldung in der Statuszeile ausgegeben und das Skript wird beendet.



6.2.3 Externe Skripte

Hinweis

Da Sie nur Skripte ausführen können, die in der Projektnavigation vom SIMIT liegen und auch nur diese Skripte mit dem Projekt archiviert werden, empfehlen wir Ihnen, Skripte stets mit dem Skripteditor von SIMIT zu erstellen und zu bearbeiten und damit im Simulationsprojekt zu halten.

Wenn Sie dennoch Skripte nicht mit dem Skripteditor, sondern als externe Skripte mit einem eigenen Editor erstellen wollen, beachten Sie Folgendes:

Skripte sind Textdateien in der üblichen Windows-Zeichenkodierung (Codepage 1252). Die Namen von Skriptdateien müssen die Dateiendung ".script" haben. Die Eigenschaften eines Skripts, die Sie im Eigenschaftsfenster des Skripteditors finden, sind in der Skriptdatei in den ersten beiden Zeilen entsprechend den Angaben in der Tabelle "Schlüsselwörter für die Eigenschaften von Skripten" wie folgt zu setzen:

```
Schlüsselwort = Wert
```

Beispiel:

```
META_BLOCKGUI = False
META_CYCLE = 2
```

Tabelle 6-1 Schlüsselwörter für die Eigenschaften von Skripten

Schlüsselwort	Bedeutung
`META_BLOCKGUI`	`True`: Benutzeroberfläche ist während der Skriptausführung blockiert `False`: Benutzeroberfläche ist während der Skriptausführung nicht blockiert
`META_CYCLE`	Zeitscheibe (1 .. 8), in der das Skript ablaufen soll

Weitere Informationen hierzu finden Sie im Abschnitt: Skripte zusammensetzen (Seite 314).

Hinweis

Beim Archivieren eines Projekts werden nur die Skripte archiviert, die sich im Simulationsprojekt befinden.

Externe Skripte werden nicht mit dem Projekt archiviert. Das archivierte Projekt ist somit nicht vollständig.

6.3 Skriptsyntax

6.3.1 Steuern des Skripts

Allgemeines

Sie können die Abarbeitung eines Skripts durch eine Anweisung im Skript beenden oder über einen Dialog zur Laufzeit des Skripts über die weitere Abarbeitung entscheiden.

Das Skript abbrechen

Mit der Anweisung
```
BREAK
```

wird die Bearbeitung eines Skriptes abgebrochen. Steht die *BREAK*-Anweisung in einem inkludierten Skript, dann wird mit der Abarbeitung im aufrufenden Skript fortgefahren.

Das Skript beenden

Mit der Anweisung
```
QUIT
```

wird das Skript beendet. Steht die *QUIT*-Anweisung in einem inkludierten Skript, dann wird auch das aufrufende Skript beendet.

Der Abfragedialog

Mit der Anweisung
```
DIALOG "text" MODUS
```

wird die Simulation angehalten und es wird ein Abfragedialog mit dem Inhalt text geöffnet. Mit dem Parameter *Modus* kann nach der Tabelle "Schaltflächen für Dialoge" festgelegt werden, welche Schaltflächen dieser Dialog besitzen soll.

Tabelle 6-2 Schaltflächen für Dialoge

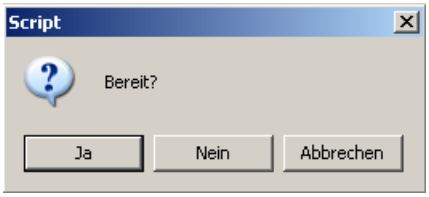
Modus	Schaltfläche			
	"Ja"	"Nein"	"OK"	"Abbrechen"
YESNOCANCEL	X	X	–	X
OKCANCEL	–	–	X	X
YESNO	X	X	–	–
OK	–	–	X	–

Voreingestellt ist der Modus *YESNOCANCEL*. Bei Betätigen von "Abbrechen" im Dialog wird das Skript beendet, die Simulation bleibt angehalten. In allen anderen Fällen wird die Simulation wieder gestartet und die Skriptabarbeitung wird fortgesetzt.

Für das Beispiel

DIALOG "Bereit?" YESNOCANCEL

wird das in der folgenden Abbildung gezeigte Dialogfeld eingeblendet.



Wurde im Skript vor der *DIALOG*-Anweisung bereits eine Protokoll-Datei geöffnet, so wird in dieser Datei vermerkt, ob der Dialog mit der Schaltfläche "Ja" oder "Nein" geschlossen wurde. Dementsprechend entsteht ein Eintrag
`DIALOG text True`

oder
`DIALOG text False`

in der Protokolldatei.

Im Skript kann bei einem *YESNOCANCEL*-Dialog mit den binären Variablen
`_result`

und
`_ok`.

abgefragt werden, welche Entscheidung der Benutzer getroffen hat.

Die Variable `_result` wird mit *True* belegt, wenn der zuletzt geöffnete Dialog mit "Ja" quittiert wurde und mit *False*, wenn er mit "Nein" quittiert wurde. Zusätzlich wird die Variable `_ok` auf *False* gesetzt, wenn ein Dialog mit "Nein" quittiert wurde. Diese Variable wird nicht wieder automatisch auf *True* zurückgesetzt, kann aber bei Bedarf per Anweisung gesetzt werden.

6.3.2 Skripte zusammensetzen

Sie können ein Skript aus mehreren Einzelskripten zusammensetzen. Folgende Anweisung führt dazu, dass die Skriptbearbeitung mit einem anderen Skript fortgesetzt wird:
`INCLUDE "Name"`

Der Name des Skripts wird dabei relativ zum aufrufenden Skript angegeben. Trennstriche im Namen sind zu verdoppeln.

Beispiel:
`INCLUDE "..\\subscripts\\pruef2"`

Die *INCLUDE*-Anweisung ist ausgeführt, wenn das inkludierte Skript beendet ist.

Skripte können auch mehrfach verkettet werden, indem inkludierte Skripte ihrerseits wieder *INCLUDE*-Anweisungen enthalten. Die maximale Tiefe einer solchen Verkettung ist auf 10 begrenzt.

Sie können auch externe Skripte unter Angabe eines absoluten Pfades und des Namens der Skriptdatei inkludieren:
`INCLUDE "C:\\scripts\\pruef3"`

Die inkludierte Skriptdatei muss die Dateiendung *.script* besitzen. Die Dateiendung wird nicht im Dateinamen der *INCLUDE*-Anweisung angegeben.

Hinweis

Nach dem Starten eines Skripts wird für dieses Skript und alle inkludierten Skripte eine ausführbare Anweisungsliste erstellt. Die inkludierten Skripte müssen folglich schon beim Starten des aufrufenden Skripts existieren, Änderungen in einem inkludierten Skript nach dem Start des aufrufenden Skripts sind wirkungslos.

Hinweis

Beim Archivieren eines Projekts werden nur die Skripte archiviert, die sich im Simulationsprojekt befinden.

Externe Skripte werden nicht mit dem Projekt archiviert. Das archivierte Projekt ist somit nicht vollständig.

6.3.3 Skripte kommentieren

Kommentarzeilen beginnen mit zwei Schrägstrichen. Sie werden bei der Abarbeitung des Skripts ignoriert.

Beispiel:

```
// Das ist ein einzeiliger Kommentar
```

Außerdem können Sie ganze Bereiche im Skript auskommentieren, indem Sie den Anfang des Kommentars mit **/*** und das Ende mit ***/** kennzeichnen.

Beispiel:

```
/* Das ist ein

mehrzeiliger Kommentar */
```

Es ist nicht erlaubt, in derselben Zeile einen Kommentarbereich zu schließen und wieder einen neuen Kommentarbereich zu öffnen.

6.3.4 Signale in Skripten

Sie können mit Hilfe eines Skriptes auf Eingänge, Ausgänge, Zustände und online änderbare Parameter des Simulationsmodells zugreifen. Signale werden in SIMIT mit der *Quelle* des Signals und dem *Signalnamen* bezeichnet. Diese beiden Angaben werden für die Signalbezeichnung mit einem Schrägstrich zusammengefügt und in Anführungszeichen wie folgt gesetzt:
`"Quelle/Signalnamen"`

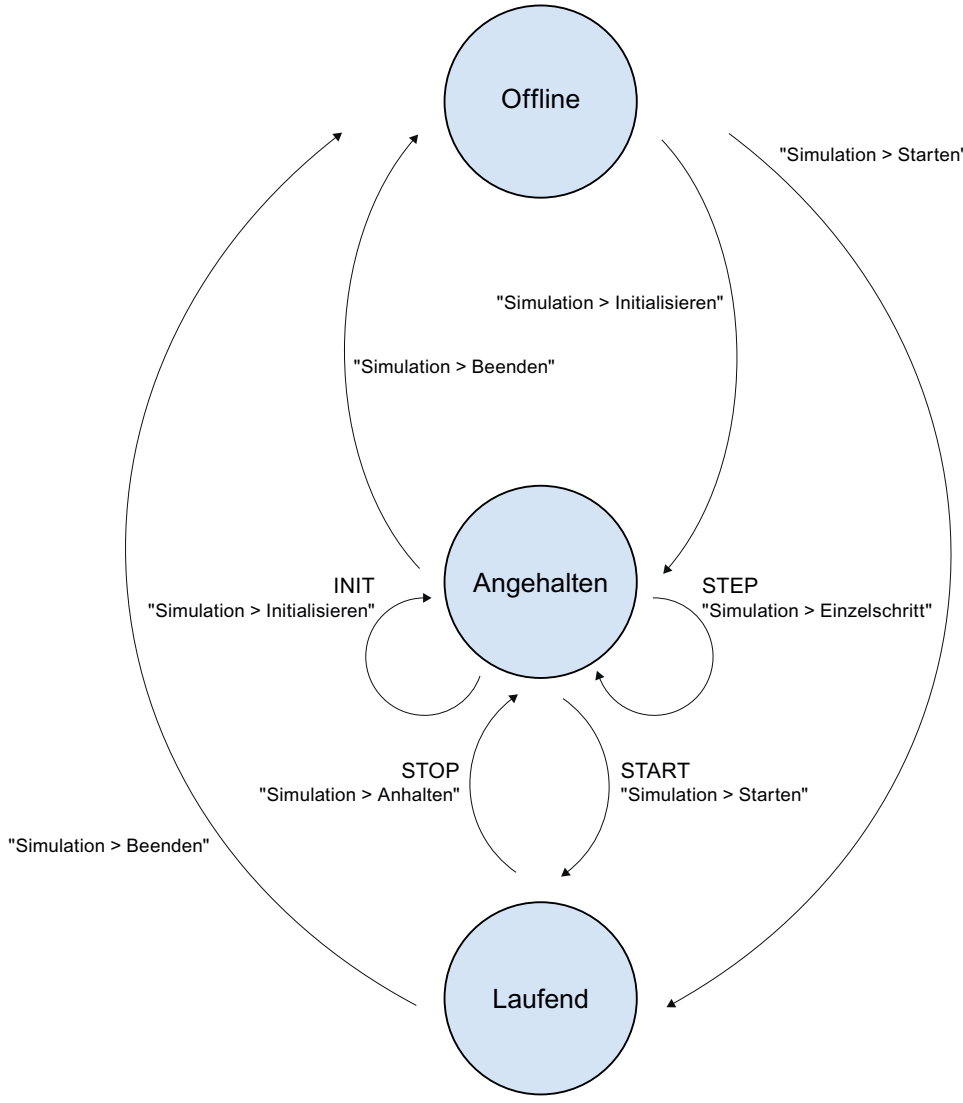
Beispiel:
`"ADD#1/IN1"`

Für den Sonderfall, dass im Namen der Quelle oder des Signals Anführungszeichen oder Schrägstriche enthalten sind, gelten folgende Regeln:

- Alle Anführungszeichen müssen mit einem vorangestellten "\"gekennzeichnet werden.
- Falls der Signalname selbst einen Schrägstrich enthält, so müssen alle Schrägstriche, die nicht als Trenner zwischen Quelle und Signal dienen sollen, mit einem vorangestellten "\" gekennzeichnet werden.
- Falls nur die Quelle Schrägstriche enthält, nicht aber der Signalnamen, so ist keine Kennzeichnung nötig, da der letzte Schrägstrich im Signalnamen als Trenner interpretiert wird, was in diesem Fall zu einem korrekten Ergebnis führt.

6.3.5 Steuern des Ablaufs der Simulation

Eine Simulation befindet sich immer in einem der Zustände "Offline", "Stopped" oder "Running".



Der Simulationszustand kann von einem Skript verändert werden. Beispielsweise kann die Simulation angehalten und wieder gestartet werden. Allerdings gehen alle Skriptbefehle vom Zustand "Stopped" oder "Running" aus, d. h. eine Simulation kann über ein Skript aus dem Zustand "Offline" heraus weder gestartet werden, noch ist das Beenden über Skript möglich.

6.3.5.1 Initialisieren der Simulation

Mit der Anweisung
`INIT`

wird die Simulation initialisiert. Diese Anweisung entspricht dem Befehl "Simulation > Initialisieren" im Menü von SIMIT.

Dieser Befehl ist nicht zulässig, wenn die Simulation bereits gestartet ist. In diesem Fall wird in der Statuszeile folgende Fehlermeldung ausgegeben und das Skript wird beendet:

Falscher Simulationszustand, Script '...', Zeile ... .

Hinweis

Der Übergang von "Offline" nach "Online" kann nicht über den Skriptbefehl *INIT* erreicht werden, da ein Skript nur im Online-Modus ausgeführt werden kann.

6.3.5.2 Starten der Simulation

Mit der Anweisung
`START`

wird die Simulation gestartet. Diese Anweisung entspricht dem Befehl "Simulation > Starten" im Menü bzw. dem Befehl über die Schaltfläche  in der Symbolleiste von SIMIT.

Eine *START*-Anweisung im Skript wird ignoriert, falls die Simulation bereits gestartet ist.

Hinweis

Der Übergang von "Offline" nach "Online" kann nicht über den Skriptbefehl *START* erreicht werden, da ein Skript nur im Online-Modus ausgeführt werden kann.

Das Starten über diesen Befehl setzt die Simulationszeit automatisch auf Echtzeit (100 %).

6.3.5.3 Starten und Warten auf einen absoluten Zeitpunkt

Mit der Anweisung
`GO-TO time`

wird mit der weiteren Ausführung des Skripts gewartet, bis die Simulationszeit größer oder gleich *time* Millisekunden ist.

Diese Anweisung wird ignoriert, falls die Simulationszeit bereits größer oder gleich *time* ist. Falls die Simulation initialisiert, aber noch nicht gestartet ist, wird sie mit diesem Befehl gestartet.

6.3.5.4 Starten und Warten auf einen relativen Zeitpunkt

Mit der Anweisung
`GO-FOR time`

wird mit der weiteren Ausführung des Skripts gewartet, bis *time* Millisekunden vergangen sind.

Falls die Simulation initialisiert, aber noch nicht gestartet ist, wird sie mit diesem Befehl gestartet.

6.3.5.5 Starten und eine bestimmte Anzahl von Zyklen warten

Mit der Anweisung
`GO n`

wird mit der weiteren Ausführung des Skripts für *n* Zyklen gewartet. Die Zyklen werden hier in der Zeitscheibe des Skripts gezählt. Für einen Zyklus, d. h. für `n=1` kann der Parameter *n* auch weggelassen werden.

Falls die Simulation initialisiert, aber noch nicht gestartet ist, wird sie mit diesem Befehl gestartet.

6.3.5.6 Starten und Warten auf ein Ereignis

Mit der Anweisung
`GO-UNTIL Bedingung TIMEOUT time`

wird mit der weiteren Ausführung des Skripts gewartet bis die *Bedingung* erfüllt ist oder bis in der Simulation die in Millisekunden angegebene Zeit *TIMEOUT* abgelaufen ist:

Beispiel:
`GO-UNTIL "OR#1/Y" TIMEOUT 10000`

Die Angabe von *TIMEOUT* mit einem Wert für Zeit ist nicht erforderlich. Das Skript beendet sich dann aber nicht mehr von alleine, falls die Bedingung nicht erfüllt wird.

Falls die Simulation initialisiert, aber noch nicht gestartet ist, wird sie mit diesem Befehl gestartet.

6.3.5.7 Anhalten der Simulation

Mit der Anweisung
`STOP`

wird die Simulation angehalten. Diese Anweisung entspricht dem Befehl "Simulation > Anhalten" im Menü.

Eine *STOP*-Anweisung im Skript wird ignoriert, falls die Simulation bereits angehalten wurde.

6.3.5.8 Ausführen eines Einzelschritts

Mit der Anweisung
`STEP`

wird ein Einzelschritt in der Simulation durchgeführt. Diese Anweisung entspricht dem Befehl "Simulation > Einzelschritt" im Menü von SIMIT. Dieser Funktion ist die Funktionstaste F12 zugeordnet.

Diese Anweisung ist nicht zulässig, wenn sich die Simulation im zyklischen Betrieb befindet. Andernfalls wird in der Statuszeile folgende Fehlermeldung ausgegeben und das Skript wird beendet:

Falscher Simulationszustand, Script '...', Zeile ... .

Hinweis

Ein Simulationsschritt entspricht immer einem Schritt des schnellsten Teilmodells, also des Teilmodells mit der kürzesten Zykluszeit.

6.3.5.9 Speichern eines Schnappschusses

Mit der Anweisung
`SAVE-IC "name"`

wird ein Schnappschuss unter dem Namen *name* im Simulationsprojekt gespeichert.

Der Schnappschuss wird immer im Ordner "Schnappschuss" auf der obersten Ebene gespeichert. Die Angabe von Unterordnern im Namen ist nicht zulässig.

ACHTUNG
Datenverlust Wenn in Ihrem Simulationsprojekt bereits ein Schnappschuss gleichen Namens vorhanden ist, dann wird dieser ohne Rückfrage überschrieben, und die in diesem Schnappschuss gespeicherten Daten sind verloren.

6.3.5.10 Laden eines Schnappschusses

Mit der Anweisung
`LOAD-IC "name"`

wird der Schnappschuss mit dem Namen *name* aus dem Ordner "Schnappschuss" des Simulationsprojekts geladen.

Die Angabe von Unterordnern im Namen ist zulässig. Die Trennstriche im Namen müssen dabei verdoppelt werden.

Beispiel:
```
LOAD-IC "name"
LOAD-IC "unterordner\\name"
```

Existiert kein Schnappschuss mit diesem Namen, wird in der Statuszeile folgende Fehlermeldung ausgegeben und das Skript wird beendet:

Der Schnappschuss '...' kann nicht geladen werden, möglicherweise existiert die Datei nicht, Script '...', Zeile ... .

Nach dem Laden eines Schnappschusses befindet sich die Simulation in gestopptem Zustand und muss per *STEP*- oder *START*-Anweisung weiter gesteuert werden.

6.3.5.11 Rücksetzen der Simulationszeit

Mit der Anweisung
`SIMTIME-RESET`

setzen Sie die Simulationszeit auf null zurück. Der Zustand der Simulation bleibt unverändert.

Hinweis

Das Zurücksetzen der Simulationszeit führt in den Kurvenbildern des Trend and Messaging Editors zu Inkonsistenzen, da alle im Kurvenbild dargestellten Werte über der Simulationszeit aufgezeichnet werden.

6.3.6 Protokollierung

Sie können Ergebnisse Ihrer Simulation, d. h. Signalwerte oder Ereignisse, protokollieren. Es steht Ihnen hierzu die Möglichkeit zur Verfügung, mit Anweisungen im Skript Protokolldateien anzulegen und darin Ausgaben zu dokumentieren.

Anweisungen im Skript zur Ausgabe werden bei der Abarbeitung des Skripts ignoriert, wenn keine Protokolldatei geöffnet ist.

6.3.6.1 Protokolldatei öffnen und schließen

Mit der Anweisung
`OPEN-LOG "name"`

wird eine Protokolldatei mit dem Namen *name* geöffnet. Eine eventuell bereits geöffnete Protokolldatei wird zuvor geschlossen. Die Protokolldatei muss mit ihrem absoluten Pfad angegeben werden. Trennstriche im Namen müssen verdoppelt werden.

Beispiel:
`OPEN-LOG "c:\\protocol\\log.txt"`

Hinweis

Existiert bereits eine Protokolldatei gleichen Namens, dann wird diese ohne Rückfrage überschrieben. Die in der Protokolldatei bereits gespeicherten Daten sind dann verloren.

Am Ende des Skriptes, bzw. sobald Sie keine weiteren Ausgaben machen wollen, müssen Sie die Protokolldatei schließen.

Mit der Anweisung
`CLOSE-LOG`

wird die Protokolldatei geschlossen.

Hinweis

Falls die Protokolldatei vor dem Beenden eines Skripts nicht per Anweisung geschlossen wurde, werden die Ausgaben nicht in die Protokolldatei gespeichert. Die Protokolldatei ist in diesem Fall leer.

6.3.6.2 Unformatierte Ausgabe

Mit der Anweisung
```
PRINT "Signal"
```

wird der aktuelle Wert des Signals *Signal* in der geöffneten Protokolldatei gespeichert. Als Signale können Eingänge, Ausgänge, Zustände und online änderbare Parameter der Komponenten verwendet werden. Binärwerte werden als True bzw. False ausgegeben.

6.3.6.3 Formatierte Ausgabe

Mit der Anweisung
```
PRINTF "Formatstring", "Signal1", "Signal2", …
```

werden die aktuellen Werte der in der Anweisung gelisteten Signale in der geöffneten Protokolldatei gespeichert. Der *Formatstring* enthält die Formatangabe für jedes Signal, wobei für jedes Signal genau eine Formatangabe enthalten sein muss. Der Formatstring kann auch Text enthalten. Die erlaubten Formatanweisungen sind in der folgenden Tabelle gelistet.

Tabelle 6-3 Formatierungsanweisungen

Type des Signals	Formatierungsanweisung
analog, integer, binär	%f oder %.nf (n: Anzahl Nachkommastellen), Binärwerte werden als "0" bzw. "1" ausgegeben.
integer	%i
binär	%b (ergibt "True" bzw. "False")

Wenn Sie das Prozentzeichen als normales Zeichen ausgeben möchten, müssen Sie ihm im Formatstring einen Schrägstrich voranstellen: **\%**. Wollen Sie einen festen Text ausgeben, dann können Sie die *PRINTF*-Anweisung auch ohne Signalangaben verwenden.

Als Signale können Sie Eingangssignale, Ausgangssignale, sowie Zustände und online änderbare Parameter der Komponenten verwenden. Weitere Informationen hierzu finden Sie im Kapitel: Signale in Skripten (Seite 315).

Für das Beispiel
```
PRINTF "Rampe: %.2f [ULR=%b] [LLR=%b]", "Ramp#1/Y", "Ramp#1/ULR",
"Ramp#1/LLR"
```

wird folgendes der Protokolldatei ausgegeben:
```
Rampe: 0.00 [ULR=False] [LLR=True]
```

6.3.6.4 Ausgabe von Uhrzeit und Datum

Mit den in der folgenden Tabelle gelisteten Systemvariablen können Sie das Datum und die Uhrzeit Ihres Rechners in der Protokolldatei ausgeben:

Tabelle 6-4 Systemvariablen

Syntax	Bedeutung
_t_day	Tag
_t_mon	Monat
_t_year	Jahr
_t_hour	Stunde
_t_min	Minute
_t_sec	Sekunde

Alle Ausgaben erfolgen zweistellig.

Beispielsweise wird für die Anweisungen

```
PRINTF "Datum:   %.0f.%.0f.%.0f", "_t_day", "_t_mon", "_t_year"
PRINTF "Uhrzeit:  %.0f:%.0f:%.0f", "_t_hour", "_t_min", "_t_sec"
```

folgendes in der Protokolldatei eingetragen:

Datum: 20.06.11

Uhrzeit: 12:55:13

Hinweis

Die Systemvariablen in der Tabelle "Systemvariablen" sind nur im Rahmen der Protokollierung nutzbar und können nicht für allgemeine Berechnungen innerhalb des Skriptes genutzt werden.

6.3.6.5 Ausgabe von Versionsinformation

Mit den Systemvariablen *_ProjectVersion* und *_SIMITVersion* können Sie die Version Ihres Projekts und die Version von SIMIT in der Protokolldatei ausgeben.

Beispielsweise wird für die Anweisungen

PRINTF "Projektversion: %s", "_ProjectVersion"

PRINTF "SIMIT-Version: %s", "_SIMITVersion"

folgendes in der Protokolldatei eingetragen:

Projektversion: AA12345-344332-3.4

SIMIT-Version: 7.1.0

6.3.6.6 Die Systemfunktion _printlog

Es besteht auch die Möglichkeit, direkt aus Komponenten heraus in eine geöffnete Protokolldatei zu schreiben. Sie müssen dazu aber in die Verhaltensbeschreibung der entsprechenden Komponententypen modifizieren, d. h. die Systemfunktion

```
_printlog("string");
```

zur Ausgabe von Text einfügen. Diese Funktion schreibt den in *string* übergebenen Text beim Aufruf direkt in die geöffnete Protokolldatei.

Hinweis

Zum Editieren der Verhaltensbeschreibung von Komponententypen benötigen Sie den Komponententypeditor (CTE) von SIMIT. Der Komponententypeditor ist ein Erweiterungsmodul von SIMIT mit dem Sie eigene Komponententypen erstellen können.

6.3.7 Signalverläufe

6.3.7.1 Plot-Datei öffnen und schließen

Sie können im Skript Signale festlegen, deren Werte zyklisch in eine Plot-Datei geschrieben werden sollen. In einem Skript kann immer nur eine Plot-Datei geöffnet sein.

Mit der Anweisung
```
OPEN-PLOT "Dateiname"
```

öffnen Sie die Plot-Datei *Dateiname* und starten die Aufzeichnung der Signale. Der Name *Dateiname* der Datei ist mit absolutem Pfad anzugeben, wobei Trennstriche verdoppelt werden müssen.

Eine eventuell bereits geöffnete Plot-Datei wird zuvor geschlossen. Existiert die Datei *Dateiname* schon, wird sie ohne Rückfrage überschrieben.

Hinweis

Die aufzuzeichnenden Signale müssen bereits vor dem Öffnen der Datei festgelegt sein.

In der ersten Zeile der Plot-Datei stehen die Signalnamen der aufzuzeichnenden Signale. In den weiteren Zeilen steht je Zeile die Simulationszeit in Millisekunden und mit Tabulator getrennt die Werte der Signale in der in der ersten Zeile angegebenen Reihenfolge.

Mit der Anweisung
```
CLOSE-PLOT
```

wird die Plot-Datei geschlossen und die Liste der aufzuzeichnenden Signale gelöscht.

Für die Folge von Anweisungen
```
STOP
INIT
START
PLOT "Ramp#2/Y"
PLOT "Ramp#2/ULR"
PLOT-CYCLE 500
OPEN-PLOT "D:\\plot.txt"
GO-FOR 4000
CLOSE-PLOT
```

wird beispielsweise folgendes in der Plot-Datei eingetragen:

	Ramp#2/Y	Ramp#2/ULR
0	1.66666666666667	Ramp#2/ULR
500	18.3333333333333	False
1000	35.0	False
1500	51.6666666666667	False
2000	68.3333333333333	False
2500	85.0	False
3000	100.0	True
3500	100.0	True
4000	100.0	True

6.3.7.2 Signale festlegen

Mit der Anweisung
`PLOT "Signalname"`

fügen Sie das Signal *Signalname* zur Liste der aufzuzeichnenden Signale hinzu. Beachten Sie, dass die Liste der auszugebenden Signale bereits vor dem Öffnen der Plot-Datei vollständig aufgebaut sein muss.

Sie können Eingangssignale, Ausgangssignale, sowie Zustände und online änderbare Parameter der Komponenten Ihres Simulationsprojekts aufzeichnen.

6.3.7.3 Zyklus festlegen

Mit der Anweisung
`PLOT-CYCLE time`

legen Sie fest, in welchem Zyklus die Signale aufgezeichnet werden. Die in Millisekunden angegebene Zykluszeit *time* wird auf das nächste Vielfache der Zykluszeit des Skriptes aufgerundet. Wenn Sie diese Anweisung nicht in Ihr Skript einfügen, wird die Zykluszeit des Skriptes zur Aufzeichnung verwendet. Beachten Sie, dass der Zyklus vor dem Öffnen der Plot-Datei festgelegt sein muss.

6.3.8 Setzen von Signalen

Eingangssignale sowie Zustandsgrößen und online änderbare Parameter von Komponenten können Sie durch Anweisungen im Skript setzen. Sie können dazu diese Größen auf vorgegebene Werte setzen oder eine lineare Änderung der Werte vorgeben.

Hinweis

Wenn Sie Eingänge setzen, die im Simulationsprojekt mit Ausgängen verknüpft sind, oder Werte für Zustände einer Komponente vorgeben, die in der Komponente selbst gesetzt werden, wird der im Skript vorgegebene Wert entweder gar nicht oder nur während eines Rechenzyklus wirksam sein.

6.3.8.1 Einzelwerte setzen

Mit der Anweisung
`"Signal" = Wert`

können Sie das spezifizierte *Signal* auf den gegebenen *Wert* setzen.

Beispiel:
`"Status/1/BI" = True`

Anstelle eines konstanten Wertes können Sie auch einen Ausdruck für den Wert angeben. Ein Ausdruck besteht aus Konstanten oder Signalen, die durch Operatoren miteinander verknüpft sind. In der Tabelle "Arithmetische Operatoren" sind die zulässigen arithmetischen Operatoren gelistet, in der Tabelle "Boole'sche Operatoren" die zulässigen Boole'schen Operatoren.

Tabelle 6-5 Arithmetische Operatoren

Operator	Bedeutung
( )	Klammerung
* /	Multiplikation Division
+ –	Addition Subtraktion

Tabelle 6-6 Boole'sche Operatoren

Operator	Bedeutung
( )	Klammerung
!	nicht
&&	und
||	oder

Die Operatoren sind in beiden Tabellen in der Reihenfolge von höchster zu niedrigster Priorität eingetragen.

Beispiel:
`"Display/E" = 2 * ("Slider/A" + 0.5)`

Hinweis

Für ein Signal, dem Sie einen Wert zugewiesen haben, steht dieser Wert erst im nächsten Abarbeitungszyklus zur Verfügung. Wenn Sie also beispielsweise ein Signal durch eine Anweisung im Skript gesetzt haben und in der unmittelbar darauffolgenden Anweisung dieses Signal zur Wertzuweisung verwenden wollen oder seinen Wert ausgeben wollen, dann hat dieses Signal noch den Wert, den es zu Beginn des Zyklus hatte.

6.3.8.2 Verbundene Signale trennen

Mit der Anweisung
`FIX "Signal"`
können Sie das spezifizierte Signal trennen ("forcen"), um anschließend einen Wert setzen zu können.

Beispiel:
```
FIX "Status/1/BI"

"Status/1/BI"=True
```

Mit der Anweisung
`UNFIX "Signal"`
können Sie die Trennung des spezifizierten Signals wieder aufheben.

Beispiel:
```
UNFIX "Status/1/BI"
```

Die beiden Befehle FIX und UNFIX entsprechen der Bedienung des Signaltrenners an der Benutzeroberfläche.

6.3.8.3 Vorgeben eines Signalverlaufs

Mit den Anweisungen
```
RAMP "Signal" FROM start TO end IN time
RAMP "variable" TO end IN time
```

können Sie ein *analoges* Eingangssignal oder den Zustand einer Komponente auf einen linear ansteigenden oder abfallenden Wert setzen. Diese Anweisung bewirkt, dass der Wert *start* gesetzt wird und anschließend der Wert so erhöht oder erniedrigt wird, dass er in *time* Millisekunden den Wert *end* erreicht, er sich also mit einem linearen Verlauf (Rampe) ändert.

In der zweiten Anweisung ist kein Startwert (FROM start) enthalten. Startwert ist in diesem Fall der aktuelle Wert.

Die Abarbeitung des Skripts wird während der gegebenen Zeit *time* fortgesetzt, d. h. weitere Anweisungen im Skript werden ausgeführt ohne auf den Endwert für das Signal zu warten. Ein Skript ist erst beendet, wenn alle über Rampen gesetzte Signale ihren Endwert erreicht haben.

Beispiele:
```
RAMP "Display#1/X" FROM 20.0 TO 10.0 IN 15000
```

6.3.9 Bedingte Abarbeitung

Grundlagen

Mit folgenden Anweisungen machen Sie die Ausführung von Bedingungen abhängig:

```
IF expression THEN
    block1
ELSE
    block2
ENDIF
```

Wenn die Bedingung *expression* wahr ist, wird die Anweisungsliste *block1* ausgeführt, ansonsten die Liste *block2*. Die beiden Anweisungslisten bestehen aus einer oder mehreren Zeilen mit beliebigen Anweisungen, können also z. B. auch ihrerseits wieder bedingte Anweisungen enthalten.

Die *ELSE*-Anweisung kann auch entfallen:

```
IF expression THEN
    block
ENDIF
```

Jede IF-Anweisung müssen Sie mit ENDIF schließen.

Mit der Bedingung *expression* können Sie auf Eingangssignale, Ausgangssignale, sowie auf Zustände und online änderbare Parameter von Komponenten zugreifen. Folgende Vergleichsoperatoren werden unterstützt:

Tabelle 6-7 Vergleichsoperatoren

Operator	Bedeutung
<	Kleiner
<=	Kleiner oder gleich
>	Größer
>=	Größer oder gleich
==	Gleich
!=	Ungleich

Eine Bedingung kann aus mehreren Ausdrücken bestehen, die mit Boole'schen Operatoren verknüpft werden. Weitere Informationen zu den Boole'schen Operatoren finden Sie in den Tabellen im Kapitel: Einzelwerte setzen (Seite 325).

Beispiele

```
IF "Button/1/Z" == True THEN
    "Display/E" = 2.0 * ("Slider/A" + 0.5)
ELSE
    "Display/E" = "Slider/A"
ENDIF
```

```
IF "Button#1/Z" THEN
    "Display/E" = 2.0 * ("Slider/A" + 0.5)
ELSE
    IF !"Button#2/Z" THEN
        "Display/E" = 3.0 * ("Slider/A" + 0.5)
    ELSE
        "Display/E" = "Slider/A"
    ENDIF
ENDIF
```

Vergleichen von analogen Signalen

Die Werte von analogen Signalen werden im Gleitkommaformat abgebildet. Durch Rundungen können dabei unterschiedliche Werte entstehen, die beim Vergleich zu falschen Ergebnissen führen können. Verwenden Sie zum Wertevergleich mit analogen Signalen die Funktion `RANGECHECK`. Die Funktion hat folgende Syntax:

- RANGECHECK("Analogwert", "Sollwert", "Toleranz" | "Untere Toleranz", "Obere Toleranz")
  - Analogwert: Wert des analogen Signals, das mit dem "Sollwert" verglichen wird
  - Toleranz: Wert, um den der Analogwert vom "Sollwert" abweichen darf. Alternativ können Sie auch unterschiedliche Toleranzwerte für Unter- und Obergrenze angeben.

Die Funktion gibt True oder False zurück.

Im folgenden Beispiel wird "True" zurückgegeben, wenn der Wert des analogen Signals "TemperatureValue" zwischen "0,5" und "1" liegt.

```
IF RANGECHECK("TemperatureValue", 0.75, 0.25) == True THEN
    Block1
ELSE
    Block2
ENDIF
```

Im folgenden Beispiel wird "True" zurückgegeben, wenn der Wert des analogen Signals "TemperatureValue" zwischen "0,5" und "1,25" liegt.

```
IF RANGECHECK("TemperatureValue", 0.75, 0.25, 0.5) == True THEN
    Block1
ELSE
    Block2
ENDIF
```

6.3.10 Zugriff auf die Simulationszeit

Die Systemvariable
`_Time`

enthält die aktuelle Simulationszeit in Millisekunden.

Für das Beispiel
```
IF ("_Time" > 20000) THEN
    PRINTF "Simulationszeit: %.0fms", "_Time"
ENDIF
```

kann in der Protokolldatei dann folgende Ausgabe zu sehen sein:
```
Simulationszeit: 26600ms
```

Komponententypeditor 7

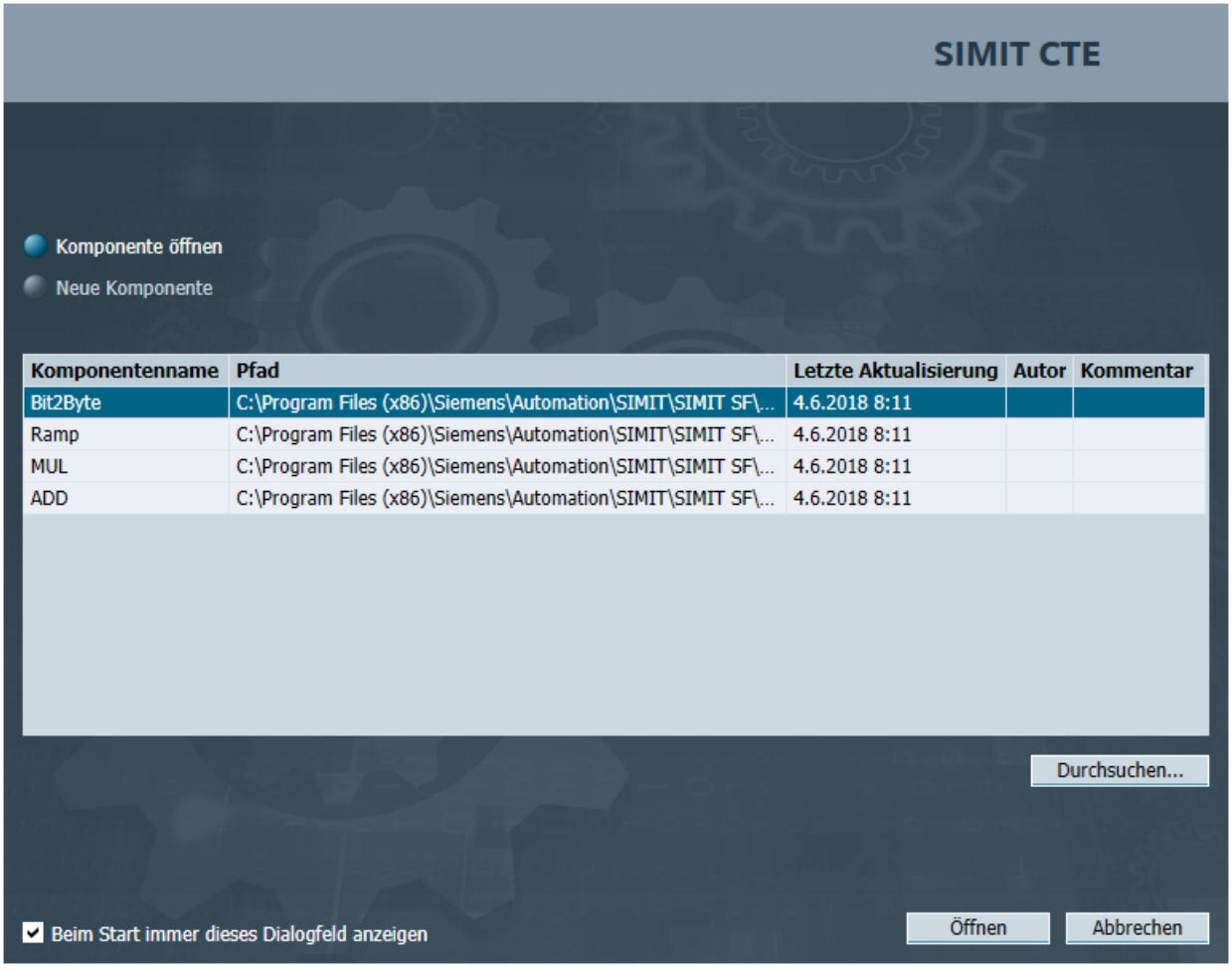
7.1 Die Benutzeroberfläche

7.1.1 Starten des CTE

Der Komponententypeditor (Component Type Editor, CTE) ist eine eigenständige Applikation von SIMIT. Sie starten ihn über das Startmenü im Ordner **Programme > Siemens Automation > SIMIT > SIMIT CTE**. Nach dem Start haben Sie die Auswahl, einen vorhandenen Komponententyp zu öffnen oder einen neuen Komponententyp anzulegen. Sie können diesen Dialog auch jederzeit über das Menü *Komponente* erreichen.

Hinweis

In der Benutzeroberfläche des CTE wird verkürzend der Begriff Komponente anstelle von Komponententyp verwendet.



Starten des CTE aus SIMIT oder über eine Komponentendatei

Sie können Komponententypen auch aus SIMIT heraus in der Task-Card *Komponenten* zum Bearbeiten öffnen. Markieren Sie dazu den gewünschten Komponententyp und führen Sie einen Doppelklick aus. Alternativ wählen Sie das Kommando *Öffnen* im Kontextmenü des zu öffnenden Komponententyps. Falls der CTE noch nicht gestartet war, startet er zum Öffnen des Komponententyps automatisch.

Komponententypen werden als eine Datei unter einem Namen mit der Endung *simcmp* im Dateisystem gespeichert. Sie können einen Komponententyp im CTE auch öffnen, indem Sie die Datei doppelklicken. Auch dann startet der CTE zum Öffnen des Komponententyps automatisch, falls er noch nicht gestartet war.

7.1.2 Gliederung der Benutzeroberfläche

Die Benutzeroberfläche des CTE folgt dem Oberflächenkonzept von SIMIT. Sie gliedert sich in die folgenden Paletten:

Die **Menüleiste** und die **Symbolleiste** ermöglichen einen einfachen Zugriff auf die Funktionen des CTE. Zusätzliche Funktionen stehen in Kontextmenüs zur Verfügung.

Im **Projektfenster** wird der geöffnete Komponententyp in einer Baumansicht dargestellt.

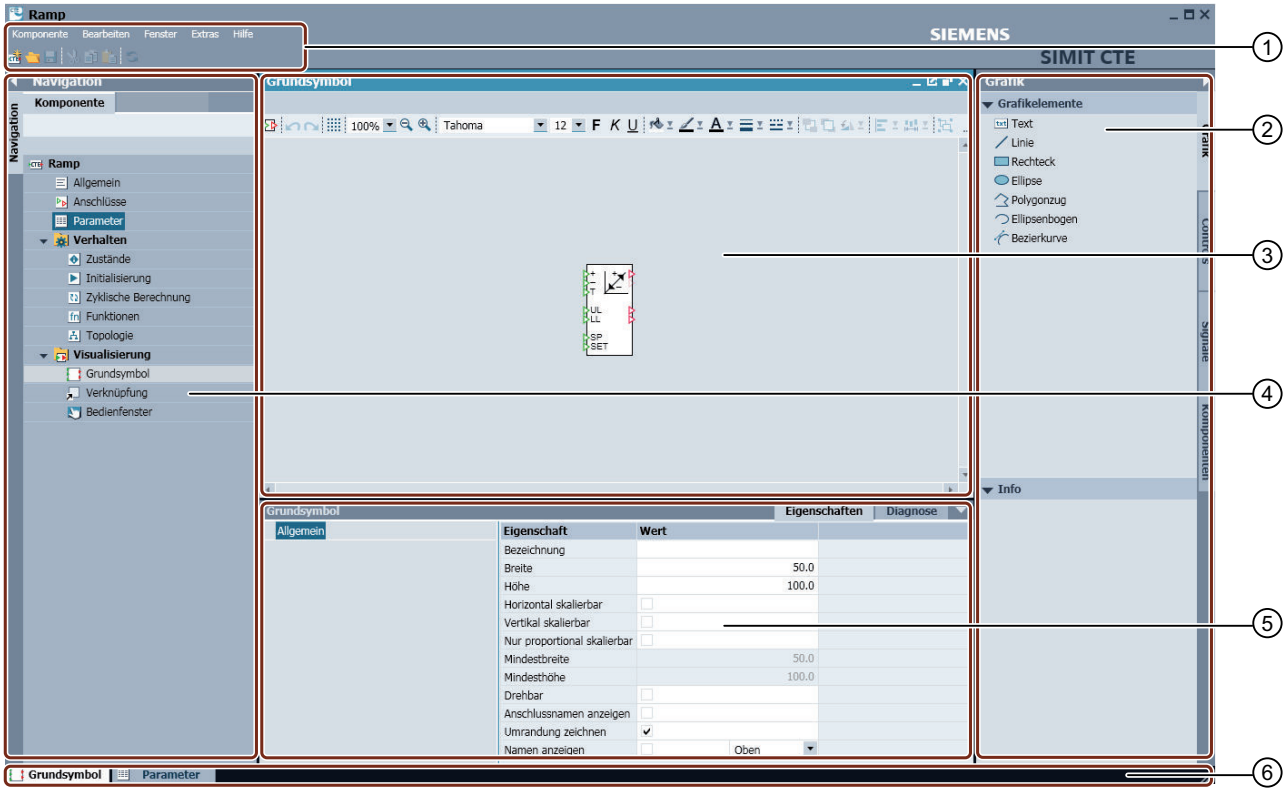
Im **Arbeitsbereich** werden die Editoren zum Bearbeiten geöffnet. Jeder Editor enthält eine Symbolleiste für den schnellen Zugriff auf die editorspezifischen Funktionen.

Im **Werkzeugfenster** finden Sie die mit dem jeweiligen Editor benutzbaren Werkzeuge wie Verbindungstypen und Grafikwerkzeuge geordnet in Task-Cards.

Im **Eigenschaftsfenster** werden die Eigenschaften eines im Arbeitsbereich selektierten Objektes dargestellt.

Über die **Editorleiste** links am unteren Rand der Oberfläche können Sie zwischen den geöffneten Editoren umschalten.

In der **Statuszeile** rechts am unteren Rand der Oberfläche werden Informationen zum aktuellen Status des CTE eingeblendet.



① Menüleiste und Symbolleiste
② Werkzeugfenster
③ Arbeitsbereich
④ Projektfenster
⑤ Eigenschaftsfenster
⑥ Editorleiste, Statuszeile

Alle Editoren werden im Arbeitsbereich geöffnet. Im Werkzeugfenster werden ausschließlich die für jeden Editor spezifischen Task-Cards zur Verfügung gestellt. Der Arbeitsbereich lässt sich über Menübefehle horizontal (*Fenster > Horizontal teilen*) oder vertikal (*Fenster > Vertikal teilen*) teilen um zwei Editoren nebeneinander oder untereinander im Arbeitsbereich öffnen zu können.

7.1.3 Die Menü- und Symbolleiste

In der Menüleiste des CTE finden Sie alle Befehle zum Anlegen bzw. Öffnen und Bearbeiten von Komponententypen.

Häufig benutzte Funktionen stehen auch in der Symbolleiste zur Verfügung. Im Einzelnen handelt es sich um folgende Funktionen:

(Neue Komponente) zum Erstellen eines neuen Komponententyps,

(Öffnen ...) zum Öffnen eines Komponententyps,

(Speichern) zum Speichern eines Komponententyps,

(Ausschneiden) zum Ausschneiden von selektierten Objekten,

(Kopieren) zum Kopieren von selektierten Objekten,

(Einfügen) zum Einfügen von kopierten Objekten,

(Aktualisieren) zum Aktualisieren des Komponententyps.

Verschiedene Aspekte eines Komponententyps beeinflussen sich wechselseitig. Die Funktion *Aktualisieren* sorgt dafür, dass alle Informationen in den unterschiedlichen Aspekten miteinander abgeglichen sind. Beim Aktualisieren wird außerdem noch geprüft, ob die Komponente formal korrekt beschrieben ist.

7.1.4 Der Projektbaum

Im Projektbaum werden alle Aspekte eines Komponententyps aufgelistet. Zu jedem Aspekt gehört ein entsprechender Editor, der durch Doppelklicken auf den entsprechenden Eintrag im Projektbaum geöffnet werden kann. Formale Fehler in der Implementierung eines Aspekts werden durch ein Overlay im Projektbaum gekennzeichnet: . Zusätzlich werden alle übergeordneten Ebenen dieses Aspekts mit diesem Overlay gekennzeichnet.



7.1.5 Tastaturkürzel

Sie können zum schnelleren Editieren eines Komponententyps die in folgender Tabelle gelisteten Tastaturkürzel benutzen. Alle Tastaturkürzel sind kontextabhängig, d. h. sie sind nur anwendbar, wenn der zugehörige Editor den Tastaturfokus besitzt.

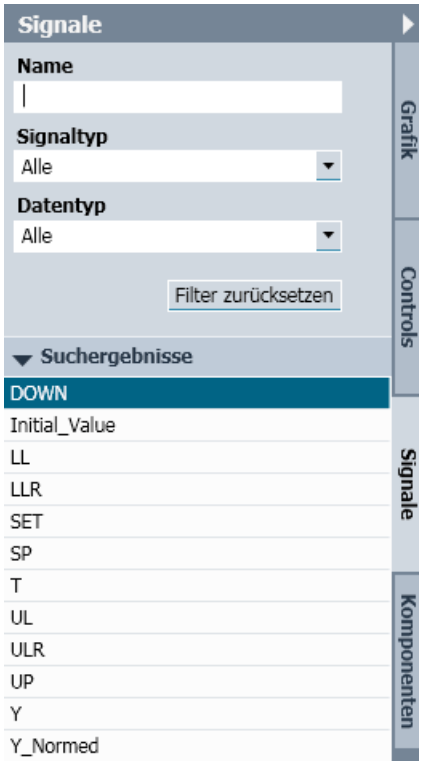
Tabelle 7-1 Tastaturkürzel

Tastenkürzel	Bedeutung
Strg+A	Alles selektieren
Strg+C	Kopieren
Strg+F	Suchen
Strg+H	Ersetzen
Strg+S	Speichern
Strg+V	Einfügen
Strg+X	Ausschneiden
F2	Umbenennen
F3	Weitersuchen
F5	Aktualisieren

7.1.6 Die Task-Cards

7.1.6.1 Die Task-Card "Signale"

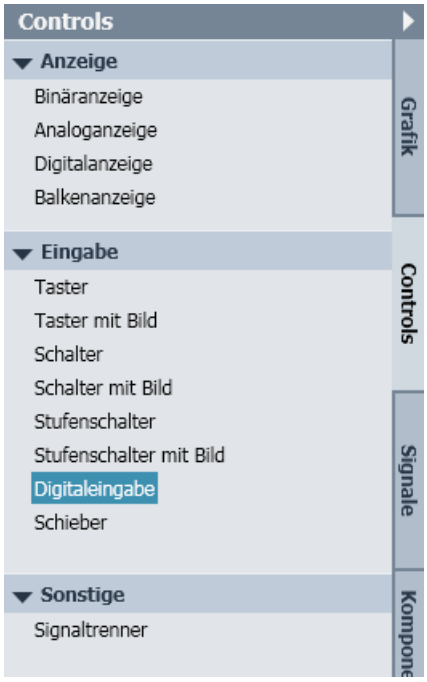
Zu jedem Texteditor der Verhaltensbeschreibung steht Ihnen die Task-Card "Signale" zur Verfügung. In ihr stehen alle in diesem Komponententyp vorhandenen Signale zur Verfügung. Da hier die "Quelle" von Signalen der Komponententyp selbst ist, ist für Signale hier nur der Name erforderlich.



Die Task-Card *Signale* erlaubt eine Filterung nach dem Namen, Signaltyp und Datentyp und bietet Ihnen somit einen schnellen Überblick über die in diesem Komponententyp vorhandenen Signale. Sie können einen Signalnamen auch per Drag & Drop aus der Task-Card in den Texteditor ziehen.

7.1.6.2 Die Task-Card "Controls"

Mit der Task-Card "Controls" stehen alle Controls der Basisbibliothek von SIMIT zur Verfügung.



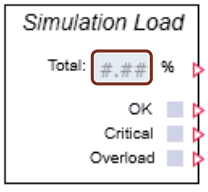
Es stehen drei Paletten von Controls zur Verfügung:

- Controls zur Anzeige von Signalwerten in der Palette *Anzeige*,
- Controls zur Eingabe von Signalwerten in der Palette *Eingabe* und
- weitere Controls in der Palette *Sonstige*

Controls sind unter ihrem Namen in der Bibliothek zu finden.

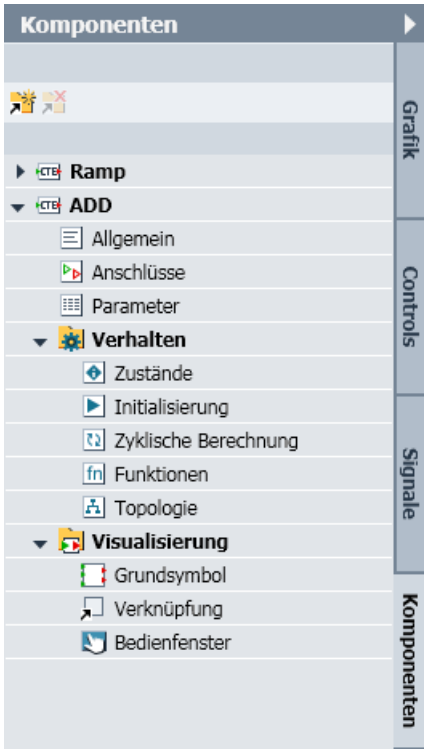
Sie können Controls auf das Grundsymbol legen und diese mit Ein- bzw. Ausgangsignalen verknüpfen.

Für Komponenten, deren Grundsymbole mit Controls entsprechend gestaltet sind, können Sie direkt auf dem Symbol Werte setzen und sich anzeigen lassen, ohne dass Sie erst das Bedienfenster der Komponente öffnen müssen.



7.1.6.3 Die Task-Card "Komponenten"

In der Task-Card "Komponenten" können Sie weitere Komponenten im Lesemodus öffnen.



Klicken Sie auf das Symbol , um eine Komponente zu öffnen und deren Eigenschaften anzuzeigen.

Klicken Sie auf das Symbol , um die markierte Komponente zu schließen.

Sie können einzelne Objekte einer Komponente im entsprechenden Editor öffnen und kopieren und in Objekte einer anderen Komponente einfügen. Sie können mehrere Komponenten gleichzeitig öffnen und damit eine Komponente aus Teilen von mehreren anderen Komponenten zusammenstellen.

Hinweis

Selbst erstellte Komponenten können passwortgeschützt werden. Beim Öffnen einer solchen Komponente erscheint dann eine Passwortabfrage. Einige Komponenten aus der Basisbibliothek sind von Haus aus geschützt und können nicht geöffnet werden. Weitere Informationen hierzu finden Sie im Abschnitt: Schutz des Komponententyps (Seite 345).

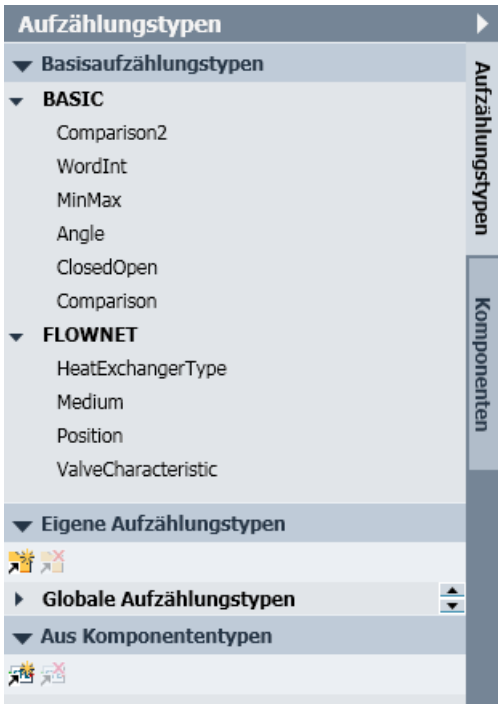
Weitere Informationen zu den Eigenschaften einer Komponente finden Sie im Abschnitt: Eigenschaften von Komponententypen (Seite 343).

Übertragen einer Komponente nach SIMIT

Wenn Sie eigene Komponententypen mit dem CTE erstellen, müssen Sie diese entweder in einem Bibliotheksverzeichnis oder unter *Globale Komponenten* abspeichern, damit sie Ihnen in SIMIT zur Verfügung stehen. SIMIT aktualisiert dazu die Palette der *Eigene Komponenten* automatisch, wenn Sie dort einen Komponententyp mit dem Komponententypeditor speichern.

7.1.6.4 Die Task-Card "Aufzählungstypen"

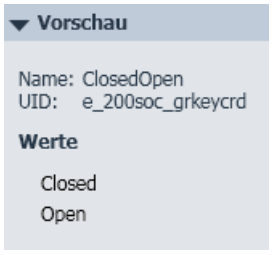
In der Task-Card "Aufzählungstypen" sind alle Aufzählungstypen aufgeführt, die Sie für Aufzählungsparameter verwenden können.



Die Task-Card ist in drei Paletten unterteilt:

- **Basisaufzählungstypen**
  In dieser Palette sehen Sie die Aufzählungstypen, die in den Komponenten der Basisbibliothek verwendet sind. Sie können diese Aufzählungstypen als Grundlage eigener Aufzählungstypen nutzen, indem Sie sie in die Palette *Eigene Aufzählungstypen* kopieren.
- **Eigene Aufzählungstypen**
  Sie können in dieser Palette eigene Aufzählungstypen anlegen. Kopieren Sie dazu einen bereits vorhandenen Aufzählungstypen oder klicken Sie auf den Befehl *Neuer Aufzählungstyp*. Es öffnet sich dann ein Fenster zur Definition des Aufzählungstyps. Geben Sie dort die Namen für die einzelnen Elemente der Aufzählung ein.
- **Aus Komponententypen**
  Sie können beliebige Komponententypen in dieser Palette der Task-Card mit dem Befehl öffnen. Die verwendeten Aufzählungstypen werden unter diesem Komponententyp und in der Auswahlliste angezeigt.

Wenn Sie einen Aufzählungstyp in einem dieser drei Paletten der Task-Card selektieren, wird Ihnen in der Vorschau im unteren Teil der Task-Card angezeigt, welche Elemente in diesem Aufzählungstyp enthalten sind.

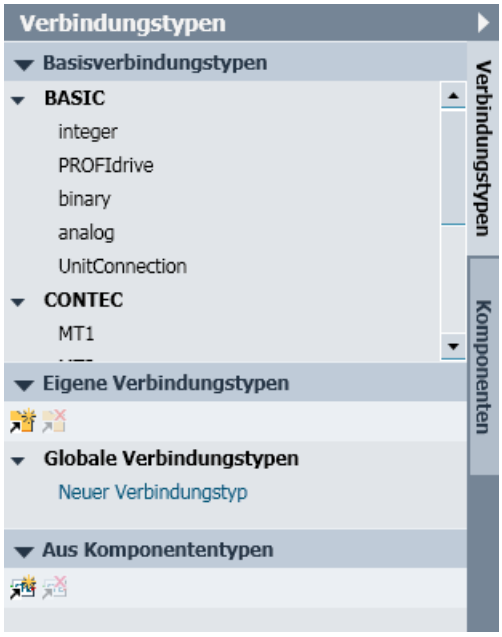


Weiter wird Ihnen zu einem Aufzählungstyp in der Vorschau auch seine eindeutige Kennung *UID* angezeigt.

In der Verhaltensbeschreibung verwenden Sie ein Element eines Aufzählungstyps durch die Angabe des Namens des Aufzählungstyps, gefolgt von einem Punkt und dem Namen des Elements selber. Das gesamte Konstrukt muss noch in einfache Anführungszeichen gestellt werden, also beispielsweise: `'ClosedOpen.Closed'`.

7.1.6.5 Die Task-Card "Verbindungstypen"

Die Task-Card "Verbindungstypen" listet alle in SIMIT bekannten Verbindungstypen und bietet die Möglichkeit, eigene Verbindungstypen zu definieren.



Sie ist in drei Paletten unterteilt:

Basisverbindungstypen

In dieser Palette sehen Sie die Verbindungstypen, die in den Komponententypen der Basisbibliothek verwendet sind. Das sind im Wesentlichen die Grundverbindungstypen wie in folgender Tabelle beschrieben.

Tabelle 7-2 Grundverbindungstypen

Verbindungstyp	Wert	Wertebereich
binary	Binärwerte	True/False
analog	Gleitkommawerte	$\pm 5.0 \times 10^{-324}$ bis $\pm 1.7 \times 10^{308}$
integer	Ganzzahlige Werte	−9.223.372.036.854.775.808 bis +9.223.372.036.854.775.807

Sie können die Basisverbindungstypen als Grundlage für Ihre eigenen Verbindungstypen nutzen. Kopieren Sie dazu den Verbindungstyp in die Palette *Eigene Verbindungstypen* und bearbeiten Sie ihn.

Eigene Verbindungstypen

In dieser Palette können Sie Ihre eigenen Verbindungstypen anlegen. Kopieren Sie dazu einen bereits vorhandenen Verbindungstypen oder klicken Sie auf den Eintrag *Neuer Verbindungstyp*. Es öffnet sich dann ein Fenster zum Definieren der Signale.

Sie können beliebig viele Signale in Richtung *Vorwärts* und *Rückwärts* anlegen. Für den Typ der Signale haben Sie nur die Auswahl zwischen den Datentypen *analog*, *binary* und *integer*.

Aktivieren Sie das Optionskästchen "Mehrfachverbindung erlauben", wenn ein Komponentenausgang mit mehreren Eingängen verschaltet werden soll. Für Verbindungstypen, die Rückwärtssignale enthalten, sind keine Mehrfachverbindungen zugelassen, da sonst mehrere Ausgangssignale auf den gleichen Eingang geführt würden. Die Option Mehrfachverbindung können Sie somit nur setzen, wenn keine Signale in Rückwärtsrichtung definiert sind.

Mit dem Schließen des Dialogs wird diesem Verbindungstyp eine eindeutige Kennung (UID) gegeben.

Aus Komponententypen

Wenn Sie Komponenten erstellen, deren Anschlüsse mit den Anschlüssen anderer Komponenten kompatibel sein sollen, ist es wichtig, dass Sie für diese Anschlüsse denselben Verbindungstyp benutzen. Sie können dazu beliebige Komponententypen in dieser Palette der Task-Card mit dem Befehl  öffnen. Die verwendeten Verbindungstypen werden dann unter diesem Komponententyp und in der Auswahlliste für Verbindungstypen angezeigt.

Wenn Sie einen Verbindungstypen in einer dieser drei Paletten der Task-Card selektieren, wird Ihnen in der Vorschau im unteren Teil der Task-Card aufgelistet, welche Signale mit diesem Verbindungstyp übertragen werden können:



Weiter wird Ihnen zu einem Verbindungstypen in der Vorschau auch seine eindeutige Kennung *UID* angezeigt. Beachten Sie, dass Verbindungstypen nur dann identisch sind, wenn sie die gleiche UID besitzen. Der Name des Verbindungstypen ist kein ausreichendes Kriterium. Auch Verbindungstypen, für die dieselben Signale definiert sind, müssen nicht identisch sein.

7.2 Grundlagen der Komponententypen

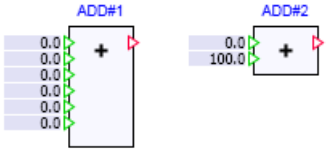
7.2.1 Das Typ-Instanz-Konzept von SIMIT

Einleitung

In SIMIT sind Komponenten die kleinsten Einheiten aus denen eine Simulation zusammengesetzt ist. Alle Komponenten sind Instanzen von Typen, die in Bibliotheken zur Verfügung gestellt werden. Komponententypen werden mit dem Komponententypeditor (CTE) erstellt und bearbeitet. Es werden dabei alle Aspekte berücksichtigt, die für Komponenten in SIMIT nutzbar sind.

Das Typ-Instanz-Konzept

Das funktionale Simulationsmodell setzt sich in SIMIT aus dem funktionalen Verhalten der einzelnen Komponenten zusammen, die grafisch auf Diagrammen abgelegt, parametriert und miteinander verschaltet werden. SIMIT folgt dabei einem Typ-Instanz-Konzept: Im Typ ist die parametrierbare Funktion definiert, in Diagramme werden individuell parametrierbare Instanzen des Typs eingefügt. Wir sprechen folglich hier auch von Komponententypen und Komponenten als Instanz.



Dieses Typ-Instanz-Konzept ermöglicht Ihnen, einen Komponententyp zu ändern ohne die bereits davon gebildeten Instanzen zu verändern.

Hinweis

Wenn Sie Änderungen an einem Komponententyp durchgeführt haben und die bereits in Ihrem Simulationsprojekt gebildeten Instanzen aktualisieren wollen, können Sie hierzu die Funktion *Suchen&Ersetzen* von SIMIT zum Ersetzen von Komponententypen verwenden.

Jede Komponenteninstanz wird in SIMIT durch einen eigenen, eindeutigen Namen identifiziert. Jede Instanz kann individuell parametriert werden und zwar bezüglich der eigentlichen Parameter sowie Vorbelegungen von Eingängen und der Skalierung des Symbols.

7.2.2 Eigenschaften von Komponententypen

Ein Komponententyp ist eine abgeschlossene Einheit, die mit dem Komponententypeditor erstellt und modifiziert werden kann. Technisch gesehen ist ein Komponententyp eine Datei mit der Dateiendung *.simcmp*. SIMIT-Bibliotheken sind somit lediglich Verzeichnisse in Ihrem Dateisystem, in denen Komponententypen zur Verwendung abgelegt sind.

In einem Komponententyp sind seine gesamten, in SIMIT nutzbaren Eigenschaften implementiert. Die Implementierung eines Komponententyps umfasst die folgenden Aspekte:

- **Allgemeine Angaben**
  Allgemeine Angaben beziehen sich auf die Verwaltung, den Schutz und Besonderheiten einer Komponente.
- **Anschlüsse**
  Anschlüsse sind alle sichtbaren und unsichtbaren Signaleingänge und -ausgänge eines Komponententyps. Im Komponententyp werden die Anschlüsse mit ihren Eigenschaften definiert.
- **Parameter**
  Parameter dienen der Individualisierung der einzelnen Komponenteninstanzen. Im Komponententyp wird festgelegt, welche seiner Eigenschaften in der Instanz parametrierbar sein sollen.
- **Verhalten**
  Die Definition von Zustandsgrößen sowie die funktionale Verhaltensbeschreibung legen fest, welches funktionale Verhalten eine Komponente hat. Im Detail werden damit die Abhängigkeiten der Ausgangssignale von den Eingangssignalen und Parametern definiert.
- **Visualisierung**
  Komponenten werden auf Diagrammen mit einem Grundsymbol grafisch dargestellt. Optional können Komponenten auch ein Symbol für eine Verknüpfung sowie ein Bedienfenster besitzen.

Jeder funktionsfähige, d. h. in SIMIT verwendbare Komponententyp erhält beim Abspeichern mit dem CTE automatisch eine eindeutige Kennzeichnung (ID).

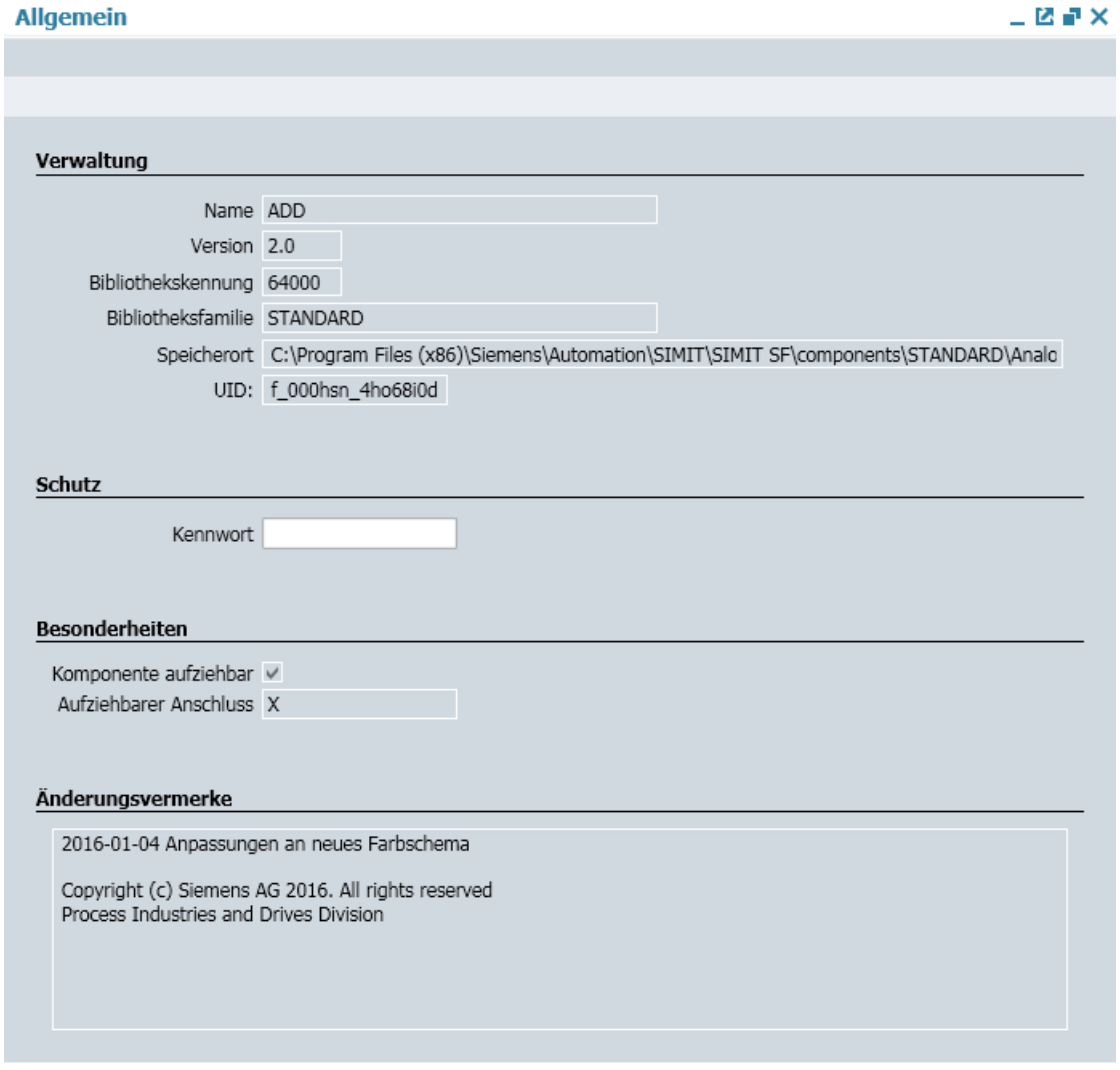
7.2.3 Die allgemeinen Eigenschaften der Komponententypen

7.2.3.1 Übersicht

Die allgemeinen Eigenschaften eines Komponententyps betreffen

- die Verwaltung,
- den Schutz,
- Besonderheiten und
- Änderungen

des Komponententyps. Zur Definition der allgemeinen Eigenschaften öffnen Sie den entsprechenden Editor durch Doppelklick auf den Aspekt *Allgemein* im Projektbaum. Die einzelnen Eigenschaften können Sie dann in einer fest vorgegebenen, gruppierten Anordnung von Eingabefeldern bearbeiten.



7.2.3.2 Die Verwaltungseigenschaften

Verwaltungsinformationen sind:

- Name des Komponententyps
- Version des Komponententyps
- Kennung der Bibliothek
- Familie der Bibliothek

Der *Name* kann frei gewählt werden. Mit ihm wird der Komponententyp in der Task-Card *Komponenten* von SIMIT angezeigt. Er ist damit auch Basis der automatischen Namensvergabe bei der Instanziierung des Komponententyps auf einem Diagramm. Der Name ist unabhängig davon, unter welchem Dateinamen der Komponententyp im Dateisystem gespeichert wird.

Die *Version* und *Bibliotheksfamilie* eines Komponententyps können Sie beliebig setzen.

Diese Angaben werden in der Vorschau der Task-Card *Komponenten* in SIMIT angezeigt, aber nicht weiter ausgewertet.

Wenn Sie den Namen oder die Version eines Komponententyps ändern, wird beim Speichern automatisch der Dialog zur Dateiauswahl geöffnet, damit Sie diese Komponente auch unter einem anderen Dateinamen speichern können. Der vorgeschlagene Dateiname entspricht dem Namen der Komponente.

Bibliotheken, die im Produktspektrum von SIMIT angeboten werden, besitzen eine definierte Bibliothekskennung. Die Bibliothekskennung ist in jedem Komponententyp einer Bibliothek eingetragen. Wenn Sie eigene Komponententypen erstellen, wird als Bibliothekskennung automatisch der Wert "0" eingetragen.

Der *Speicherort* des Komponententyps im Dateisystem und seine eindeutige Kennung *UID* werden Ihnen zur Information angezeigt.

7.2.3.3 Schutz des Komponententyps

Durch einen Kennwortschutz können Sie verhindern, dass der von Ihnen erstellte Komponententyp von Unbefugten im CTE geöffnet werden kann. Tragen Sie dazu ein beliebiges Kennwort ein. Zur Sicherheit wird das Kennwort ein zweites Mal abgefragt.

Beim Versuch, einen Komponententyp mit Kennwortschutz zu öffnen, erscheint eine Kennwortabfrage.

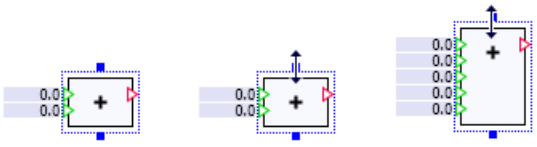
Der Kennwortschutz hat keinerlei Auswirkung auf die Verwendung eines Komponententyps in SIMIT. Er kann wie jeder andere Komponententyp auf ein Diagramm gezogen, instanziiert und verschaltet werden.

Hinweis

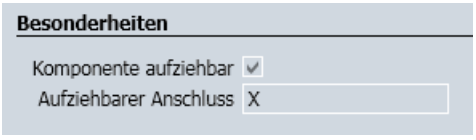
Bewahren Sie das Kennwort sicher auf. Ohne das richtige Kennwort können auch Sie selbst diesen Komponententyp nicht mehr im Komponententypeditor öffnen.

7.2.3.4 Besondere Eigenschaften

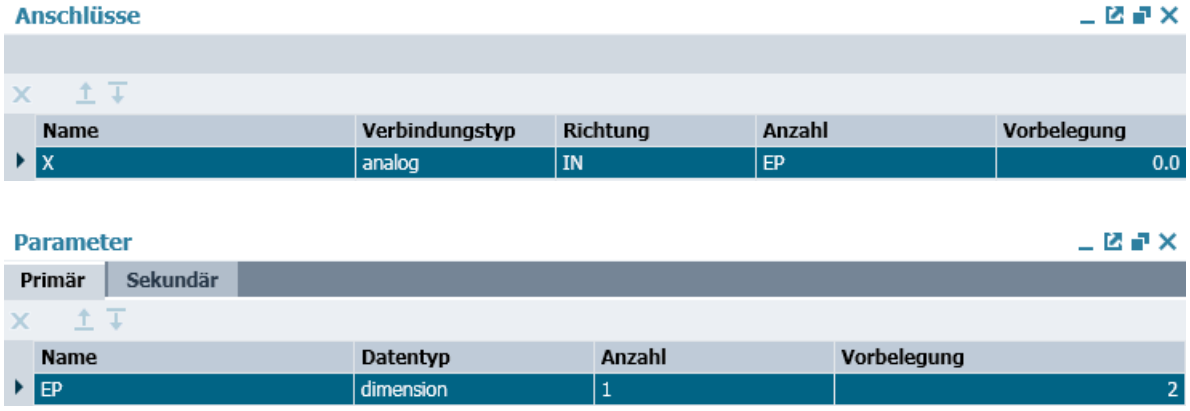
Einem Komponententyp kann die besondere allgemeine Eigenschaft "aufziehbar" gegeben werden. In diesem Fall bietet der Komponententyp genau einen definierten Anschluss (*Aufziehbarer Anschluss*) als Ein- oder Ausgang, der in jeder Instanz in unterschiedlich großer Anzahl gesetzt werden kann. Die Anzahl der Anschlüsse wird auf dem Diagramm gesetzt, indem das Symbol vertikal über seine Anfasser am Selektionsrahmen skaliert wird.



Diese Eigenschaft setzen Sie im Editor über die Option *Komponente aufziehbar* und geben dazu den aufziehbaren Anschluss an.



Der aufziehbare Anschluss muss als Vektor von Anschlüssen mit einer variablen Anzahl definiert sein, wobei die Anzahl als Parameter vom Typ *dimension* angelegt sein muss.



Ein Komponententyp kann neben dem aufziehbaren Anschluss auch weitere Anschlüsse haben. Der aufziehbare Anschluss muss aber an seinem Symbol immer unterhalb all der anderen Anschlüssen angeordnet werden.

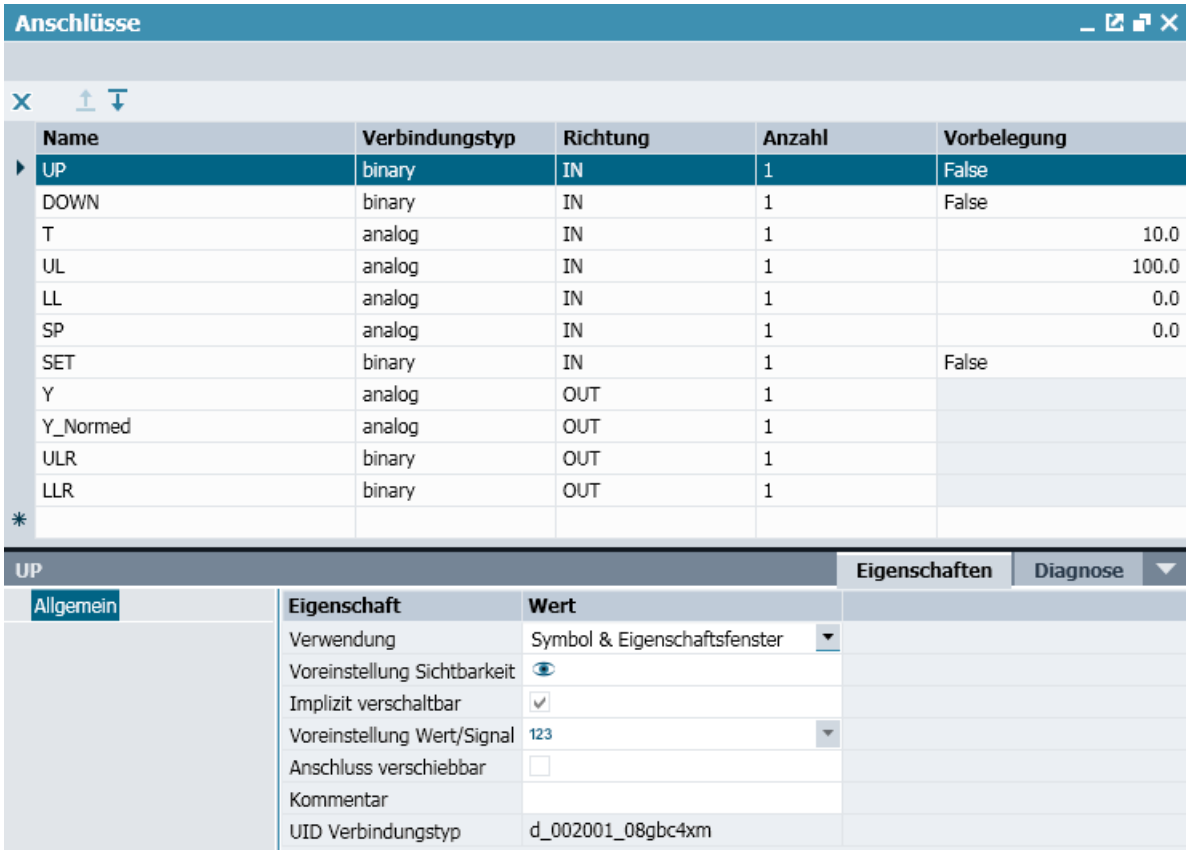
7.2.3.5 Änderungsvermerke

Die Änderungsvermerke im Komponententyp dienen lediglich der Dokumentation und werden von SIMIT nicht ausgewertet. Sie können hier z. B. die Änderungshistorie eines Komponententyps führen.

7.3 Die Anschlüsse und Parameter von Komponententypen

7.3.1 Der Anschlusseditor

Die Anschlüsse einer Komponente definieren in erster Linie die Schnittstelle, über die Informationen mit anderen Komponenten ausgetauscht werden können. Weiter werden über Anschlüsse auch Signale in ihr Bedienfenster eingekoppelt. Alle Anschlüsse eines Komponententyps werden im Anschlusseditor bearbeitet, der als Tabelleneditor angelegt ist. Öffnen Sie den Anschlusseditor mit einem Doppelklick auf den Menüpunkt "Anschlüsse" im Projektbaum. Folgende Abbildung zeigt den Anschlusseditor:



Jeder Anschluss wird durch folgende Eigenschaften gekennzeichnet:

- **Name**
  Jeder Anschluss muss einen eindeutigen Namen besitzen. Der Name darf nur Buchstaben, Ziffern und den Unterstrich enthalten und muss mit einem Buchstaben beginnen. Es wird zwischen Groß- und Kleinschreibung unterschieden. Auf den Namen eines Anschlusses wird z. B. in der Verhaltensbeschreibung Bezug genommen.
- **Verbindungstyp**
  Alle Anschlüsse in SIMIT sind typisiert, d. h. im Verbindungstyp ist genau festgelegt, welche Information über einen Anschluss dieses Typs ausgetauscht werden kann. Anschlüsse müssen immer vom gleichen Typ sein, damit sie auf einem Diagramm miteinander verbunden werden können. Die zur Verfügung stehenden Verbindungstypen werden Ihnen in einem Auswahlfeld angeboten.
- **Richtung**
  Über die Richtung legen Sie fest, ob der Anschluss in Richtung IN oder OUT definiert ist. Binäre, ganzzahlige (integer) und analoge Anschlüsse sind damit als Ein- oder Ausgang festgelegt.
  Der Sonderfall eines Anschlusses ohne Richtung (NONE) ist nur im Zusammenhang mit speziellen Bibliotheken relevant. Näheres entnehmen Sie den Handbüchern zu diesen Bibliotheken.
- **Anzahl**
  Haben Sie hier einen anderen Wert als die voreingestellte Anzahl eins eingegeben, dann haben Sie einen Anschlussvektor mit der gegebenen Anzahl an Elementen definiert. In der Komponenteninstanz werden diese Anschlüsse einfach durchnummeriert indem an den Namen die mit eins beginnende Indexnummer angehängt wird.
  Sie können als Anzahl auch einen Parameter eintragen, der die Anzahl der Anschlüsse bestimmt. Dieser Parameter muss dann vom Typ *dimension* sein.
- **Vorbelegung**
  Anschlüsse, die als Eingang definiert sind, können mit einem Zahlenwert vorbelegt werden. Diese Vorbelegung kann in jeder Komponenteninstanz überschrieben werden.

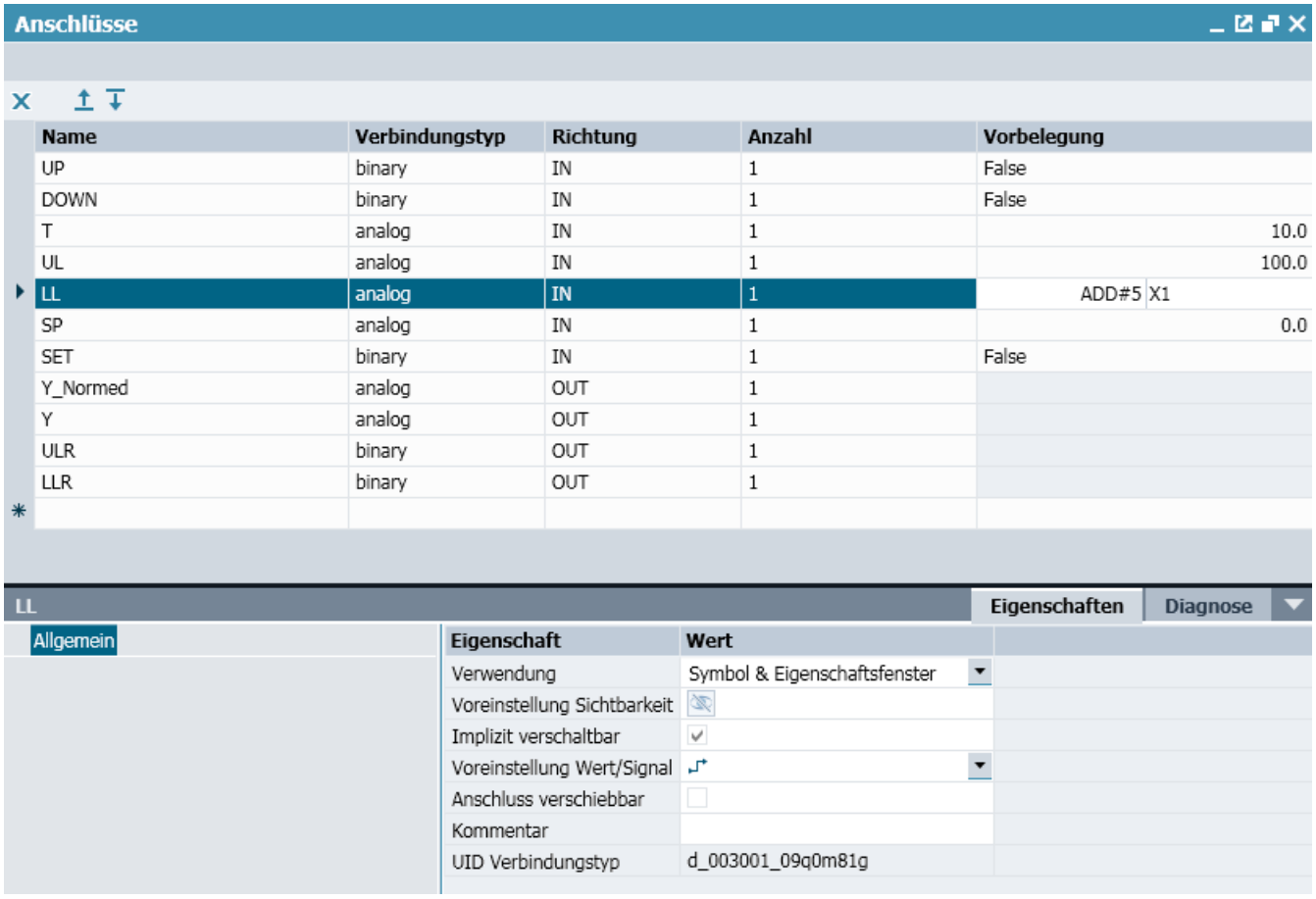
Darüber hinaus besitzen Anschlüsse weitere Eigenschaften, die im Eigenschaftsfenster für jeden Anschluss definiert werden können:

- **Verwendung**
  Ein Anschluss kann unterschiedlich verwendet werden. In der Regel soll der Anschluss am Symbol der Komponente auf einem Diagramm sichtbar sein und so die Verschaltung mit anderen Komponenten ermöglichen. Stellen Sie dafür die Verwendung auf *Symbol und Eigenschaftsfenster*.
  Falls der Anschluss nur im Eigenschaftsfenster einer Komponente, aber nicht am Komponentensymbol auf dem Diagramm sichtbar sein soll, stellen Sie die Verwendung auf *Nur im Eigenschaftsfenster*. Im Eigenschaftsfenster der Komponente ist dieser Anschluss dann durch das Symbol unveränderbar als unsichtbarer Anschluss gekennzeichnet.
  Die Einstellung *Ausschließlich im CTE* bewirkt, dass der Anschluss im Komponententyp verwendet werden kann, aber weder am Symbol noch im Eigenschaftsfenster der Komponente sichtbar ist.
- **Voreinstellung Sichtbarkeit**
  Wenn ein Anschluss die Verwendung *Symbol und Eigenschaftsfenster* hat, können Sie einstellen, ob er nach dem Instanziieren auf einem Diagramm zunächst sichtbar ( ) oder unsichtbar ( ) ist.

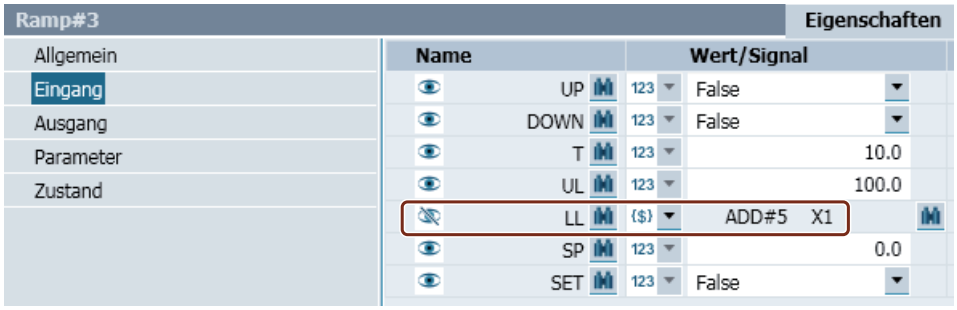
- **Implizit verschaltbar (Nur für Eingangssignale)**
  Alle Anschlüsse, die die Verwendung *Symbol und Eigenschaftsfenster* haben, können im Eigenschaftsfenster der Komponenteninstanz auch implizit verschaltet werden. Die Eigenschaft *Implizit verschaltbar* ist für diese Verwendung fest gesetzt.
  Für Anschlüsse mit der Verwendung *Nur im Eigenschaftsfenster* können Sie hier einstellen, ob es sich um einen implizit verschaltbaren Anschluss handeln soll oder nicht. Wenn Sie einen Anschluss mit dieser Verwendung nicht implizit verschaltbar einstellen, dann kann im Eigenschaftsfenster der Komponente lediglich die Vorbelegung dieses Anschlusses überschrieben werden. Wenn Sie implizit verschaltbar einstellen, dann wird die Voreinstellung für Wert/Signal auf Signal umgestellt.
- **Voreinstellung Wert/Signal (Nur für Eingangssignale)**
  Wenn die Verwendung des Anschlusses auf *Symbol und Eigenschaftsfenster* steht, können Sie wählen, ob der Anschluss in der Komponenteninstanz auf *Wert* (123) oder *Signal* ( ) voreingestellt ist.
  Ist die Voreinstellung *Signal*, so wird die Voreinstellung der Sichtbarkeit automatisch auf *nicht sichtbar* gestellt.
- **Anschluss verschiebbar**
  Sie können festlegen, ob der Anschluss in der Komponenteninstanz auf dem äußeren Rand der Komponente verschoben werden darf oder nicht. Zum Verschieben des Anschlusses mit der Maus ist die "Alt"-Taste gedrückt zu halten.
- **Kommentar**
  Der Kommentar zu einem Anschluss dient lediglich der Dokumentation und wird von SIMIT nicht ausgewertet.
- **Verbindungstyp ID**
  Da der Name eines Verbindungstyps nicht eindeutig sein muss, können Sie ihn im Zweifelsfall anhand der eindeutigen ID identifizieren.

7.3.2 Spezielle Vorbelegung implizit verschaltbarer Eingänge

Eingänge sind normalerweise mit einem Zahlenwert bei analogen und ganzzahligen (Integer) Eingängen oder dem Wert True/False bei binären Eingängen vorbelegt. Wenn die Voreinstellung für *Wert/Signal* auf *Signal* ( ) gestellt wird, gibt es aber noch eine weitere Option für die Vorbelegung: Sie können jetzt einen Signalnamen vorbelegen.



Für eine Instanz der Komponente ist dann für diesen Eingang das Symbol {$} mit dem vorgegebenen Signalnamen im Eigenschaftsfenster gesetzt. Dieser Eingang der Komponente ist damit fest mit dem Ausgang einer anderen Komponente verschaltet. Bei dem in der Abbildung oben dargestellten Beispielsignal ist das der Ausgang *X1* einer Komponente mit dem Namen *ADD#5*.

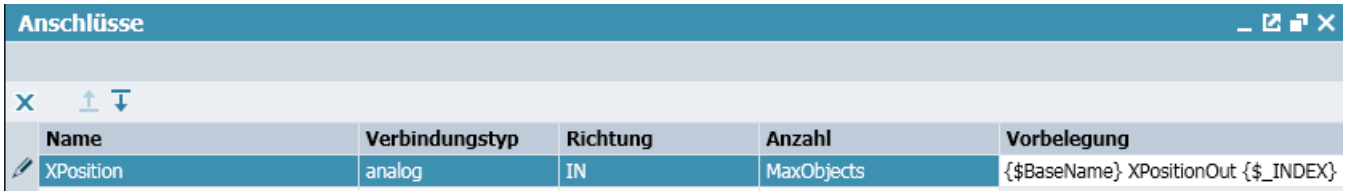


Anstelle von festen Namen für das Signal können Sie auch Parameter oder den Instanznamen der Komponente verwenden. Schreiben Sie dazu den Parameternamen oder *_NAME* für den Instanznamen in geschweiften Klammern mit vorangestelltem $-Zeichen für die Quelle und/ oder den Anschluss des Signals:

- `{$Parametername}` bzw.
- `{$_NAME}`.

Der Parametername und *_NAME* sind folglich lediglich Platzhalter im Komponententyp für die in der Komponenteninstanz zugewiesenen Werte Parameterwert oder Instanzname. Im Übrigen können Sie den Signalnamen auch beliebig aus Platzhaltern und festen Namen zusammensetzen.

Mit Hilfe der Systemvariablen *_INDEX* können Sie implizite Verbindungen für die einzelnen Elemente von Vektoren definieren. Benutzen Sie {$_INDEX}, wie in folgender Abbildung z. B. zu sehen ist. Bei der Instanziierung der Komponente wird dann dieser Ausdruck in jedem Element durch den Index des Elements ersetzt, wobei die Indexzählung mit eins beginnt.



Wenn für eine Komponente nach diesem Beispiel die Werte

- Parameter MaxObjects ist 2 und
- Paramter BaseName ist "LifterBase#1"

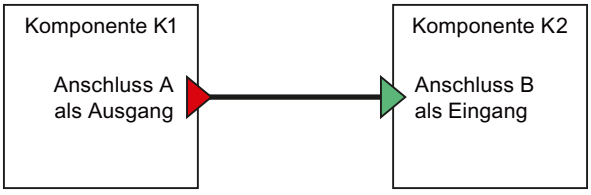
eingestellt werden, dann ergibt sich für den Eingangsvektor XPosition folgende Vorbelegung:

- XPosition1: LifterBase#1 XPositionOut1
- XPosition2: LifterBase#1 XPositionOut2

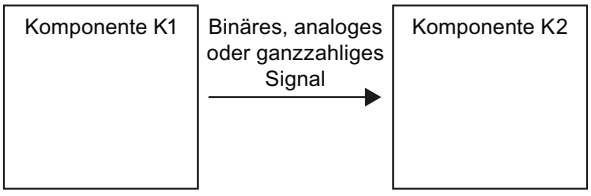
7.3.3 Komplexe Verbindungstypen

Im grundlegenden Verbindungskonzept wird ein einziges analoges, ganzzahliges (integer) oder binäres Signal zwischen verbundenen Anschlüssen von Komponenten übertragen. Die Signalverbindung ist immer von Ausgang zu Eingang gerichtet, d. h. ihre Richtung ergibt sich implizit aus der Art des Anschlusses. In folgender Abbildung ist dieses Konzept für die Verbindung eines Ausgangs mit einem Eingang skizziert.

Verbindungssicht:



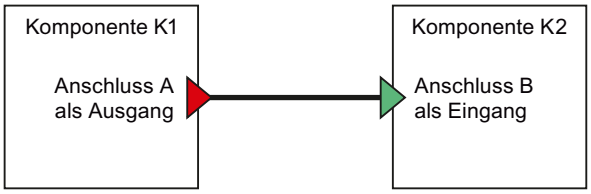
Datentechnische Sicht:



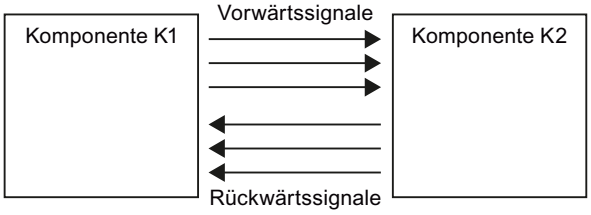
Verbindungen dieses Typs werden in SIMIT als Grundverbindungstypen zur Verfügung gestellt. In der Auswahlmaske werden Ihnen diese Typen als *analog*, *integer* und *binary* angeboten.

Das Verbindungskonzept von SIMIT ist gegenüber diesem Basiskonzept erweitert: Über eine Verbindung können mehrere Signale zwischen Anschlüssen in beide Richtungen übertragen werden. Die Richtung eines Signals kann somit nicht mehr aus den verbundenen Anschlüssen abgeleitet werden, sie ist im Verbindungstyp als *Vorwärts-* oder *Rückwärtssignal* zu definieren: Vorwärtssignale werden von einem Ausgang zu einem Eingang übertragen, Rückwärtssignale genau umgekehrt. In folgender Abbildung ist das Schema einer solchen komplexen Verbindung skizziert.

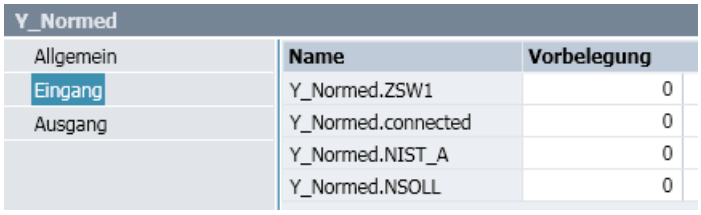
Verbindungssicht:



Datentechnische Sicht:

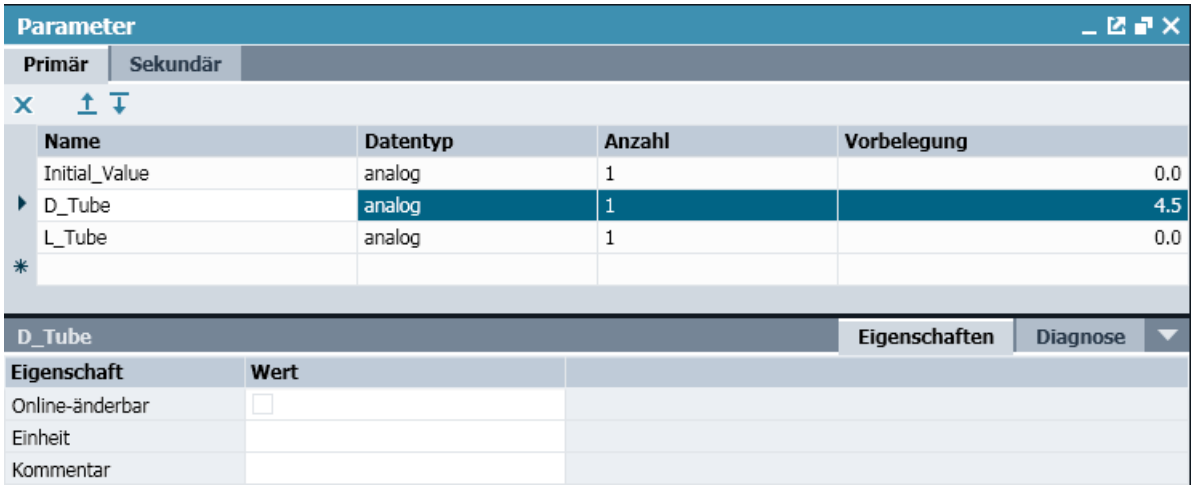


Für einen Anschluss eines komplexen Verbindungstyps können sich somit sowohl Ein- wie auch Ausgangssignale ergeben. Diese Signale werden im Eigenschaftsfenster des Anschlusses aufgelistet Sie können dort die Eingänge individuell vorbelegen.



7.3.4 Die Parameter von Komponententypen

Komponenten können durch Parameter individuell eingestellt werden. Dazu müssen im Komponententyp die entsprechenden Parameter vorgesehen werden. Zur Definition der Parameter öffnen Sie den Parametereditor durch Doppelklick auf den Aspekt *Parameter* im Projektbaum.



Teilen Sie die Parameter in die Registerkarten "Primär" und "Sekundär" ein. Damit trennen sie die Parameter in "wichtig" und "weniger wichtig" auf. Wichtige Parameter werden z. B. zum Parametrieren der Komponente verwendet. Wenn Sie hier Sekundärparameter definieren, wird SIMIT diese Unterscheidung im Eigenschaftsfenster der Komponenteninstanz berücksichtigen. Es wird dann im Eigenschaftsfenster zusätzlich zur Kategorie *Parameter* eine weitere Kategorie *Zusatzparameter* angezeigt, welche die Sekundärparameter enthält.

Parameter werden durch folgende Eigenschaften gekennzeichnet:

- **Name**
  Jeder Parameter muss einen eindeutigen Namen besitzen. Der Name darf nur Buchstaben, Ziffern und den Unterstrich enthalten und muss mit einem Buchstaben beginnen. Es wird zwischen Groß- und Kleinschreibung unterschieden.
- **Datentyp**
  Parameter können einen der in folgender Tabelle dargestellten Datentypen besitzen. Zusätzlich stehen für Parameter alle Aufzählungstypen zur Verfügung. Die Aufzählungstypen sind im Abschnitt "Aufzählungstypen definieren (Seite 339)" detailliert beschrieben.

Tabelle 7-3 Datentypen für Parameter

Datentyp	Bedeutung	Wertebereich
binary	Binärwerte	True/False
analog	Gleitkommawerte	$\pm 5.0 \times 10^{-324}$ bis $\pm 1.7 \times 10^{308}$
integer	Ganzzahlige Werte	−9.223.372.036.854.775.808 bis +9.223.372.036.854.775.807
dimension	Anzahl eines Anschluss- oder Parametervektors	1 .. 256

Datentyp	Bedeutung	Wertebereich
text	Einzeiliger Text	
characteristic	Kennlinie	

- **Anzahl**
  Haben Sie hier einen anderen Wert als die voreingestellte Anzahl eins eingegeben, dann haben Sie einen Parametervektor mit der gegebenen Anzahl an Elementen definiert. In der Komponenteninstanz werden diese Parameter einfach durchnummeriert indem an den Namen die mit eins beginnende Indexnummer angehängt wird.
  Sie können als Anzahl auch einen weiteren Parameter eintragen, der die Anzahl dieser Parameter bestimmt. Dieser Parameter muss dann vom Typ *dimension* sein.
- **Vorbelegung**
  Parameter können mit einem Zahlenwert vorbelegt werden.

Darüber hinaus besitzen Parameter weitere Eigenschaften, die Sie im Eigenschaftsfenster des Parameters bearbeiten können:

- **Online-änderbar**
  Online änderbare Parameter sind Parameter, die für eine Komponenteninstanz bei laufender Simulation geändert werden können.
  Parameter vom Typ *dimension* sind nicht online änderbar.
- **Einheit**
  Die hier eingetragene Einheit erscheint lediglich als zusätzliche Eigenschaft des Parameters im Eigenschaftsfenster der Komponenteninstanz.
- **Kommentar**
  Der Kommentar zu einem Parameter dient lediglich der Dokumentation und wird von SIMIT nicht ausgewertet.

Alle Namen von Parametern und Anschlüsse müssen eindeutig sein, d. h. ein Anschluss darf nicht den gleichen Namen wie ein Parameter haben und umgekehrt.

7.4 Das Verhalten von Komponententypen

7.4.1 Einleitung

Das funktionale Verhalten eines Komponententyps wird durch Zustandsgrößen und die Verhaltensbeschreibung definiert. Die Verhaltensbeschreibung wird unterteilt in die folgenden Aspekte:

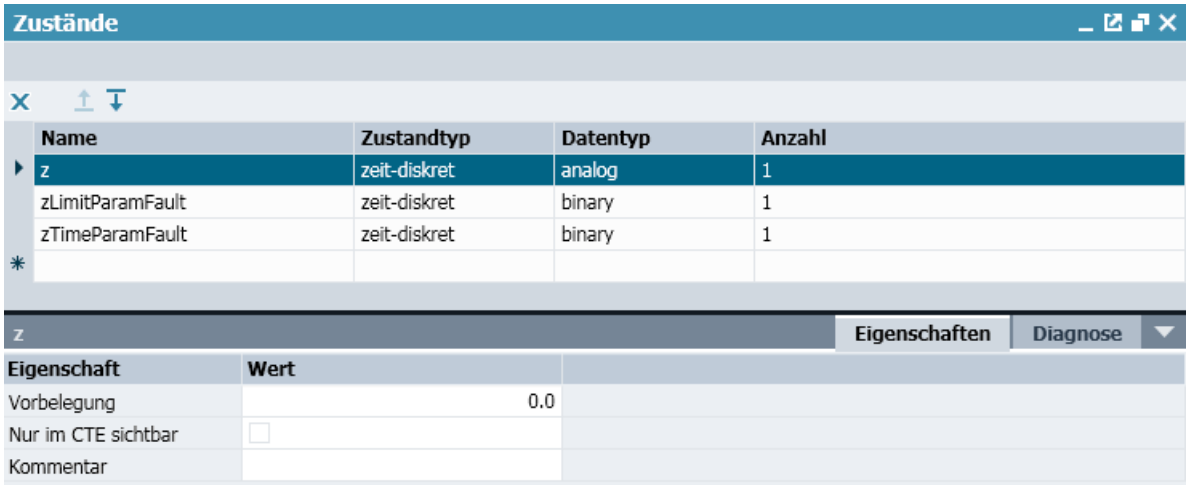
- Initialisierung
- zyklische Berechnung
- Funktionen

Der Komponententypeditor stellt für alle Aspekte geeignete Editoren zur Verfügung: Einen Tabelleneditor für die Zustände und einen Texteditor für die Teilaspekte der Verhaltensbeschreibung.

7.4.2 Die Zustandsgrößen

Zustandsgrößen einer Komponente sind sozusagen das Gedächtnis einer Komponente. Sie enthalten Werte, die zu einem beliebigen Zeitpunkt nicht alleine aus den Eingangsgrößen und Parametern berechnet werden können, sondern davon abhängen, was in der Vergangenheit geschehen ist. Die Füllung eines Behälters ist z. B. nicht berechenbar, wenn man nur die Zu- und Abflüsse in einem Zeitpunkt bilanziert. Sie hängt auch davon ab, welchen Inhalt der Behälter schon vor dem betrachteten Zeitpunkt hatte.

Zum Editieren der Zustände wird Ihnen ein Tabelleneditor zur Verfügung gestellt, den Sie durch Doppelklick auf den Aspekt *Zustände* im Projektbaum öffnen können.



Ein Zustand wird grundsätzlich durch folgende Eigenschaften gekennzeichnet:

- **Name**
  Jede Zustandsgröße muss einen eindeutigen Namen besitzen. Der Name darf nur Buchstaben, Ziffern und den Unterstrich enthalten und muss mit einem Buchstaben beginnen. Es wird zwischen Groß- und Kleinschreibung unterschieden.
- **Zustandstyp**
  Es gibt die beiden unterschiedlichen Zustandstypen *zeit-diskret* und *kontinuierlich*. Der Unterschied besteht darin, wie der jeweils neue Wert eines Zustands berechnet wird: Für zeit-diskrete Zustände wird der Wert in jedem Bearbeitungszyklus durch eine Berechnungsvorschrift in Form einer expliziten Gleichung berechnet, die Sie in der Verhaltensbeschreibung entsprechend festlegen. Die Berechnungsvorschrift für eine kontinuierliche Zustandsgröße legen Sie durch eine Differentialgleichung fest. Durch Lösung dieser Differentialgleichung mit Hilfe eines geeigneten numerischen Lösungsverfahrens berechnet SIMIT die Zustandswerte in jedem Bearbeitungszyklus. Weitere Informationen wie Sie zeit-diskrete und kontinuierliche Zustandsgrößen mit Hilfe von expliziten Gleichungen und Differentialgleichung in der Verhaltensbeschreibung behandeln, entnehmen Sie dem entsprechenden Abschnitt im Kapitel: Der gleichungsorientierte Ansatz (Seite 370).
- **Datentyp**
  Zeit-diskrete Zustandsgrößen können jeden der in folgender Tabelle gelisteten Datentypen besitzen.
  Kontinuierliche Zustandsgrößen sind immer vom Typ *analog*.

Tabelle 7-4 Datentypen für zeit-diskrete Zustände

Datentyp	Bedeutung	Wertebereich
binary	Binärwerte	True/False
analog	Gleitkommawerte	$\pm 5.0 \times 10^{-324}$ bis $\pm 1.7 \times 10^{308}$
integer	Ganzzahlige Werte	−9.223.372.036.854.775.808 bis +9.223.372.036.854.775.807
byte	Byte	0 bis 255

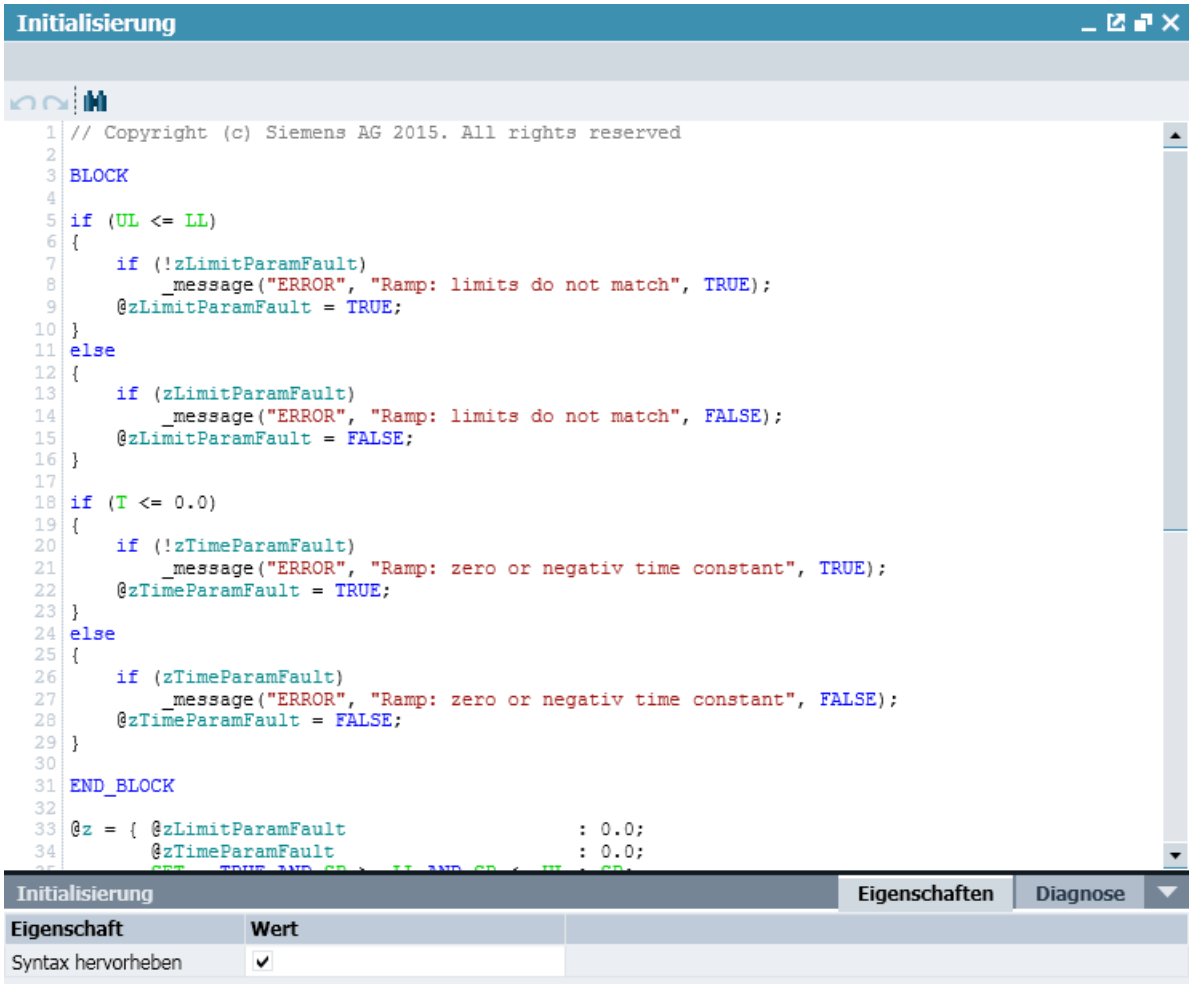
- **Anzahl**
  Haben Sie hier einen anderen Wert als die voreingestellte Anzahl eins eingegeben, dann haben Sie einen Zustandsvektor mit der gegebenen Anzahl an Elementen definiert. In der Komponenteninstanz werden diese Zustände einfach durchnummeriert indem an den Namen die mit eins beginnende Indexnummer angehängt wird.
  Sie können als Anzahl auch einen weiteren Parameter eintragen, der die Anzahl dieser Parameter bestimmt. Dieser Parameter muss dann vom Typ *dimension* sein.

Darüber hinaus besitzen Zustandsgrößen weitere Eigenschaften, die im Eigenschaftsfenster des Komponententyps bearbeitet werden können:

- **Vorbelegung**
  Jede Zustandsgröße hat eine zu ihrem Typ passende Vorbelegung. Diese Vorbelegung kann in der Komponenteninstanz überschrieben werden.
- **Nur im CTE sichtbar**
  Wenn Sie nicht möchten, dass dieser Zustand im Eigenschaftsfenster der Komponente sichtbar ist, setzen Sie diese Option.
- **Kommentar**
  Der Kommentar zu einer Zustandsgröße dient der Dokumentation und wird von SIMIT nicht ausgewertet.

7.4.3 Initialisierung, zyklische Berechnung und Funktionen

Die Verhaltensbeschreibung einer Komponente besteht aus einem einmalig während der Initialisierung ausgeführten Teil und einem in jedem zyklischen Berechnungsschritt ausgeführten Teil. Optional können mehrfach benutzte Berechnungen als Funktionen definiert werden. Der Komponententypeditor bietet Ihnen für diese drei Teilaspekte der Verhaltensbeschreibung jeweils einen Texteditor. Den Editor für einen dieser Teilaspekte öffnen Sie durch Doppelklick auf den entsprechenden Teilaspekt im Projektbaum.



Zur besseren Orientierung können Sie im Eigenschaftsfenster die Option *Syntax hervorheben* aktivieren. Wesentliche Elemente der Beschreibungssyntax werden dann durch verschieden farbige Schreibung leichter erfassbar. Die folgende Tabelle gibt die für die einzelnen Elemente verwendeten Farben an.

Tabelle 7-5 Farben der Elemente

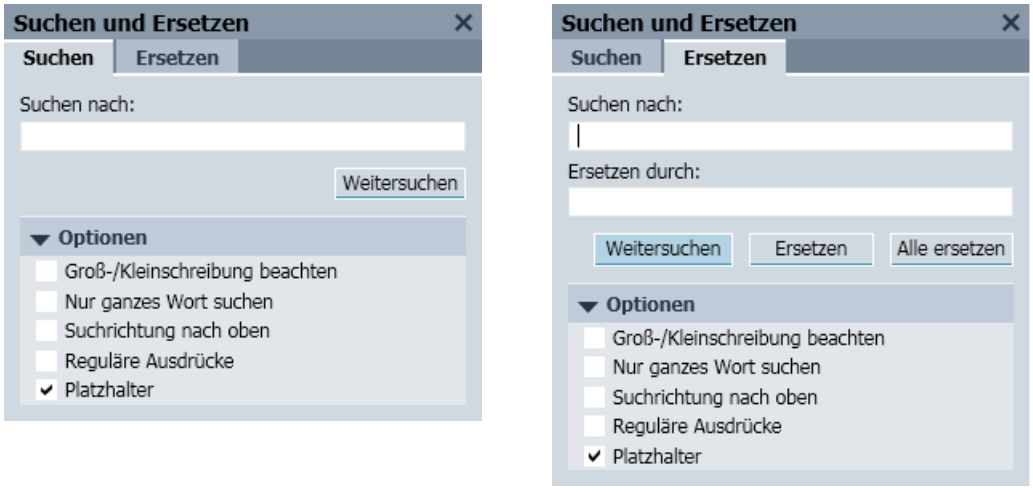
Element	Farbe
Eingangssignal	Grün
Ausgangssignal	Rot
Zustand	Olivgrün

Element	Farbe
Parameter	Rosa
Textkonstante	Braun
Schlüsselwort	Blau
Kommentar	Grau

Hinweis

Die farbliche Hervorhebung erfordert einigen Rechenaufwand für die Aktualisierung der Oberfläche des Texteditors, der sich unter Umständen dadurch bemerkbar macht, dass Sie beim Tippen eine gewisse Verzögerung spüren. Schalten Sie dann vor allem bei sehr umfangreichen Texten die Hervorhebung zumindest zeitweise einfach aus.

In allen drei Text-Editoren steht Ihnen unter dem Icon eine Funktion zum Suchen und Ersetzen von Text zur Verfügung. Sie können diese Funktion auch mit der Tastenkombination *Strg+F* bzw. *Strg+H* aufrufen. Wie in folgender Abbildung zu sehen ist, können Sie mit verschiedenen Optionen suchen.



Weitere Informationen zur Syntax für die Verhaltensbeschreibung finden Sie im Kapitel Syntax der Verhaltensbeschreibung (Seite 369).

7.4.4 Topologie

Die Topologiebeschreibung hat nur Bedeutung im Zusammenhang mit den Bibliotheken CONTEC und FLOWNET.

Weitere Informationen zu den toplogischen Aspekten finden Sie im Kapitel Topologische Eigenschaften (Seite 926) der Bibliothek CONTEC und im Kapitel Topologische Eigenschaften (Seite 803) der Bibliothek FLOWNET.

7.5 Die Visualisierung von Komponententypen

7.5.1 Einleitung

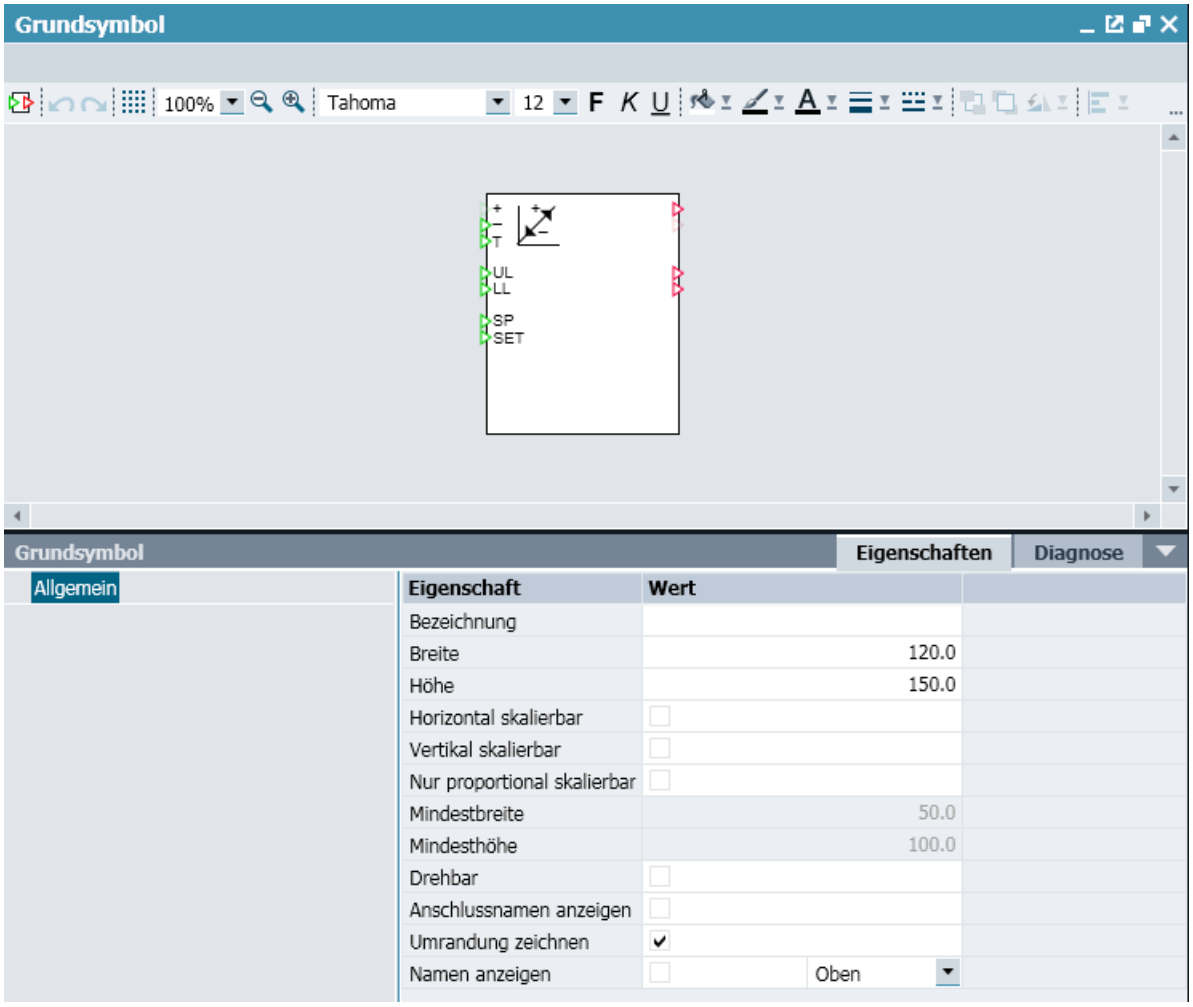
Für jeden Komponententyp ist eine grafische Repräsentation einer Komponenteninstanz in Form eines Grundsymbols angelegt. Das Grundsymbol wird in der Vorschau der Task-Card *Komponenten* angezeigt. Es repräsentiert jede Komponenteninstanz auf dem Diagramm.

Optional kann für einen Komponententyp eine Verknüpfungssicht angelegt sein, die einen zusätzlichen Zugriff auf die Komponenteninstanz erlaubt. Ebenfalls optional ist die Möglichkeit, ein Bedienfenster für einen Komponententyp zu definieren, das für jede Instanz bei laufender Simulation geöffnet werden kann, um Werte der Komponente zu setzen und anzuzeigen.

7.5.2 Das Grundsymbol

7.5.2.1 Editieren des Grundsymbols

Zum Editieren des Grundsymbols öffnen Sie den grafischen Symboleditor durch Doppelklick auf den Aspekt *Grundsymbol* im Projektbaum. Für die grafische Gestaltung des Grundsymbols stehen Ihnen in der Task-Card *Grafik* die grafischen Elemente des Diagrammeditors zur Verfügung. Sie können mit diesen grafischen Funktionen das Grundsymbol grafisch völlig frei in der zur Verfügung stehenden Fläche gestalten.



Grafische Elemente, die Sie beim Erstellen des Grund- und Verknüpfungssymbols einsetzen, können ohne Einschränkungen mit Animationen versehen werden.

7.5.2.2 Editieren von Grafiken

Für das Editieren der Grafik steht Ihnen der volle Funktionsumfang des Grafikeditors von SIMIT zur Verfügung. Weiterführende Informationen zu diesem Thema finden Sie unter "Grafiken visualisieren (Seite 227)".

7.5.2.3 Editieren der Anschlüsse

Sie können alle im Anschlusseditor spezifizierten Anschlüsse automatisch am Grundsymbol anordnen lassen. Klicken Sie dazu in der Werkzeugleiste auf das Symbol . Die Anschlüsse werden automatisch in der definierten Reihenfolge am Rand des Grundsymbols angeordnet: Eingänge am linken Rand, Ausgänge am rechten Rand des Symbols. Mit der Maus können Sie jeden Anschluss im vorgegebenen Raster von 5 Pixeln an die gewünschte Position ziehen.

Die Koordinaten eines Anschlusses werden Ihnen im Eigenschaftsfenster angezeigt. Sie können die gewünschte Position dort auch manuell eintragen. Im Unterschied zur Positionierung mit der Maus sind die von Hand eingetragenen Positionen nicht an das Raster gebunden. Sie müssen daher auch nicht mehr zwingend ganzzahlig sein.



Wenn ein Vektor von Ein- bzw. Ausgängen mit einer festen, nicht variablen Dimension gegeben ist, können Sie die einzelnen Anschlüsse des Vektors frei positionieren.

Hinweis

Änderungen im Anschlusseditor stehen erst dann im Symboleditor zur Verfügung, wenn Sie den Komponententyp speichern oder eine Aktualisierung in der Werkzeugleiste ( ) oder mit der Funktionstaste F5 veranlassen.

7.5.2.4 Editieren der Eigenschaften

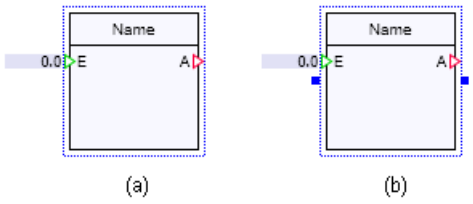
Im Eigenschaftsfenster können Sie Eigenschaften für das Grundsymbol definieren:

- **Bezeichnung**
  Wenn Sie eine *Bezeichnung* angeben, wird diese zentriert im oberen Bereich des Grundsymbols dargestellt und mit einer horizontalen Trennlinie vom restlichen Bereich abgesetzt.
  In Kombination mit der Option *Umrandung zeichnen* stellt das eine rudimentäre Möglichkeit zur Gestaltung einer Komponentensicht dar.

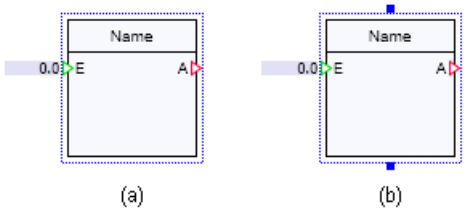


- **Breite**
  Die *Breite* des Grundsymbols in Pixeln können Sie hier als Zahlenwert angeben. Alternativ können Sie auch den linken oder rechten Rand der Symbolfläche im Editorfenster mit der Maus bei gedrückter linker Maustaste verschieben. Der Zahlenwert für die Breite wird automatisch nachgeführt.

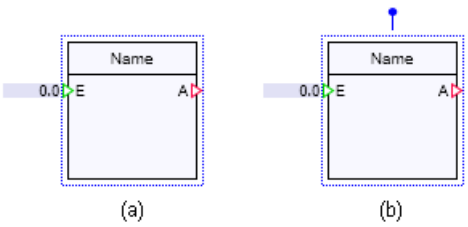
- **Höhe**
  Die *Höhe* des Grundsymbols in Pixeln können Sie hier als Zahlenwert angeben. Alternativ können Sie auch den oberen oder unteren Rand der Symbolfläche im Editorfenster mit der Maus bei gedrückter linker Maustaste verschieben, der Zahlenwert für die Höhe wird automatisch nachgeführt.
- **Horizontal skalierbar**
  Mit der Option *Horizontal skalierbar* legen Sie hier fest, ob das Grundsymbol der Komponenteninstanz in horizontaler Richtung auf einem Diagramm skalierbar sein soll oder nicht. Am Selektionsrahmen des Grundsymbols stehen dann entsprechende Anfasser zum Skalieren zur Verfügung.



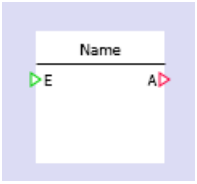
- **Vertikal skalierbar**
  Mit der Option *Vertikal skalierbar* legen Sie hier fest, ob das Grundsymbol der Komponenteninstanz in vertikaler Richtung auf einem Diagramm skalierbar sein soll oder nicht. Am Selektionsrahmen des Grundsymbols stehen dann entsprechende Anfasser zum Skalieren zur Verfügung.



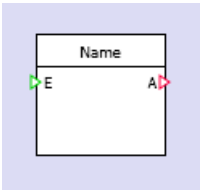
- **Mindestbreite**
  Die *Mindestbreite* des Grundsymbols kann durch horizontales Skalieren des Grundsymbols auf dem Diagramm nicht unterschritten werden.
- **Mindesthöhe**
  Die *Mindesthöhe* des Grundsymbols kann durch vertikales Skalieren des Grundsymbols auf dem Diagramm nicht unterschritten werden.
- **Drehbar**
  Mit dieser Option legen Sie fest, ob das Grundsymbol auf einem Diagramm drehbar sein soll oder nicht. Für ein drehbares erscheint ein entsprechender Anfasser am Selektionsrahmen des Grundsymbols.



- **Anschlussnamen anzeigen**
  Sie können mit dieser Option festlegen, ob die Anschlussnamen der Ein- und Ausgänge im Symbol angezeigt werden sollen. Beachten Sie, dass diese Anzeige nur für Eingänge am linken Rand und Ausgänge am rechten Rand des Grundsymbols möglich ist.



- **Umrandung zeichnen**
  Mit dieser Option können Sie festlegen, ob das Grundsymbol mit einer schwarzen Umrandung dargestellt werden soll.



- **Nur proportional skalierbar**
  Sie können mit dieser Option festlegen, dass die Höhe und Breite der Komponente nicht unabhängig voneinander skaliert werden kann, sondern nur proportional, d. h. mit konstantem Verhältnis Höhe zu Breite. Diese Option kann nur gesetzt werden, wenn die Komponente auch horizontal und vertikal skalierbar ist.
- **Maßstab**
  Bei der Erstellung eigener Typen von Fördertechnikkomponenten (Komponenten vom Typ der Bibliothek CONTEC) können Sie einen Maßstab für das Grundsymbol einstellen. Dieser Maßstab erfüllt dann zwei Aufgaben:
  - Bei der Gestaltung des Symbols für eine Fördermittelkomponente können Sie alle Abmessungen und Positionen bezogen auf den ausgewählten Maßstab in Millimetern angeben.
  - Bei der Gestaltung des Symbols für eine Fördergutkomponente ist die Abmessung, die sich aus der Wahl des Maßstabes ergibt, die voreingestellte Größe, mit der das Fördergut in der Materialliste von SIMIT angelegt wird.

Hinweis

Die Eigenschaft *Maßstab* ist nur verfügbar, wenn Sie die CONTEC-Bibliothek mit SIMIT lizenziert haben.

7.5.2.5 Maßstäblichkeit

Bei der Erstellung eigener Typen von Fördertechnikkomponenten können Sie auch im CTE einen Maßstab für das Grundsymbol und die Verknüpfung einstellen. Dieser Maßstab erfüllt zwei Aufgaben:

1. Bei der Modellierung eines Fördermittels können alle Abmessungen und Positionen bezogen auf den ausgewählten Maßstab in Millimetern angegeben werden.
2. Bei der Modellierung eines Fördergutes ist die Abmessung, die sich aus der Wahl des Maßstabes im CTE ergibt, die voreingestellte Größe, mit der das Fördergut in der Materialliste angelegt wird.

Hinweis

Diese Funktion ist nur verfügbar, wenn die CONTEC-Bibliothek lizenziert wurde.

Folgende Systemvariablen erlauben den Zugriff auf die tatsächlichen Abmessungen einer Komponente in der Verhaltensbeschreibung:

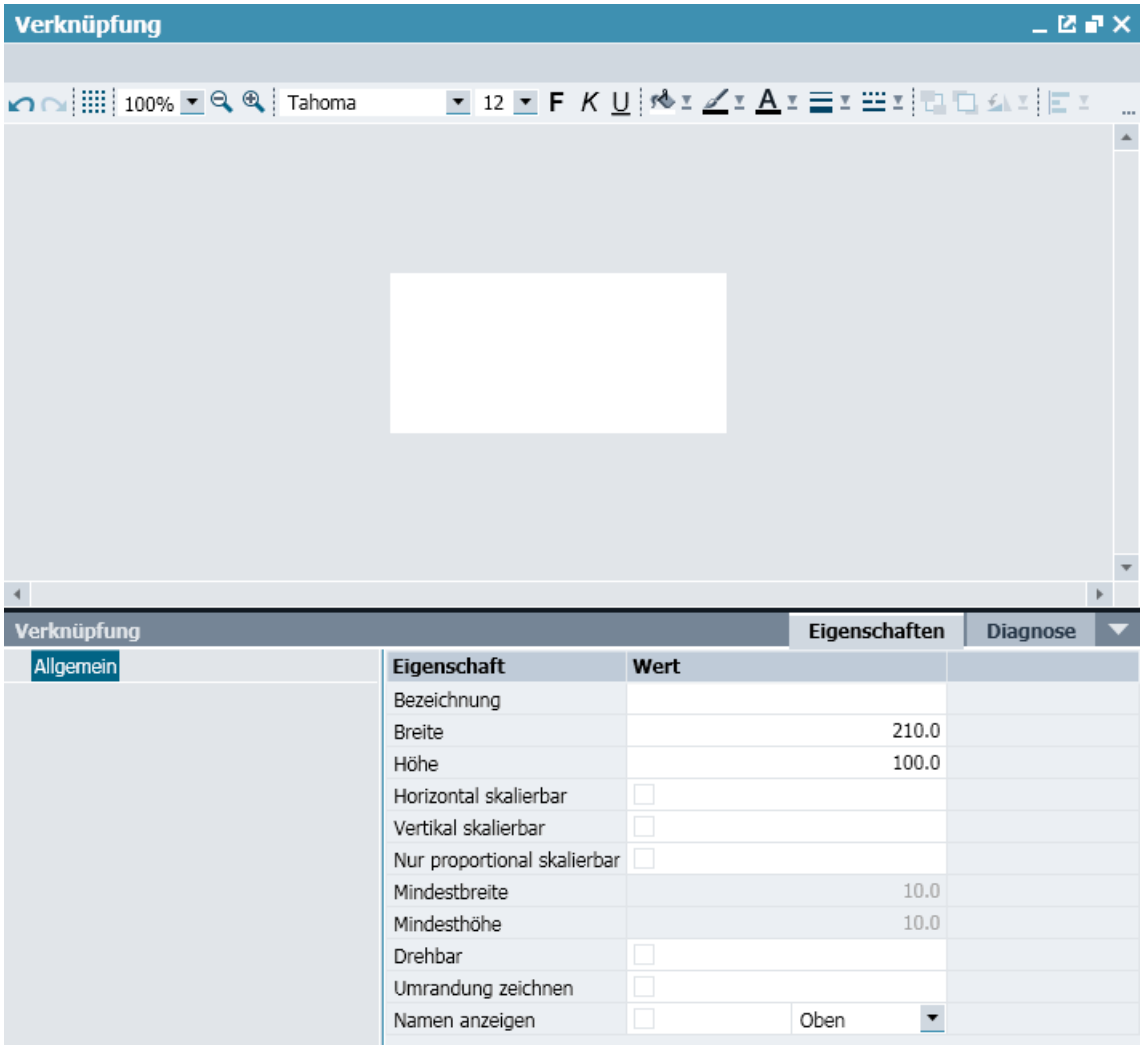
Tabelle 7-6 **Systemvariablen zur Bestimmung der Komponentenabmessung**

Systemvariable	Datentyp	Bedeutung
_WIDTH	analog	Breite (unskaliert) der Komponente in Pixel
_HEIGHT	analog	Höhe (unskaliert) der Komponente in Pixel
_SCALEX	analog	Horizontale Skalierung der Komponente
_SCALEY	analog	Vertikale Skalierung der Komponente
_TECHSCALE	analog	Maßstab des Planes, auf dem die Komponente liegt (Anzahl mm pro Pixel)

7.5.3 Das Verknüpfungssymbol

Das Grundsymbol eines Komponententyps dient zum Parametrieren und Verschalten der Komponenteninstanz auf einem Diagramm. Optional kann ein Komponententyp eine Verknüpfungssicht haben. Das in dieser Sicht gezeigte Symbol, das Verknüpfungssymbol, kann grafisch völlig unabhängig vom Grundsymbol gestaltet sein. Es besitzt im Unterschied zum Grundsymbol keine Anschlüsse. Die Grafikelemente des Verknüpfungssymbols können ansonsten wie im Grundsymbol frei gestaltet werden, also beispielsweise auch animiert sein, um aktuelle Simulationszustände einer Komponenteninstanz zu visualisieren.

Den Editor für das Verknüpfungssymbol öffnen Sie durch Doppelklick auf den Aspekt *Verknüpfung* im Projektbaum. Der sich öffnende grafische Editor bietet Ihnen, bis auf die auf Anschlüsse bezogenen Funktionen, die gleichen Funktionen wie der Grundsymboleditor.



Falls Sie für einen Komponententyp eine Verknüpfung vorsehen, können Sie in Ihrem SIMIT-Projekt beliebig viele Verknüpfungen für eine Instanz dieses Komponententyps auf Diagrammen anlegen.

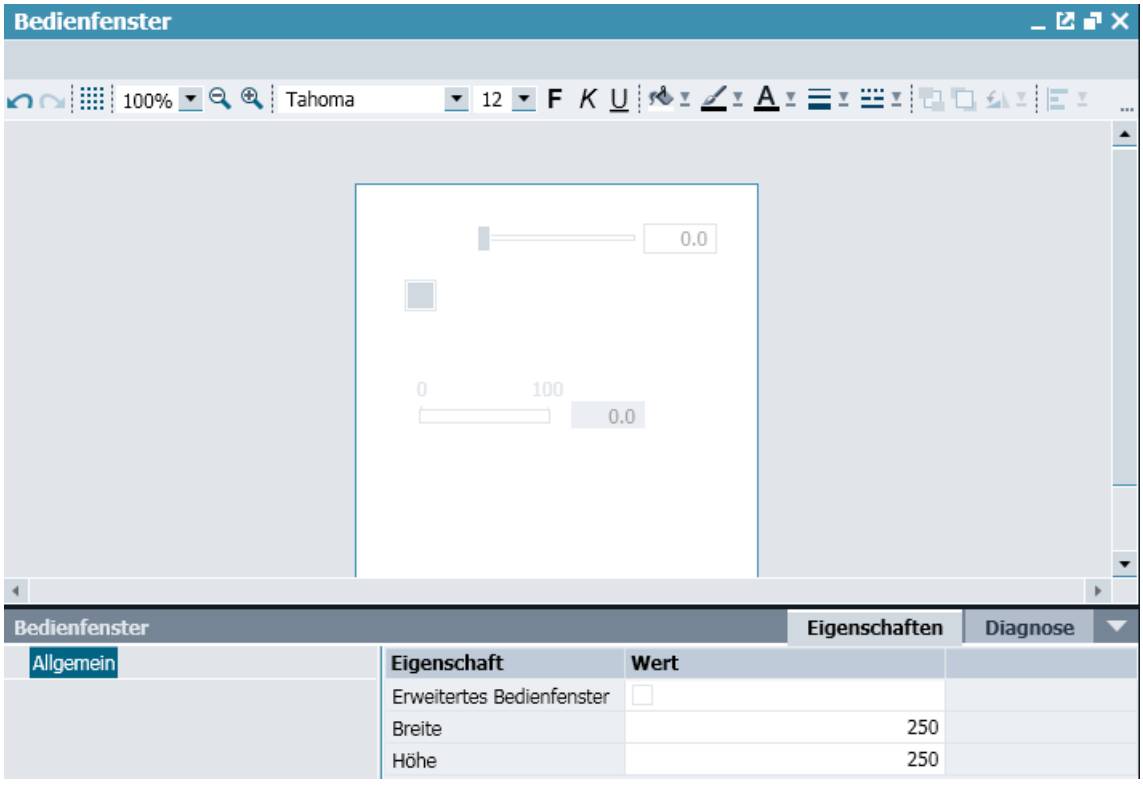
Siehe auch

Erstellen und Bearbeiten von Diagrammen (Seite 225)

7.5.4 Das Bedienfenster

Mit einem Bedienfenster können Sie bei laufender Simulation Werte der Komponenteninstanz setzen und anzeigen. Das Bedienfenster öffnen Sie durch Doppelklick auf die Komponente im Diagramm.

Zum Erstellen eines Bedienfensters öffnen Sie den Editor durch Doppelklick auf den Aspekt *Bedienfenster* im Projektbaum.

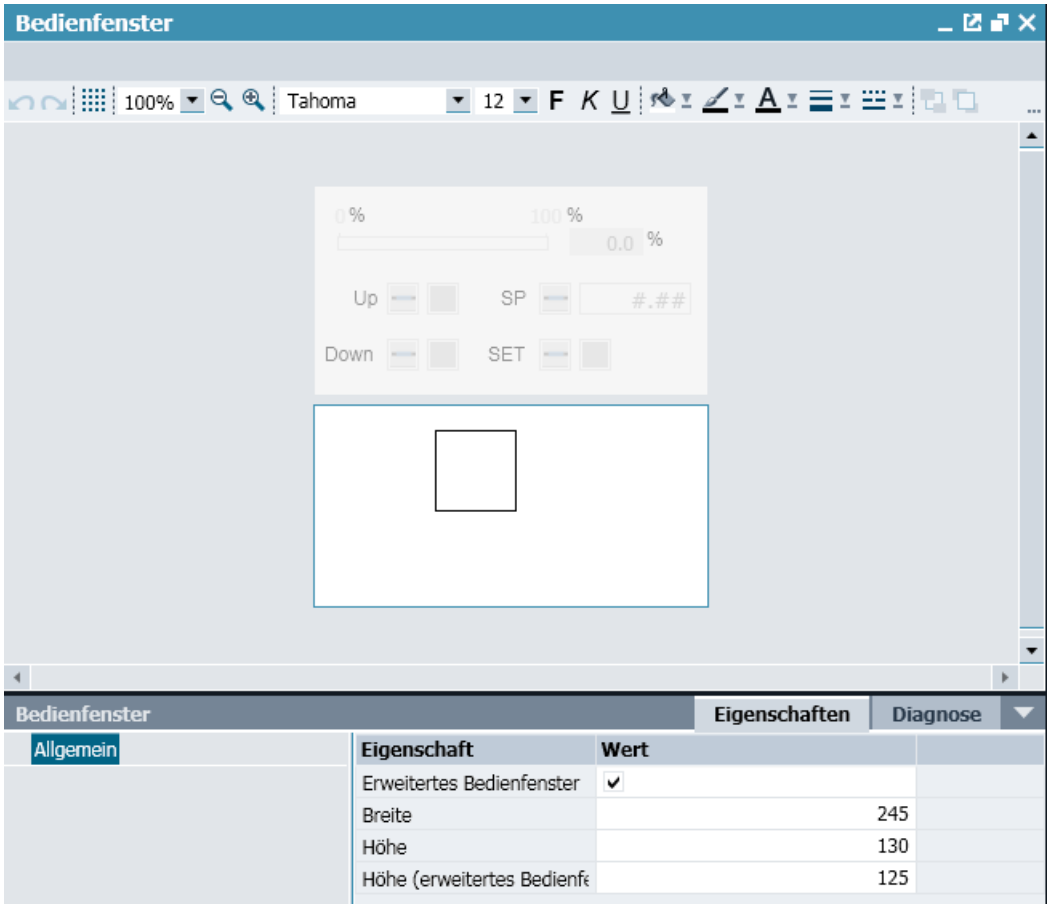


In einem Bedienfenster können alle Controls der Basisbibliothek von SIMIT eingesetzt werden. Diese Controls stehen Ihnen in der Task-Card *Controls* zur Verfügung. Sie können diese Controls aus der Task-Card auf die Zeichenfläche des Editors ziehen und mit geeigneten Ein- oder Ausgangssignalen des Komponententyps verbinden.

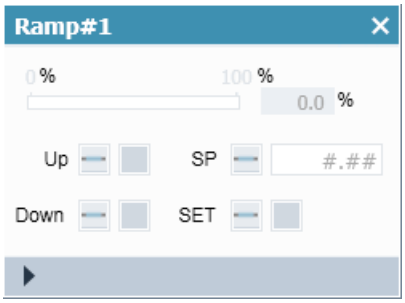
Weiter stehen Ihnen mit der Task-Card *Grafik* die Grafikfunktionen von SIMIT zur grafischen Gestaltung des Bedienfensters zur Verfügung. Beachten Sie, dass die Grafikobjekte im Bedienfenster nicht animiert werden können.

Im Eigenschaftsfenster stehen weitere Optionen für die Gestaltung des Bedienfensters zur Verfügung:

- **Erweitertes Bedienfenster**
  Sie können das Bedienfenster in zwei Bereiche unterteilen, um z. B. häufig benötigte Bedienelemente von seltener benötigten abzusetzen. Aktivieren Sie dazu die Option *Erweitertes Bedienfenster*. Im Editor steht dann eine weitere Fläche für das erweiterte Bedienfenster zur Verfügung. Diese Fläche ist gleich breit wie die Fläche für das Bedienfenster und kann in der Höhe beliebig variiert werden.



Im geöffneten Bedienfenster einer Komponenteninstanz können Sie das erweiterte Bedienfenster einfach durch einen Klick auf den unteren Rand öffnen.



- **Breite**
  Die Breite des Bedienfensters in Pixeln können Sie hier als Zahlenwert angeben. Alternativ können Sie den linken oder rechten Rand der Fläche des Fensters im Editor mit der Maus bei gedrückter linker Maustaste verschieben. Der Zahlenwert wird automatisch nachgeführt.
- **Höhe**
  Die Höhe des Bedienfensters in Pixeln können Sie hier als Zahlenwert angeben. Alternativ können Sie den oberen oder unteren Rand der Fläche des Fensters im Editor mit der Maus bei gedrückter linker Maustaste verschieben. Der Zahlenwert wird automatisch nachgeführt.

7.6 Syntax der Verhaltensbeschreibung

7.6.1 Übersicht

Die Verhaltensbeschreibung einer Komponente besteht aus einem einmalig während der Initialisierung ausgeführten Teil und einem in jedem zyklischen Berechnungsschritt ausgeführten Teil. Für beide Teile gilt die gleiche Beschreibungssyntax.

Zur Verhaltensbeschreibung einer Komponente gibt es zwei unterschiedliche Ansätze:

- **Gleichungsorientierter Ansatz**
  Der gleichungsorientierte Ansatz beschreibt jeden neuen Zustandswert bzw. Ausgangswert als explizite Funktion der Eingänge, Parameter und Zustände. Er eignet sich besonders bei der Modellierung physikalischer Zusammenhänge. Mit diesem Ansatz können Sie die Änderung von kontinuierlichen Zustandsgrößen auch durch gewöhnliche Differentialgleichungen beschreiben.
- **Anweisungsorientierter Ansatz**
  Der anweisungsorientierte Ansatz beschreibt die Berechnung von neuen Zustandsgrößen oder Ausgängen in Form von Programmieranweisungen, die sequentiell in der gegebenen Reihenfolge abgearbeitet werden. Dieser Ansatz eignet sich besonders, um technisches Verhalten zu modellieren.

Beide Ansätze können in einem Komponententyp auch kombiniert werden.

7.6.2 Umsetzung der Verhaltensbeschreibung auf C#-Code

Unabhängig davon, mit welchem Ansatz das Verhalten eines Komponententyps definiert ist, letztendlich erfolgt im Simulationsprojekt für jede Komponenteninstanz eine Umsetzung in C#-Code. Diese Umsetzung erfolgt automatisch und ist für den Benutzer nicht sichtbar und auch nicht von Bedeutung.

Da nicht alle formalen Fehler vor Generierung des Codes erkannt werden, kann gegebenenfalls auch der Compiler Fehlermeldungen erzeugen. Beachten Sie, dass sich die Angaben in derartigen Fehlermeldungen auf den generierten Code beziehen, der sich syntaktisch und in der Gliederung vom im Komponententyp spezifizierten Verhalten unterscheidet.

7.6.3 Der gleichungsorientierte Ansatz

7.6.3.1 Übersicht

Die Syntax, in der Sie das Verhalten Ihres Komponententyps formulieren, besteht beim gleichungsorientierten Ansatz nicht aus Anweisungen, sondern aus Beziehungen in Form von Gleichungen. Diese Gleichung beschreiben in expliziter Form, wie eine Zustandsgröße oder ein Ausgang aus anderen Größen berechnet wird, z. B. wie folgt:

```
Ausgang = Parameter * Eingang;
```

Grundsätzlich gilt:

- Jede Gleichung enthält das Gleichheitszeichen.
- Links vom Gleichheitszeichen steht die Größe, die bestimmt wird, rechts vom Gleichheitszeichen stehen die Größen, auf die lesend zugegriffen wird (explizite Form einer Gleichung).
- Am Ende jeder Gleichung steht ein Semikolon.
- Jede Größe darf nur genau einmal links vom Gleichheitszeichen stehen, d. h. jede Größe darf nur einmal bestimmt sein.

Auf der linken Seite des Gleichheitszeichens dürfen folgende Größen stehen:

- Ausgänge,
- Zustände (bei zeit-diskreten Zuständen nur der neue Wert, bei kontinuierlichen Zuständen nur das Differential, d. h. die Wertänderung) und
- lokale Variablen.

Auf der rechten Seite des Gleichheitszeichens dürfen folgende Größen stehen:

- Eingänge,
- Zustände (bei kontinuierlichen Zuständen darf hier nicht das Differential stehen),
- Parameter,
- lokale Variablen und
- Konstanten.

7.6.3.2 Lokale Variablen

Sie können innerhalb der Verhaltensbeschreibung für die Initialisierung oder die zyklische Berechnung lokale Variablen wie folgt definieren:

```
Datentyp Name[,Name];
```

Beispiel:

```
binary b1, b2, b3;
```

Erlaubt sind die in folgender Tabelle gelisteten Datentypen.

Tabelle 7-7 **Datentypen für lokale Variablen**

Datentyp	Bedeutung	Wertebereich	Vorbelegung
binary	Binärwerte	True/False	False
analog	Gleitkommawerte	$\pm 5.0 \times 10^{-324}$ bis $\pm 1.7 \times 10^{308}$	0.0
integer	Ganzzahlige Werte	−9.223.372.036.854.775.808 bis +9.223.372.036.854.775.807	0

Der Name einer lokalen Variablen darf nur aus Buchstaben, Ziffern und dem Unterstrich bestehen und muss mit einem Buchstaben beginnen.

Lokale Variablen dienen dazu, Zwischenergebnisse abzuspeichern, die im gleichen Bearbeitungsschritt noch einmal benötigt werden.

Wenn Sie auf berechnete Werte auch im nächsten Berechnungsschritt wieder zugreifen möchten, legen Sie dafür keine lokalen Variablen, sondern zeit-diskrete Zustandsgrößen an.

7.6.3.3 Konstanten

Welche Konstanten Sie benutzen können, hängt vom Datentyp der Ergebnisgröße ab. In folgender Tabelle sind die Konstanten zu jedem Datentyp beschrieben.

Tabelle 7-8 Datentypen für Konstanten

Datentyp	Konstanten
binary	"FALSE" oder "TRUE"
analog	Dezimalbruch mit einem Punkt als Dezimaltrennzeichen, z. B. "125.61" Exponentialschreibweise, z. B. "62.2e-4"
integer	Ziffernfolge ohne Tausendertrennzeichen, z. B. "125985"

7.6.3.4 Die Berechnungsreihenfolge

Die Verhaltensbeschreibung eines Komponententyps sowohl für die Initialisierung, als auch die zyklische Berechnung besteht aus einzelnen Gleichungen. Sie definieren mit der Beschreibung nur Beziehungen und Abhängigkeiten in den Beziehungen. Sie definieren insbesondere keine Berechnungsreihenfolge. Die Reihenfolge, in der Sie diese Gleichungen niederschreiben, spielt keine Rolle für die Berechnung.

Beim Erzeugen einer ablauffähigen Simulation analysiert SIMIT alle Gleichungen in einer Komponenteninstanz und ermittelt die Reihenfolge, in der Sie berechnet werden anhand der gegenseitigen Abhängigkeiten. SIMIT ordnet die Berechnungsreihenfolge immer so, dass Gleichungen, die eine Größe festlegen, vor den Gleichungen berechnet werden, in denen diese Größe benötigt wird.

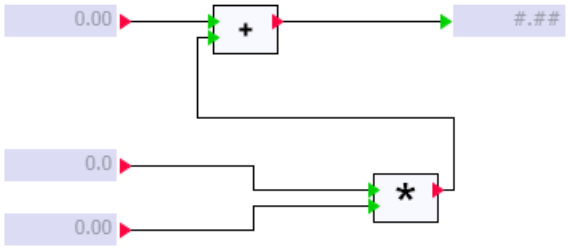
In folgendem Beispiel wird also zunächst der lokalen Variablen *p* der Wert 3.14 zugewiesen, anschließend wird der neue Zustand *Z* berechnet und dann dem Ausgang der Wert des neuberechneten Zustands zugewiesen.

Zyklische Berechnung

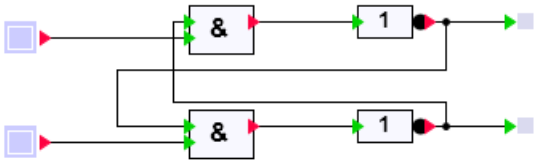
```
1
2  // ---------------------------------------------------
3  // Copyright (c) Siemens AG 2006. All rights reserved.
4  // Industrial Solutions and Services, I&S IS E&C
5  // ---------------------------------------------------
6  analog p;
7
8  A = @Z
9  @Z = E * 2.0 * p;
10 p = 3.14;
11
```

Die Berechnungsreihenfolge ergibt sich automatisch aus der Analyse der Abhängigkeiten und ist in diesem Fall sogar genau umgekehrt zur Reihenfolge in der Beschreibung.

SIMIT analysiert die Gleichungen aber nicht nur innerhalb einer Komponenteninstanz, sondern über alle Komponenteninstanzen mit ihren gegenseitigen Verknüpfungen in Ihrem SIMIT-Projekt und definiert eine entsprechende Berechnungsreihenfolge für das Projekt. In folgendem Beispiel wird immer die Multiplikation vor der Addition ausgeführt, weil das Produkt für die Addition benötigt wird.



Es können Fälle auftreten, in denen keine eindeutige Berechnungsreihenfolge bestimmt werden kann. In folgendem Beispiel sind die Eingangswerte der beiden Und-Verknüpfungen abhängig vom Ausgangswert der jeweils anderen. Durch diese Rückkopplungen wird immer eine der beiden Und-Verknüpfungen mit einem nicht aktuellen Eingangswert berechnet, d. h. mit einem Wert, der aus dem vorhergehenden Rechenzyklus stammt.



In derartigen Fällen ist nicht festgelegt, in welcher Reihenfolge die Gleichungen bearbeitet werden. SIMIT versucht aber immer die Berechnungsreihenfolge zu maximieren, d. h. so viele Gleichungen wie möglich, in eine Reihenfolge zu sortieren. Im obigen Beispiel wird so von SIMIT immer die Kette "Schalter – AND – NOT – Binäranzeige" in dieser Reihenfolge bearbeitet. Allerdings ist unbestimmt, ob zuerst die obere oder die untere Kette bearbeitet wird.

7.6.3.5 Operatoren

Die Größen rechts vom Gleichheitszeichen können Sie durch Operatoren miteinander verknüpfen. Je nach Datentyp sind die in folgender Tabelle gelisteten Operatoren zulässig.

Tabelle 7-9 Erlaubte Operatoren

Operation	Operator	Datentyp	Priorität
Klammerung	(Ausdruck)	binary, integer, analog	höchste
Funktionsaufruf	Funktionsname (Parameterliste)	binary, integer, analog	
Vorzeichenwechsel	–	integer, analog	
Negation	NOT	binary	
Multiplikation	*	integer, analog	
Division	/	integer, analog	
Modulo	%	integer	
Addition	+	integer, analog	
Subtraktion	–	integer, analog	
Vergleich	<, >, <=, >=	integer, analog	
Gleichheit	=	binary, integer, analog	
Ungleichheit	!=	binary, integer, analog	
logisches Und	AND	binary	
logisches Exklusiv-Oder	XOR	binary	
logisches Oder	OR	binary	niedrigste

7.6.3.6 Bedingte Zuweisungen

Einer Größe können Sie abhängig von einer oder mehreren Bedingungen verschiedene Werte mit folgender Syntax zuweisen:

```
y = {Bedingung1: Ausdruck1; Bedingung2: Ausdruck2; ...
            ELSE Ausdruck0};
```

Beispiel:

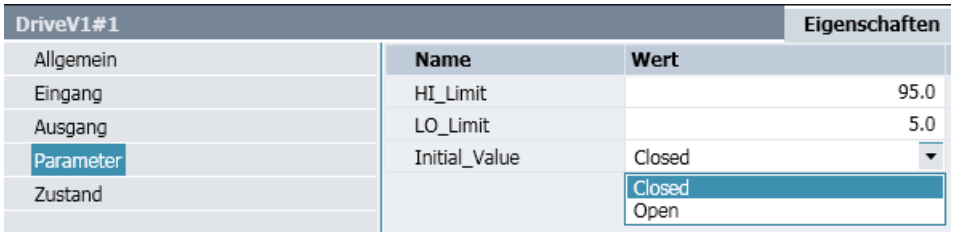
```
m = {p<1: m1; p<0: m2; ELSE m3};
```

Die Bedingungen werden von links nach rechts abgearbeitet. Sobald eine Bedingung erfüllt ist, wird der zugehörige Ausdruck ausgewertet und der Variablen links vom Gleichheitszeichen zugewiesen. Trifft keine Bedingung zu, wird der Ausdruck hinter dem Schlüsselwort *ELSE* ausgewertet.

Sie können beliebig viele Bedingungs-Ausdruckspaare angeben. Der "ELSE"-Ausdruck ist in jedem Fall anzugeben.

7.6.3.7 Aufzählungstypen

Parameter können als Typ einen Aufzählungstypen besitzen. Sie können dann im Eigenschaftsfenster des Diagramm-Editors zur Parametrierung der Komponenteninstanz aus einer Liste den gewünschten Eintrag auswählen.



In der Verhaltensbeschreibung können Sie einen solchen Parameter abfragen, indem Sie sich auf den Namen des Aufzählungstyps und mit einem Punkt getrennt davon den entsprechenden Listeneintrag beziehen, z. B. wie folgt:

```
A = {Initial_Value = ClosedOpen.Open: TRUE; ELSE FALSE};
```

7.6.3.8 Vektoren

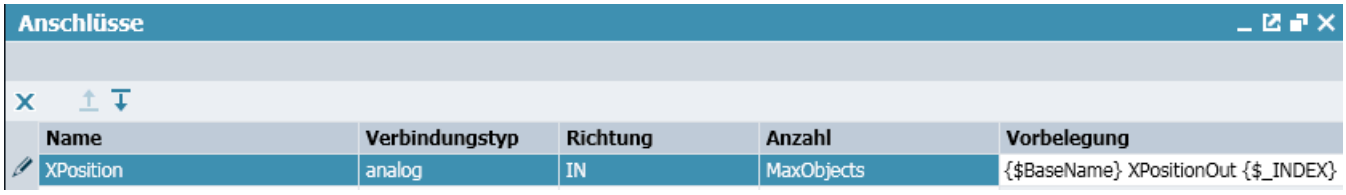
Alle Eingänge, Ausgänge, Parameter oder Zustände können Sie auch als Vektoren definieren. Zugriff in der Verhaltensbeschreibung auf die Elemente des Vektors haben Sie dann wie folgt:

- **Einzelelemente**
  Für den Zugriff auf ein Element eines Vektors fügen Sie dem Signalnamen den gewünschten Index in eckigen Klammern an.
  Beachten Sie, dass das erste Element des Vektors den Index *0* hat. Der Index muss ein konstanter, nicht-negativer ganzzahliger Wert sein und darf die vorhandene Anzahl an Elementen dieses Vektors nicht überschreiten.
  Beispiel:
  ```
  Ausgang[2] = Eingang * 2.0;
  ```
- **Bereich**
  Sie können eine Gleichung für mehrere Elemente eines Vektors spezifizieren, indem Sie einen Bereich angeben:
  ```
  Vektorname[Index1 TO Index2]
  ```
  SIMIT löst diese Vektorgleichung auf, so dass folgende zwei Beispiele genau identisch sind:
  Beispiel 1:
  ```
  Ausgang[0 TO 2] = 1.0;
  ```
  Beispiel 2:
  ```
  Ausgang[0] = 1.0;
  Ausgang[1] = 1.0;
  Ausgang[2] = 1.0;
  ```
  Wenn Sie in einer Vektorgleichung auf den Index Bezug nehmen wollen, können Sie eine Indexvariable angeben:
  ```
  Vektorname[Indexvariable: Index1 TO Index2]
  ```
  Diese Indexvariable ist dann auf der rechten Seite der Gleichung verwendbar. Sie muss nicht zusätzlich als lokale Variable deklariert werden, darf aber nur aus einem einzigen Buchstaben bestehen.
  Beispiel:
  ```
  Ausgang[i:0 TO 2] = Eingang[i + 1];
  ```
  Die Auflösung in einzelne Gleichungen sieht in diesem Fall wie folgt aus:
  ```
  Ausgang[0] = Eingang[1];
  Ausgang[1] = Eingang[2];
  Ausgang[2] = Eingang[3];
  ```
- **Gesamtvektor**
  Wenn Sie den Bereich auf den gesamten Vektor ausdehnen wollen, können Sie auch schreiben:
  ```
  Vektorname[ALL]
  ```
  oder mit Angabe einer Indexvariablen:
  ```
  Vektorname[Indexvariable: ALL]
  ```
- **Vektor einer komplexen Größe**
  Besitzt ein Signal einen komplexen Verbindungstyp, so steht die Indexangabe in eckigen Klammern am Ende des kompletten Namens.
  Beispiel:
  ```
  X1.Re[3] = Y1.Im[1] * Y2.Im[2];
  ```

Systemvariable *_INDEX*

Die Systemvariable *_INDEX* dient zur variablen Definition von impliziten Verbindungen von Vektoren.

Bei der Angabe des Signalnamens können Sie die Variable *_INDEX* mit dem Ausdruck {$_INDEX} benutzen. Bei der Instanziierung der Komponente wird dann dieser Ausdruck in jedem Element durch den tatsächlichen Index, beginnend mit eins, ersetzt.



Wenn für eine Komponente nach diesem Beispiel die Werte

- Parameter MaxObjects ist 2 und
- Parameter BaseName ist "LifterBase#1"

eingestellt werden, dann ergibt sich für den Eingangsvektor XPosition folgende Vorbelegung:

- XPosition1: LifterBase#1 XPositionOut1
- XPosition2: LifterBase#1 XPositionOut2

7.6.3.9 Funktionsaufrufe mathematischer Standardfunktionen

Sie können alle mathematischen Standardfunktionen benutzen und überall in Ihrer Verhaltensbeschreibung einsetzen, wo auch eine skalare Eingangsgröße zulässig ist.

Dem Namen der mathematischen Standardfunktion stellen Sie einfach einen Unterstrich voran und fügen eine in runden Klammern gesetzte Parameterliste an. Die einzelnen Parameter werden durch Komma getrennt. Besitzt eine Funktion keine Parameter, wird sie mit einer leeren Parameterliste aufgerufen. In folgender Tabelle sind alle zur Verfügung stehenden mathematischen Standardfunktionen gelistet.

Tabelle 7-10 **Liste der mathematischen Standardfunktionen**

Funktion	Rückgabe-wert	Beschreibung
_sqrt(x)	analog	$y = \sqrt{x};\ x \geq 0$
_abs(x)	analog/ integer	$y = \lvert x \rvert$
_exp(x)	analog	$y = e^x$
_pow(x, z)	analog	$y = x^z$
_log(x)	analog	Natürlicher Logarithmus: $y = \ln(x);\ x > 0$
_log10(x)	analog	Dekadischer Logarithmus: $y = \lg(x);\ x > 0$
_ceil(x)	analog	Kleinste ganze Zahl größer oder gleich x
_floor(x)	analog	Größte ganze Zahl kleiner oder gleich x
_rand()	integer	Zufallswert zwischen 0 und 32767
_sin(x)	analog	$y = \sin(x)$; Winkel x im Bogenmaß
_cos(x)	analog	$y = \cos(x)$; Winkel x im Bogenmaß

Funktion	Rückgabe-wert	Beschreibung
_tan(x)	analog	$y = \tan(x)$; Winkel x im Bogenmaß; $x \neq \pm(2n + 1)\pi/2$
_asin(x)	analog	$y = \arcsin(x)$; $-1 \leq x \leq 1$
_acos(x)	analog	$y = \arccos(x)$; $-1 \leq x \leq 1$
_atan(x)	analog	$y = \arctan(x)$;
_atan2(y, x)	analog	$y = \begin{cases} \arctan(\frac{y}{x}) & x > 0 \\ \pi + \arctan(\frac{y}{x}) & y \geq 0, x < 0 \\ -\pi + \arctan(\frac{y}{x}) & y < 0, x < 0 \\ \frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefiniert} & y = 0, x = 0 \end{cases}$
_sinh(x)	analog	$y = \sinh(x)$; Winkel x im Bogenmaß
_cosh(x)	analog	$y = \cosh(x)$; Winkel x im Bogenmaß
_tanh(x)	analog	$y = \tanh(x)$; Winkel x im Bogenmaß
_min(x, y)	analog/ integer	Der kleinere der beiden Werte x oder y
_max(x, y)	analog/ integer	Der größere der beiden Werte x oder y
_trunc2byte(x)	byte	Der ganzzahlige Anteil von x modulo 2^8
_trunc2int(x)	int	Der ganzzahlige Anteil von x modulo 2^{16}
_trunc2long(x)	long	Der ganzzahlige Anteil von x modulo 2^{32}
_round2byte(x)	byte	Die gerundete Zahl x modulo 2^8
_round2int(x)	int	Die gerundete Zahl x modulo 2^{16}
_round2long(x)	long	Die gerundete Zahl x modulo 2^{32}
_characteris-tic(c[ALL], x)	analog	Sonderfall Kennlinie (weitere Informationen hierzu finden Sie im Abschnitt: Der Parametertyp characteristic (Seite 389))

7.6.3.10 Selbstdefinierte Funktionen

Wenn Sie einen eigenen Operator benötigen, können Sie eine selbstdefinierte Funktion unter dem Teilaspekt *Funktionen* in der Verhaltensbeschreibung definieren. Aufrufen können Sie selbstdefinierte Funktion in den Teilaspekten *Initialisierung* und *Zyklische Berechnung*.

Mit einem Aufruf einer selbstdefinierten Funktion können Sie auch mehrere Variablen setzen, wenn Sie die Funktion entsprechend deklariert haben. Beim Aufruf einer selbstdefinierten Funktionen ist die Liste der berechneten Funktionswerte immer in runden Klammern anzugeben.

Beispiel:

```
(y1, y2) = EigeneFunktion(x1, x2);
```

Auch ein einziger Funktionswert als Rückgabewert muss in runde Klammern gesetzt werden.

Selbstdefinierte Funktionen können Sie nur über einen Aufruf wie oben verwenden. Auch wenn Ihre selbstdefinierte Funktion nur einen Rückgabewert hat, können Sie diese Funktion nicht wie die mathematischen Standardfunktionen als Skalar in andere Beziehungen einsetzen, sondern müssen diese Funktion wie oben aufrufen.

Die Syntax, die zur Formulierung von Funktionen benutzt wird, unterscheidet sich wesentlich von der Gleichungssprache. Sie folgt dem anweisungsorientierten Ansatz. Weitere Informationen hierzu finden Sie im Kapitel: Funktionen (Seite 380).

7.6.3.11 Differentialgleichungen

Einleitung

Wenn Sie einen analogen Zustand Ihres Komponententyps als *kontinuierlich* definiert haben, wird in der Verhaltensbeschreibung nicht angegeben, wie dieser Zustand berechnet wird, sondern von welchen Größen und wie sich die zeitliche Änderung dieses Zustands ergibt: Die Zustandsänderung wird in Form einer gewöhnlichen Differentialgleichung definiert.

Schreibweise des Differentials

Die mathematische Schreibweise des Differentials d/dt wird in SIMIT mit einem Dollar-Zeichen abgekürzt. Sie formulieren z. B. für die Änderung des kontinuierlichen Zustands "Masse" in einem Behälter eine Gleichung der Form:

```
$Masse = Zufluss - Abfluss;
```

Der neue Wert der Zustandsgröße wird nun von SIMIT in jedem Rechenschritt bestimmt, indem die Differentialgleichung numerisch gelöst wird.

Sie können auch mehrere kontinuierliche Zustände in Ihrem Komponententyp durch Differentialgleichung beschreiben, die voneinander abhängig sind. Es ergibt sich dann z. B. folgendes System von Differentialgleichungen:

```
$Z1 = Z2 + Faktor1 * Z3 - Eingang1;
$Z2 = Faktor2 * Z1;
$Z3 = Z1 + Z2;
```

SIMIT benutzt zur Lösung solcher gewöhnlicher Differentialgleichungssysteme ein numerisches Lösungsverfahren basierend auf dem Verfahren nach Runge-Kutta-Merson.

Wie schon bei der Festlegung der Berechnungsreihenfolge, greift auch hier die globale Sicht von SIMIT auf das Simulationsprojekt: SIMIT behandelt nicht nur die Differentialgleichungen einer Komponente, sondern alle Differentialgleichungen des gesamten Simulationsprojekts geschlossen, d. h. es bildet das Differentialgleichungssystems des gesamten Simulationsmodells auf das numerische Lösungsverfahren ab.

Korrekturen für die Zustandsgrößen

In der Simulation können in der Regel nicht alle Aspekte der Realität berücksichtigt werden. Ein Simulationsmodell ist daher gegenüber der Realität vereinfacht und gilt auch nur in einem durch die angenommenen Voraussetzungen definierten Bereich. Der definierte Bereich für Zustandsgrößen kann bei der Lösung von Differentialgleichungen im Allgemeinen überschritten werden. Typische Fälle sind z. B. die Berechnung des Füllstands eines Behälters. In diesen Fällen müssen die Zustandsgrößen nach der Lösung des Differentialgleichungssystems überprüft und gegebenenfalls korrigiert werden.

SIMIT bietet Ihnen die Möglichkeit, eine bedingungsabhängige Gleichung zur Korrektur von Zustandsgrößen zu formulieren. Den Korrekturwert für den Zustand kennzeichnen Sie einfach durch ein vorangestelltes #-Zeichen und formulieren eine Korrekturgleichung z. B. wie folgt:
`#Masse = {@Masse < 0.0: 0.0; ELSE @Masse};`

Diese Gleichung bewirkt, dass der Wert für *Masse* auf 0.0 gesetzt wird, falls das Lösungsverfahren einen Wert berechnet hat, der kleiner 0.0 ist, ansonsten ist der Wert für *Masse* identisch mit dem berechneten Wert *@Masse*.

Zugriff auf kontinuierliche Zustandsgrößen

Wie oben erwähnt, werden auch kontinuierliche Zustandsgrößen zyklisch berechnet. Die Werte stehen damit nur zu den durch den Zyklus definierten Zeitpunkten zur Verfügung. In einem Zeitschritt können Sie lediglich auf den Wert im vorherigen Berechnungszyklus und auf den neu berechneten bzw. korrigierten Wert zugreifen. Wie in der folgenden Zusammenstellung zu sehen ist, greifen Sie auf den neu berechneten Wert für einen Zustand *Z* durch ein vorangestelltes @-Zeichen zu.

Tabelle 7-11 Zugriff auf kontinuierlich Zustandsgrößen

Notation	Zugriff
Z	Wert aus dem vorherigen Rechenzyklus
@Z	Neuer Wert nach der Integration der Zustandsgleichung
#Z	korrigierter Wert, der anstelle von @Z im nächsten Rechenzyklus verwendet werden soll.

7.6.3.12 Zugriff auf diskrete Zustandsgrößen

Für diskrete Zustandsgrößen definieren Sie den Initialisierungswert oder den neuen Zustandswert in einem Zyklus durch eine explizite Gleichung. Der neue Wert für einen Zustand *Z* wird durch ein vorangestelltes @-Zeichen gekennzeichnet.

Tabelle 7-12 Zugriff auf diskrete Zustandsgrößen

Notation	Zugriff
Z	Wert aus dem vorherigen Rechenzyklus
@Z	Neuer Wert

7.6.4 Der anweisungsorientierte Ansatz

7.6.4.1 Einleitung

Der anweisungsorientierte Ansatz wird verwendet, um eigene Funktionen oder ganze Funktionsblöcke zu formulieren. Innerhalb von selbstdefinierten Funktionen und Blöcken werden nicht Gleichungen formuliert, sondern Programmbefehle, die in der definierten Reihenfolge ausgeführt werden. Die Syntax ist innerhalb der Funktions- bzw. Blockdefinition identisch.

7.6.4.2 Funktionen

Funktionen müssen Sie im Teilaspekt *Funktionen* der Verhaltensbeschreibung ablegen. Den dafür vorgesehenen Texteditor öffnen Sie einfach durch Doppelklick auf den Eintrag *Funktionen* in der Navigation.

Eine Funktion besteht aus folgenden Elementen:

- dem Schlüsselwort *FUNCTION*,
- der Funktionsdeklaration,
- dem Anweisungsteil und
- dem Schlüsselwort *END_FUNCTION*.

Syntax:

```
FUNCTION Funktionsname (Datentyp Ausgangsgröße1 [, Datentyp
    Ausgangsgröße2]) : (Datentyp Eingangsgröße1 [, Datentyp
    Eingangsgröße2])
    Anweisungsteil
END_FUNCTION
```

Als Datentypen für die Ein- und Ausgangsgrößen sind lediglich die Grunddatentypen *binary*, *integer* und *analog* erlaubt.

Beispiel:

```
FUNCTION Trigonometrie (analog y1, analog y2) :
                (analog x1, analog x2, binary state)
  if (state) {
      y1 = _sin(x1);
      y2 = _tan(x2);
  }
  else {
   y1 = 0.0;
   y2 = 0.0;
  }
END_FUNCTION
```

Diese Funktion kann dann im Initialisierungsteil oder in der zyklischen Berechnung wie folgt aufgerufen werden:

```
(Out1, Out2) = Trigonometrie(In1, In2, In3);
```

7.6.4.3 Blöcke

Sie können in der Initialisierung bzw. der zyklischen Berechnung beliebig viele Blöcke einfügen, innerhalb derer Sie eine anweisungsorientierte Verhaltensbeschreibung, wie bei Funktionen, verwenden können.

Ein Block besteht aus folgenden Elementen:

- dem Schlüsselwort *BLOCK*,
- dem Anweisungsteil und
- dem Schlüsselwort *END_BLOCK*.

Syntax:
```
BLOCK
    Anweisungsteil
END_BLOCK
```

Die Anweisungen innerhalb eines Blocks werden in genau der Reihenfolge ausgeführt, wie Sie sie definiert haben. In welcher Reihenfolge die Blöcke als Ganzes gerechnet werden, wird aber von SIMIT anhand der Abhängigkeiten, analog zum gleichungsorientierten Ansatz festgelegt. SIMIT analysiert dazu, welche Größen der Verhaltensbeschreibung innerhalb eines Blocks berechnet, d. h. verändert werden und auf welche nur lesend zugegriffen wird.

Wenn Sie eine Größe in mehr als einem Block verändern, also z. B. eine Ausgangsgröße in zwei Blöcken setzen, liegt ein Widerspruch vor, den SIMIT nicht auflösen kann. Solche Mehrfachdefinitionen von Größen sind folglich nicht erlaubt.

7.6.4.4 Lokale Variablen

Sie können innerhalb der Verhaltensbeschreibung für die Initialisierung oder die zyklische Berechnung lokale Variablen definieren, die dann auch dort in allen Blöcken verwendet werden können. Weitere Informationen hierzu finden Sie im Abschnitt: Lokale Variablen (Seite 370).
Sie können aber auch innerhalb einer Funktion oder eines Blocks lokale Variablen definieren:
```
Datentyp Name[,Name];
```

Beispiel:
```
binary b1, b2, b3;
```

Diese Variablen sind dann nur innerhalb des Blocks oder der Funktion, in der sie definiert sind, gültig. Zulässig sind für diese Variablen die in folgender Tabelle gelisteten Datentypen.

Tabelle 7-13 Datentypen für Variablen in Blöcken und Funktionen

Datentyp	Bedeutung	Wertebereich	Vorbelegung
binary, bool	Binärwerte	True/False	False
analog, double	Gleitkommawerte	$\pm 5.0 \times 10^{-324}$ bis $\pm 1.7 \times 10^{308}$	0.0
integer, long	Ganzzahlige Werte	−9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807	0
byte	Ganzzahlige Werte	0 bis 255	0
sbyte	Ganzzahlige Werte	−128 bis 127	0
ushort	Ganzzahlige Werte	0 bis 65535	0
short	Ganzzahlige Werte	−32.768 bis 32.767	0

Datentyp	Bedeutung	Wertebereich	Vorbelegung
uint	Ganzzahlige Werte	0 bis 4.294.967.295	0
int	Ganzzahlige Werte	−2.147.483.648 bis 2.147.483.647	0
ulong	Ganzzahlige Werte	0 bis 18.446.744.073.709.551.615	0
text, string	Zeichenkette (Text)		

Der Name einer lokalen Variablen darf nur aus Buchstaben, Ziffern und dem Unterstrich bestehen und muss mit einem Buchstaben beginnen.

Auch hier dienen die lokalen Variablen im Block lediglich dazu, Zwischenergebnisse im Block abzuspeichern, die im gleichen Bearbeitungsschritt noch einmal benötigt werden. Wenn Sie auf berechnete Werte auch im nächsten Berechnungsschritt wieder zugreifen möchten, legen Sie keine lokalen Variablen, sondern zeit-diskrete Zustandsgrößen an.

7.6.4.5 Felder

Sie können lokale Variablen auch als Felder (Arrays) definieren:
```
Datentyp Name[Dimension];
```

Die Dimension muss eine ganzzahlige Konstante sein. Sie können alternativ ein Feld auch gleich mit Werten definieren:
```
Datentyp Name[] = {Wert1, Wert2, …};
```

Die Dimension ergibt sich in diesem Fall automatisch aus der Anzahl der definierten Werte. Die Feldelemente sind mit den in der Definition gegebenen Werten initialisiert.

7.6.4.6 Konstanten

Welche Konstanten Sie benutzen können, hängt vom Datentyp der Ergebnisgröße ab. In folgender Tabelle sind die Konstanten zu jedem Datentyp beschrieben.

Datentyp	Konstanten
Binärwert	"FALSE" oder "TRUE"
Gleitkommawert	Dezimalbruch mit einem Punkt als Dezimaltrennzeichen, z. B. "125.61" Exponentialschreibweise, z. B. "62.2e-4"
Ganzzahliger Wert	Ziffernfolge ohne Tausendertrennzeichen, z. B. "125985"

7.6.4.7 Schleifen

DO-Schleife

Syntax:
```
do
        Anweisungsliste;
while (Bedingung);
```

Beispiel:
```
i = 0;
do
{
```

```
        i++;
        v[i] = 5 * i;
}
while (i < 10)
```

FOR-Schleife

Syntax:

```
for (Initialisierung; Abbruchbedingung; Iteration)
        Anweisungsliste
```

Im Initialisierungsteil können Variablen mit Werten vorbelegt werden. Die Variablen müssen vorher deklariert worden sein. Die Abbruchbedingung bestimmt, wie oft die Schleife durchlaufen wird. Die Iteration wird immer am Ende der Schleife ausgeführt.

Beispiel:

```
for (i = 0; i < 10; i++)
    k = k + 1;
```

WHILE–Schleife

Syntax:

```
while (Bedingung)
        Anweisungsliste;
```

Die Anweisungen der while-Schleife werden ausgeführt, solange die Bedingung wahr ergibt.

Beispiel:

```
i = 0;
while (i < 10)
{
        k++
        i++;
}
```

7.6.4.8 Verzweigungen

IF-Anweisung

Die IF-Anweisung hat folgende Syntax:

```
if (Bedingung)
        Anweisungsliste;
else
        Anweisungsliste;
```

Der *else*-Zweig ist optional und wird ausgeführt, wenn die Bedingung falsch ergibt.

SWITCH-Anweisung

Die SWITCH-Anweisung dient zum Vergleich einer Variablen mit verschiedenen Konstanten.

Syntax:

```
switch (Variable)
{
    case Konstante1:
            Anweisungsliste;
            break;
    case Konstante2:
            Anweisungsliste;
            break;
    [default:
            Anweisungsliste;
            break;]
}
```

Es können beliebig viele *case*-Blöcke gebildet werden. Das Schlüsselwort *break* ist notwendig und dient als Endemarkierung der Anweisungslisten. Der *default*-Block ist optional.

Beispiel:

```
switch (status)
{
    case 0:
            Ausgang = Eingang;
            break;
    case 1:
            Ausgang = Faktor * Eingang;
            break;
    case 2:
            Ausgang = -1 * Faktor * Eingang;
            break;
    default:
            Ausgang = 0.0;
            break;
}
```

Die abgefragte Variable in der *switch*-Anweisung muss entweder ganzzahlig sein oder ein Aufzählungstyp. Im Falle eines Aufzählungstyps stehen die Alternativen in der *case*-Anweisung in einfachen Anführungszeichen.

7.6.4.9 Systemfunktionen

In Blöcken können Sie Systemfunktionen aufrufen. Stellen Sie beim Aufruf einer Systemfunktion den Unterstrich "_" voran, z. B. "`_resetSimTime()`".

Folgende Systemfunktionen sind verfügbar:

- `message(C, T, D)`
  Der Text `T` wird als Meldung der Kategorie `C` ins Meldesystem eingetragen. Der Parameter `D` bestimmt, ob die Meldung kommend (`D=True`) oder gehend (`D=False`) ist. Die Kategorie und der Meldetext müssen einfache, nicht zusammengesetzte Textkonstanten sein.
  Beispiel: `_message("ERROR", "Fehlermeldung", True);`
- `printlog(T)`
  Der Text `T` wird in eine vom Skript geöffnete Datei geschrieben.
  In dieser Funktion dürfen zusätzlich auch Textparameter, Eingangsgrößen sowie Zustandsgrößen verwendet werden. In diesem Fall werden die aktuellen Werte zum Zeitpunkt des Aufrufs übergeben.
  Beispiel: `_printlog("Berechneter Wert: " + Z + "Sekunden.");`
- `resetSimTime()`
  Die Simulationszeit wird auf Null zurückgesetzt.
- `Tools.StringLength(S)`
  Liefert die Länge der Textvariablen `S` als Integer-Wert.
- `Tools.Substring(S, I, L)`
  Liefert einen Teil der Textvariablen `S`, beginnend an der Stelle `I` und mit der Länge `L`.
- `Tools.CharAt(S, I)`
  Liefert das Zeichen der Textvariablen `S` an der Stelle `I`. Der Rückgabewert ist vom Typ `char`. Um den Zahlencode des Zeichens zu erhalten, benutzen Sie anschließend die Funktion `_trunc2byte()`.
- `Tools.ConvertToInteger(S, I, L, B)`
  Liefert den Zahlenwert, der durch die Textvariable `S` repräsentiert wird, wobei nur der Teil ab der Stelle `I` mit Länge `L` berücksichtigt wird. `B` gibt die Zahlenbasis an, in der diese Zahl kodiert ist (2, 8, 10 oder 16).
- `Tools.ConvertToInteger(S, B)`
  Liefert den Zahlenwert, der durch die Textvariable `S` repräsentiert wird. `B` gibt die Zahlenbasis an, in der diese Zahl kodiert ist (2, 8, 10 oder 16).

Weitere Funktion finden Sie im Kapitel: Funktionsaufrufe mathematischer Standardfunktionen (Seite 376)

7.6.4.10 Operatoren

Die folgenden Operatoren stehen Ihnen innerhalb der Funktionen und Blöcken zur Verfügung. Sie sind in folgender Tabelle von höchster nach niedrigster Priorität geordnet. Wenn mehrere Operatoren zusammen in einem Abschnitt erscheinen, besitzen sie gleiche Priorität.

Tabelle 7-14 Operatoren

Operation	Schreibweise
Klammerung	(Ausdruck)
Funktionsaufruf	Funktionsname (Parameterliste)
Inkrement	I++
Dekrement	I−−
Inkrement	++I
Dekrement	−−I
Unäres Plus (Vorzeichen)	+Z
Arithmetische Negation	−Z
Bitweises Komplement	~I
Logische Negation	!B
Multiplikation	Z * Z
Division	Z / Z
Modulo-Rest	Z % Z
Addition	Z + Z
Subtraktion	Z − Z
Bitweise links schieben	I << J
Bitweise rechts schieben	I >> J
Vergleich auf kleiner	Z < Z
Vergleich auf größer	Z > Z
Vergleich auf kleiner oder gleich	Z <= Z
Vergleich auf größer oder gleich	Z >= Z
Gleichheit	Z == Z oder B == B
Ungleichheit	Z != Z oder B != B
Bitweise-AND	I & I
Bitweise-exclusiv-OR	I ^ I
Bitweise-inclusiv-OR	I | I
Logisches AND	B && B
Logisches OR	B || B
Bedingter Ausdruck	B ? Ausdruck : Ausdruck

Operation	Schreibweise
Einfache Zuweisung	B = B oder Z = Z
Zusammengesetzte Zuweisungen	Z *= Z Z /= Z Z += Z Z −= Z I <<= J I >>= J I &= I I ^= I I |= I

Der Typ der zulässigen Operanden ist eingeteilt in die gelisteten Gruppen der folgender Tabelle:

Tabelle 7-15 Datentyp der Operanden

Bezeichnung	Datentypen
B	binary, bool
I	integer, sbyte, byte, short, ushort, int, uint, long, ulong
J	sbyte, byte, short, ushort, int
Z	analog, float, double, integer, sbyte, byte, short, ushort, int, uint, long, ulong

Bei allen Zuweisungen ist zu beachten, dass nicht von einem größeren in einen kleineren Datentyp zugewiesen werden kann. Die folgende Tabelle zeigt die erlaubten Zuweisungen.

Tabelle 7-16 Datentypwandlung bei Zuweisung

Zieldatentyp	Erlaubte Zuweisungsdatentypen
binary, bool	binary, bool
sbyte	sbyte
byte	byte
short	short, byte, sbyte
ushort	ushort, byte
int	int, short, ushort, sbyte, byte
uint	uint, ushort, byte
integer, long	integer, long, int, uint, short, ushort, sbyte, byte
ulong	ulong, uint, ushort, byte
float	float, integer, long, ulong, int, uint, short, ushort, sbyte, byte
analog, double	analog, double, float, integer, long, ulong, int, uint, short, ushort, sbyte, byte

Zur Umwandlung des Datentyps stehen Ihnen Funktionen zum Runden bzw. Abschneiden des ganzzahligen Anteils zur Verfügung. Weitere Informationen finden Sie im Abschnitt: Funktionsaufrufe mathematischer Standardfunktionen (Seite 376).

7.6.4.11 Zugriff auf Zustandsgrößen

Aus Funktionen heraus können Sie nicht auf Zustandsgrößen zugreifen. Zustandsgrößen müssen Sie gegebenenfalls im Funktionsaufruf übergeben. In Blöcken haben Sie ausschließlich Zugriff auf diskrete Zustandsgrößen. Für den Zugriff ist die in folgender Tabelle gezeigte Notation zu verwenden.

Tabelle 7-17 Zugriff auf diskrete Zustandsgrößen

Notation	Zugriff
Z	Wert aus dem vorherigen Rechenzyklus
@Z	Neuer Wert

7.6.5 Interne Variablen und Konstanten

Es stehen Ihnen die in folgender Tabelle gelisteten Systemgrößen zur Verfügung, die Sie als Konstanten verwenden können.

Tabelle 7-18 Systemkonstanten

Name	Datentyp	Beschreibung
_NAME	text	Der Name der Komponenteninstanz
_TA	analog	Projektierte Abtastzeit (=Zykluszeit) in Mikrosekunden
_Time	analog	Aktuelle Simulationszeit in Millisekunden
_ta	analog	Projektierte Abtastzeit (=Zykluszeit) in Sekunden
_load	analog	– reserviert –
_scriptmode	binary	Ist "True", solange ein Skript ausgeführt wird
_PI	analog	Kreiszahl PI (3.14159…)
_E	analog	Eulersche Zahl (2.71828...)
_GRAVITY	analog	Erdbeschleunigung (9.81)

In Typen von Fördertechnikkomponenten können Sie auch die in folgender Tabelle aufgeführten Systemvariablen verwenden. Diese Variablen ermöglichen es Ihnen, die Abmessungen einer Komponente in der Verhaltensbeschreibung auszuwerten.

Tabelle 7-19 Systemvariablen zur Bestimmung der Komponentenabmessung

Systemvariable	Datentyp	Bedeutung
_WIDTH	analog	Breite (unskaliert) der Komponente in Pixel
_HEIGHT	analog	Höhe (unskaliert) der Komponente in Pixel
_SCALEX	analog	Horizontale Skalierung der Komponente
_SCALEY	analog	Vertikale Skalierung der Komponente
_TECHSCALE	analog	Maßstab des Planes, auf dem die Komponente liegt (Anzahl mm pro Pixel)

Tabelle 7-20 Systemvariablen zur Bestimmung der Systemzeit

Systemvariable	Datentyp	Bedeutung
_t_sec	integer	Sekunde
_t_min	integer	Minute
_t_hour	integer	Stunde
_t_day	integer	Tag
_t_mon	integer	Monat
_t_year	integer	Jahr

7.6.6 Der Parametertyp characteristic

Parameter können den Datentyp *characteristic* besitzen. Sie können damit eine oder auch mehrere Kennlinien in einer Komponente verwenden. Beachten Sie, dass Sie keinen Vektor von Kennlinien anlegen können, die "Anzahl" eines solchen Parameters also immer auf "1" steht.

Die Verwendung eines solchen Parameters (P), der einen Eingangswert (IN) auf einen Ausgangswert (OUT) abbildet, unterscheidet sich je nach verwendetem Beschreibungsansatz:

- **Anweisungsorientiert:**
  ```
  OUT = _characteristic(P, IN);
  ```
- **Gleichungsorientiert:**
  ```
  OUT = _characteristic(P[ALL], IN);
  ```

Bibliotheken 8

8.1 Basisbibliothek

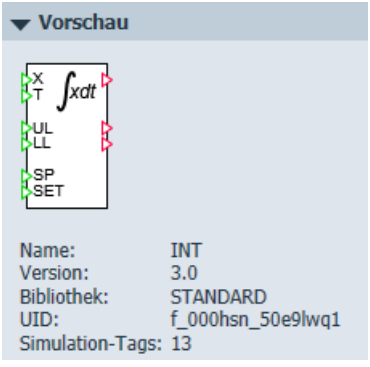
8.1.1 Allgemeines

8.1.1.1 Einführung

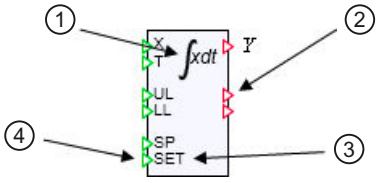
Die Basisbibliothek von SIMIT beinhaltet elementare Funktionen zur Erstellung von Simulationen, d. h. zur Modellierung von Anlagen- und Maschinenverhalten. Diese Funktionen werden in Form von Komponententypen und Controls zur Verfügung gestellt. In den folgenden Kapiteln werden die einzelnen in der Basisbibliothek von SIMIT enthaltenen Komponententypen und Controls ausführlich erläutert.

8.1.1.2 Symbole der Komponenten

Zur Erstellung einer Simulation werden Komponententypen als Komponenten instanziiert. Ziehen Sie dazu den gewünschten Komponententyp mit der Maus per "Drag & Drop" auf ein Diagramm. Jede Instanz einer Komponente wird auf einem Diagramm durch ein typspezifisches Symbol dargestellt. Das Symbol eines Komponententyps, sein Name, seine Version, die Bibliothek, zu der er gehört, seine UID und die Anzahl der Simulation-Tags werden in der Vorschau in der folgenden Abbildung angezeigt. Selektieren Sie dazu den Komponententyp in der Bibliothek per Mausklick.

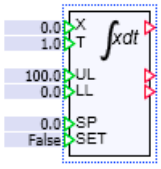


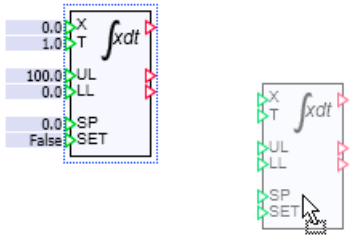
Jedes Symbol besitzt Anschlüsse mit Namen und eine Beschriftung oder Grafik, die die Funktion der Komponente auf Plänen leicht erkennen lässt (siehe folgende Abbildung). Die Symbole sind so gestaltet, dass sich die Funktion der Komponente wie auch der Anschlüsse intuitiv erfassen lässt.



① Beschriftung / Grafik
② Anschluss Ausgang
③ Name des Anschlusses
④ Anschluss Eingang

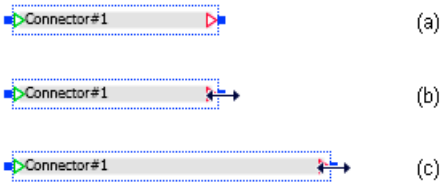
Komponenten werden somit auf Diagrammen durch das typspezifische Symbol repräsentiert. Zur Selektion einer Komponente auf dem Diagramm klicken Sie auf das Symbol. Das Symbol der selektierten Komponente wird dann mit einem blauen Selektionsrahmen dargestellt und kann mit gedrückter Maustaste auf dem Diagramm verschoben werden.



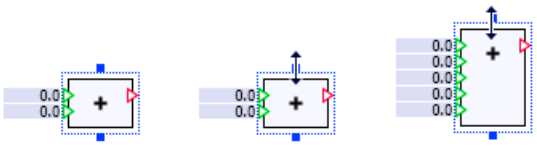


Einige Komponenten besitzen Anfasser am Selektionsrahmen. Über diese Anfasser kann die Größe des Symbols verändert werden:

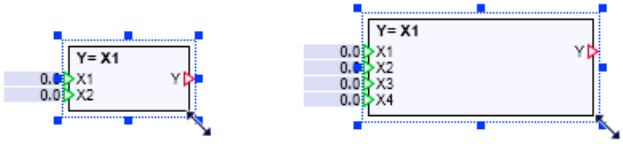
Komponenten wie Konnektoren haben Anfasser links und rechts am Selektionsrahmen wie in der Abbildung unten unter (a). Der Cursor ändert über den Anfassern seine Darstellung wie in der Abbildung unten unter (b) zu sehen. Mit gedrückter linker Maustaste können Sie den Anfasser verschieben und so die Breite des Symbols anpassen, wie in der Abbildung unten unter (c) gezeigt.



Bei den Komponenten wie beispielsweise *ADD* hat der Selektionsrahmen unten und oben Anfasser. Über diese Anfasser wird mit der Höhe des Symbols die Anzahl der Eingänge eingestellt. Ziehen Sie dazu den unteren oder oberen Anfasser mit gedrückter linker Maustaste nach unten oder oben.

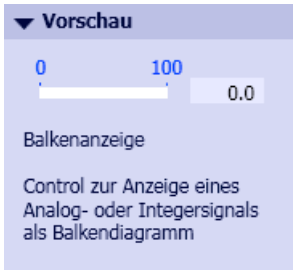


Die Formelkomponenten haben Anfasser an allen Seiten und Ecken des Selektionsrahmens. Damit können Sie sowohl die Breite wie auch die Anzahl der Eingänge einstellen. Mit den Anfassern an den Ecken des Symbols können Sie diese beiden Einstellungen gleichzeitig vornehmen.

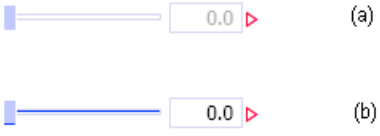


8.1.1.3 Symbole der Controls

Controls werden bei der Erstellung einer Simulation wie Komponententypen gehandhabt, also wie diese mit ihrem Symbol auf einem Diagramm platziert. In der Bibliothek selektierte Controls werden in der Vorschau mit ihrem Symbol, ihrer Bezeichnung und einer kurzen Charakterisierung ihrer Funktion angezeigt.



Da Controls nach dem Starten der Simulation als aktive Elemente fungieren, werden sie dann durch ihre Symbole entsprechend als aktive Controls dargestellt wie in der folgenden Abbildung unter (b). Ist keine Simulation aktiv, dann repräsentieren die Symbole passive Controls und werden dargestellt wie in der folgenden Abbildung unter (a).



Controls werden wie Komponenten auf Diagrammen durch das typspezifische Symbol repräsentiert. Zur Selektion klicken Sie einfach auf das Symbol. Das Symbol des selektierten Controls wird dann mit einem blauen Selektionsrahmen dargestellt und kann mit gedrückter linker Maustaste auf dem Diagramm verschoben werden. Über Anfasser am Selektionsrahmen kann die Größe des Symbols verändert werden.

8.1.1.4 Aufbau des Simulationsmodells

Ein Simulationsmodell in SIMIT wird durch das Platzieren von Komponenten aus den verfügbaren Bibliotheken (vordefiniert oder selbstdefiniert) auf Diagrammen erstellt. Die Anschlüsse der einzelnen Komponenten werden dann miteinander verbunden. Dabei können immer nur Anschlüsse des gleichen Typs miteinander verbunden werden.

Anschlüsse an das Flussnetz werden durch einen schwarzen nicht ausgefüllten Kreis an den Komponenten dargestellt. Verbundene Anschlüsse an das Flussnetz werden durch graue ausgefüllte Kreise dargestellt. Wenn mehrere Komponenten über einen Anschluss verbunden werden, wird automatisch ein interner Knoten erzeugt. Dieser wird durch einen schwarzen Punkt in der Verbindungslinie dargestellt. Alle Flussnetzanschlüsse müssen verbunden werden.

Andere Anschlüsse werden über Dreiecke dargestellt.

- Ein rotes unausgefülltes Dreieck bedeutet einen unverbundenen Ausgang.
- Ein rotes gefülltes Dreieck bedeutet einen verbundenen Ausgang.
- Ein grünes unausgefülltes Dreieck bedeutet einen unverbundenen Eingang.
- Ein grünes ausgefülltes Dreieck bedeutet einen verbundenen Eingang.

Eingänge und Ausgänge müssen nicht verbunden werden.

8.1.1.5 Anschlüsse von Komponenten

Anschlüsse von Komponenten der Basisbibliothek sind Eingänge oder Ausgänge. Eingänge sind als grüne Dreiecke links, Ausgänge als rote Dreiecke rechts am Symbol angeordnet. Um die Wirkungsrichtung der Anschlüsse optisch zu unterstreichen, sind die Dreiecke für Eingänge in das Symbol gerichtet, die Dreiecke für Ausgänge zeigen aus dem Symbol.

Funktional zusammengehörende Ein- und Ausgänge sind im Symbol einer Komponente soweit möglich gegenüberliegend angeordnet. Im obigen Beispiel des Integrators sind dies beispielsweise Eingang *X* und Integratorausgang *Y*. Des Weiteren sind funktional zusammengehörende Ein- und Ausgänge gruppiert und durch Leerplätze von anderen Gruppen abgesetzt. Dadurch sind die über die Verschaltung von Komponenten hergestellten funktionalen Zusammenhänge auf Plänen leichter zu erfassen. Im Beispiel aus dem Kapitel: Symbole der Komponenten (Seite 391) sind folgende drei Gruppen gebildet:

- die Eingänge *X* und *T* zur Berechnung des Integralwerts am Ausgang *Y*,
- die Begrenzungen *UL* und *LL* mit ihren binären Rückmeldungen und
- Setzwert *SP* und Setzbefehl *SET* zum Setzen des Integratorausgangs.

Die durch das Integral

$$Y = \frac{1}{T} \int X \, \mathrm{d}t$$

beschriebene Funktion ist damit leicht den Anschlüssen *X*, *T* und *Y* zuordenbar. Im Symbol sind Anschlüsse nur dann mit einem Namen versehen, wenn die Funktion des Anschlusses nicht offensichtlich ist.

Alle Ein- und Ausgänge sind entweder binäre (logische), analoge oder ganzzahlige (Integer) Ein- bzw. Ausgänge. Komplexe Anschlusstypen werden in den Komponenten der

Basisbibliothek nicht verwendet. Eine Ausnahme bildet der für die Verbindung zwischen Kopf- und Rumpfkomponente in der PROFIdrive-Bibliothek verwendete Anschluss vom Typ *PROFIdrive*. Die Werte der binären Ein- bzw. Ausgänge werden mit null und eins oder äquivalent mit False und True bezeichnet.

8.1.1.6 Anschlüsse von Controls

Controls zur Eingabe besitzen nur einen Ausgang als Anschluss, der als rotes Dreieck an der rechten Seite des Symbols angeordnet ist (siehe folgende Tabelle). Ein grünes Dreieck an der linken Seite des Symbols kennzeichnet den Eingang an den Controls zur Anzeige. Das Control Signaltrenner hat lediglich einen stets unsichtbaren Anschluss.



Eingabe-Control Schieber



Ausgabe-Control Balkenanzeige

Auch bei Controls sind wie bei Komponenten die Anschlüsse entweder binäre (logische), analoge oder ganzzahlige (Integer) Ein- bzw. Ausgänge.

8.1.1.7 Verbinden von Anschlüssen

Die Anschlüsse von Controls und Komponenten können unter Einhaltung der folgenden Regeln miteinander verbunden werden:

1. Es können nur Eingänge mit Ausgängen verbunden werden.
2. Ein Ausgang kann immer nur mit einem Eingang, ein Eingang aber mit mehreren Ausgängen verbunden werden.
3. Zu verbindende Anschlüsse müssen vom gleichen Typ sein.

Anschlüsse können auf unterschiedliche Arten verbunden werden:

- Verbindungslinien
- das Aufeinanderlegen von Anschlüssen
- durch implizite Verbindungen

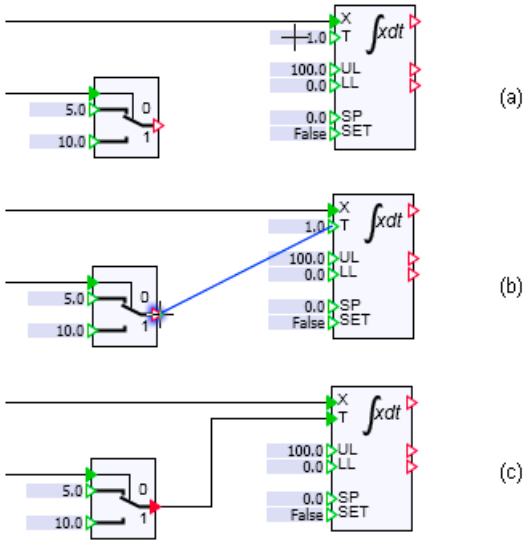
In den ersten beiden Fällen erfolgt das Verbinden grafisch im Arbeitsbereich des Diagrammeditors. Der Diagrammeditor ist dazu so konstruiert, dass die oben definierten Regeln für das Verbinden von Anschlüssen automatisch eingehalten werden.

Implizite Verbindungen werden über Einstellungen in den Eigenschaften der zu verbindenden Eingänge oder Ausgänge von Komponenten hergestellt.

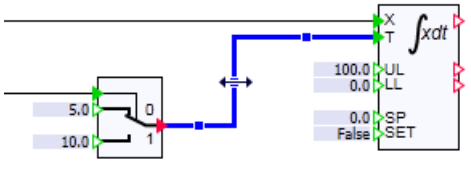
Verbinden mit Verbindungslinien

Soll beispielsweise eine Verbindung zwischen dem Ausgang der *Selection*-Komponente und dem Eingang *T* des Integrators hergestellt werden, dann bewegen Sie den Cursor über einen der beiden Anschlüsse. Sobald er seine Darstellung in ein Kreuz ändert, wie in der Abbildung unten unter (a), kann die Verbindung mit einem Mausklick oder durch Niederdrücken der Maustaste aufgenommen werden. Wenn Sie dann den Cursor bewegen, zeigt ein blaues Gummiband die Verbindung des Anschlusses mit dem Cursor an.

Bewegen Sie nun den Cursor über den zu verbindenden Anschluss. Wenn der zu verbindende Anschluss optisch hervorgehoben wird, wie in der Abbildung unten unter (b) kann die Verbindung abgeschlossen werden. Haben Sie beim Aufnehmen der Verbindung die Maustaste gedrückt gehalten, dann lassen Sie zum Schließen der Verbindung die Taste wieder los. Haben Sie die Verbindung mit dem Mausklick aufgenommen, dann klicken Sie zum Schließen der Verbindung auf den hervorgehobenen Anschluss. Ist die Verbindung geschlossen, dann erscheint anstelle des Gummibandes eine jetzt automatisch rechtwinklig eingezeichnete Verbindungslinie und die Dreiecke der verbundenen Anschlüsse werden farbig gefüllt dargestellt, wie in der folgenden Abbildung unter (c) gezeigt.



Zum Löschen von Verbindungslinien klicken Sie zunächst mit der Maus auf die zu löschende Verbindungslinie. Die Verbindungslinie wird dann als dicke blaue Linie wie in der folgenden Abbildung dargestellt und kann nun über die Menüleiste mit "Bearbeiten > Ausschneiden" oder mit der Löschtaste "Entf" gelöscht werden.

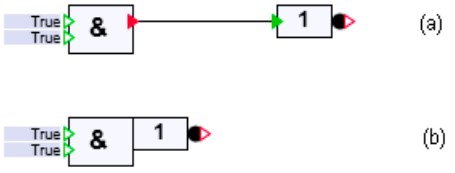


Wird eine Verbindungslinie ausgewählt, kann diese Verbindung manuell editiert werden.

Verbinden durch Aufeinanderlegen von Anschlüssen

Verbinden durch Aufeinanderlegen von Anschlüssen geschieht dadurch, dass zwei zu verbindende Komponenten und/oder Controls so auf dem Diagramm platziert werden, dass der Eingang der einen Komponente direkt auf den zu verbindenden Ausgang der anderen Komponente zu liegen kommt.

In der folgenden Abbildung ist diese Verbindungsart (b) dem Verbinden mit Verbindungslinien (a) gegenübergestellt. Die beiden Anschlüsse, die durch Aufeinanderlegen verbunden sind, werden unsichtbar.



Hinweis

Wenn beim Verbinden durch Aufeinanderlegen die Anschlüsse nicht unsichtbar werden, sind die Anschlüsse nicht vom gleichen Typ und können damit nicht verbunden werden.

Implizite Verbindungen

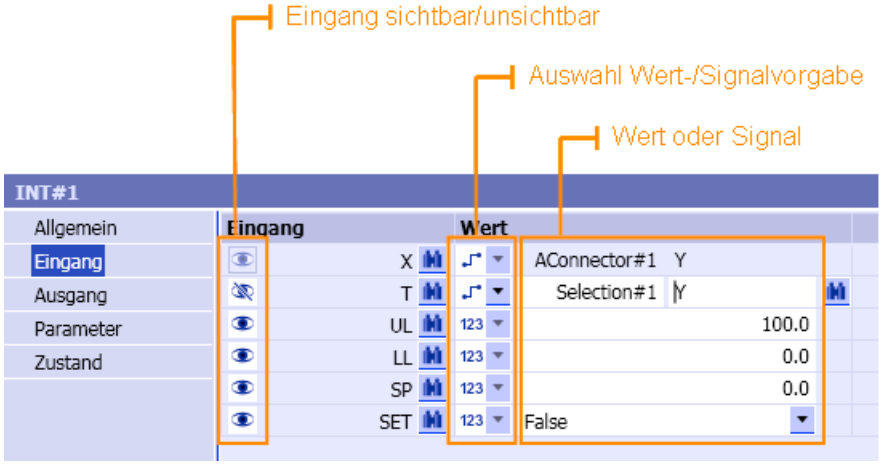
Implizite Verbindungen werden über Einstellungen in den Eigenschaften der zu verbindenden Ein- und Ausgänge von Komponenten und/oder Controls hergestellt. Öffnen Sie dazu das Eigenschaftsfenster der Komponente (s. Abbildung unten) und gehen Sie dann wie folgt vor:

1. Schalten Sie den zu verbindenden Eingang bzw. Ausgang unsichtbar ( ).
   Durch Anklicken des Symbols bzw. können Sie den Eingang zwischen sichtbar und unsichtbar umschalten.

   Hinweis

   Ein unsichtbarer Eingang bzw. Ausgang wird am Symbol der Komponente nicht angezeigt und kann somit nicht mehr über Verbindungslinien mit einem anderen Ausgang bzw. Eingang verbunden werden.

2. Stellen Sie das Auswahlfeld für die Wert-/Signalvorgabe auf Signalvorgabe ( ).
3. Geben Sie das zu verbindende Signal mit Komponentenname (Quelle) und Anschlussname an.



Analog gehen Sie bei einem Control vor. Da ein Control keine Auswahl für Wert-/Signalvorgabe bietet, entfällt der zweite Schritt; der zu verbindende Ausgang kann direkt nach dem Umschalten auf unsichtbar eingegeben werden.

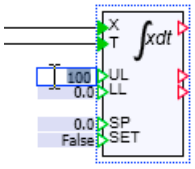
8.1.1.8 Setzen von Eingängen

Nicht verbundene Eingänge können mit Werten vorbelegt werden. Der Wert kann im

- Anschlussfeld des Eingangs auf dem Diagramm oder im
- Eigenschaftsfenster der Komponente

eingegeben werden.

Mit Doppelklick in das Anschlussfeld öffnen Sie das Feld zur Eingabe des Wertes (s. folgende Abbildung). Die Eingabe kann mit der Return-Taste oder durch einen Mausklick in das Diagramm außerhalb des Anschlussfeldes abgeschlossen werden.



Zur Eingabe eines Werts im Eigenschaftsfenster navigieren Sie zu dem entsprechenden Eingang und öffnen das Feld zur Eingabe durch Mausklick in das Eingabefeld (s. folgende Abbildung). Die Eingabe kann mit der Return-Taste oder durch einen Mausklick außerhalb des Eingabefelds im Eigenschaftsfenster abgeschlossen werden.



Bei binären Parametern sind die Eingabe von True und 1 bzw. False und 0 äquivalent. Binärwerte werden immer als True bzw. False angezeigt.

Das Setzen von Eingängen auf die oben beschriebenen Weisen ist auch möglich, wenn die Simulation gestartet ist. Allerdings sind Eingaben nur für die Dauer der gestarteten Simulation wirksam, d. h. geänderte Eingangswerte werden mit dem Beenden der Simulation wieder auf die ursprünglichen Werte gesetzt.

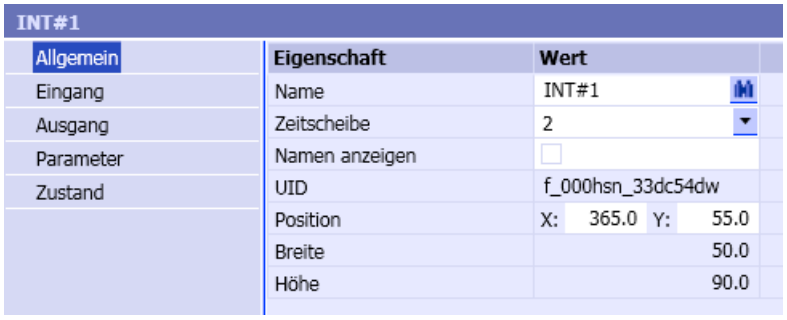
8.1.1.9 Eigenschaften von Komponenten

Die Eigenschaften von Komponenten sind im Eigenschaftsfenster zugänglich. Um die Eigenschaften einer Komponente im Eigenschaftsfenster anzuzeigen, klicken Sie mit der linken oder rechten Maustaste auf die Komponente. Bei laufender Simulation können die Eigenschaften nur mit einem Klick der rechten Maustaste eingeblendet werden. Die Eigenschaften einer Komponente sind unterteilt in:

- Allgemeine Eigenschaften
- Eigenschaften der Eingänge
- Eigenschaften der Ausgänge
- Parameter
- Zustände

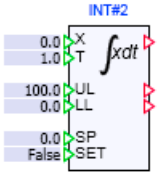
Allgemeine Eigenschaften

Allgemeine Eigenschaften von Komponenten sind der Name, die Zeitscheibe der Komponente, die eindeutige Kennung (UID) des Komponententyps, die Position der Komponente sowie die Breite und Höhe des Symbols.



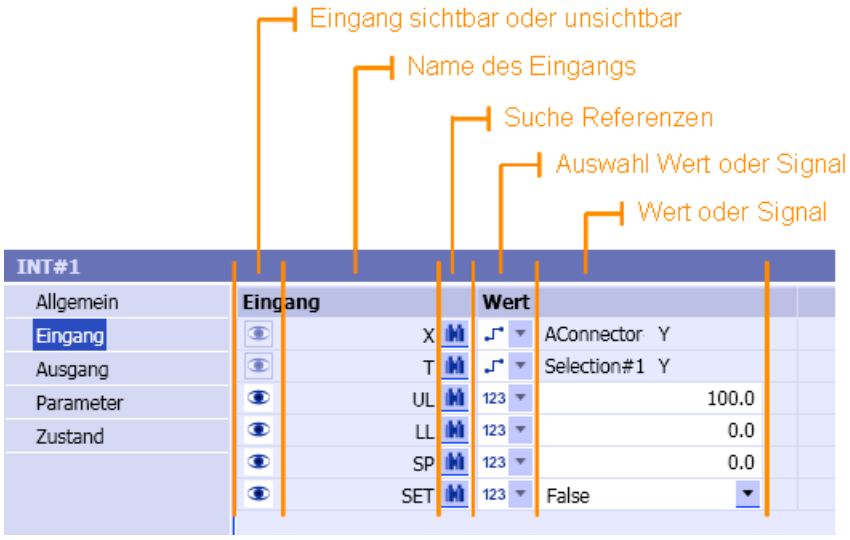
Der Name muss eindeutig für alle im Projekt verwendeten Komponenten und Controls sein, d. h. im Projekt dürfen mehrere Komponenten und/oder Controls nicht den gleichen Namen haben. Wenn Sie eine Komponente aus der Bibliothek auf ein Diagramm ziehen wird automatisch ein Name vergeben. Dieser Name wird aus der Bezeichnung des Komponententyps und einer projektweit eindeutigen Nummer für den Komponententyp gebildet.

Das Setzen des Kontrollkästchens *Namen anzeigen* bewirkt, dass der Name der Komponente auf dem Diagramm angezeigt wird.



Eigenschaften der Eingänge

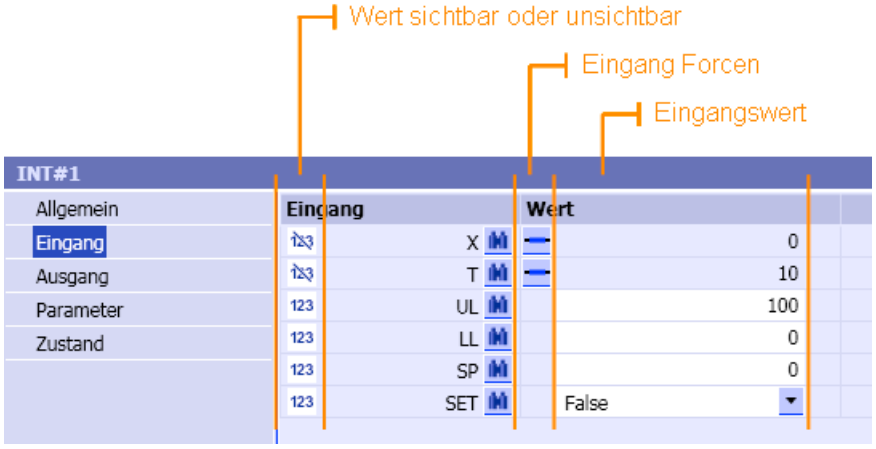
Eingänge können sichtbar oder unsichtbar sein und es kann das mit einem Eingang verbundene Signal oder der Eingangswert gesetzt werden.



Jeder Eingang besitzt die folgenden Eigenschaften:

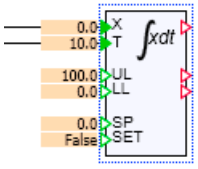
- **Sichtbarkeit**
  In der ersten Spalte kann zwischen Sichtbarkeit ( ) und Unsichtbarkeit ( ) des Eingangs umgeschaltet werden. Diese Umschaltung kann nur vorgenommen werden, wenn der Eingang nicht über eine Verbindungslinie angeschlossen ist. In der Abbildung oben sind beispielsweise die beiden ersten Eingänge *X* und *T* verbunden; sie können somit nicht unsichtbar gesetzt werden.
- **Name**
  In der zweiten Spalte wird der Name des Eingangs rechtsbündig angezeigt.
- **Referenzen**
  In der dritten Spalte kann nach Referenzen, d. h. nach Objekten, die diesen Eingang verwenden, gesucht werden.
- **Auswahl Wert oder Signal**
  In der vierten Spalte kann über eine Auswahl auf Wertvorgabe ( ) oder Signalvorgabe ( ) für den Eingang umgeschaltet werden. Ist der Eingang über eine Verbindungslinie angeschlossen, dann ist diese Auswahl nicht aktiv.
- **Wert oder Signal**
  In der fünften Spalte kann bei Wertvorgabe der Eingangswert gesetzt werden. Bei Signalvorgabe wird hier der über die Verbindungslinie angeschlossene Ausgang angezeigt, oder es kann der implizit zu verbindende Ausgang gesetzt werden.

Bei laufender Simulation ändert sich die Darstellung der Eigenschaften.



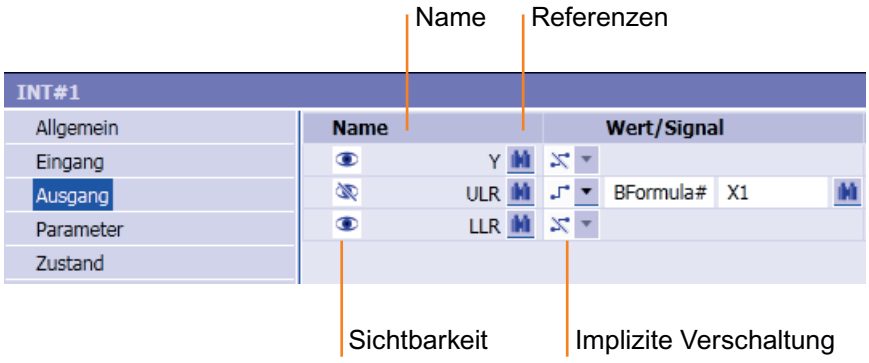
Anstelle von Signalen werden jetzt immer Eingangswerte angezeigt, und es ist möglich, jeden Eingangswert zu setzen:

- **Anzeige Ein/Aus**
  In der ersten Spalte kann die Anzeige des Werts am Eingang der Komponente auf dem Diagramm ein- (123) und ausgeschaltet (123) werden. Beim Starten der Simulation wird die Anzeige für nicht verbundene Eingänge eingeschaltet und für verbundene Eingänge ausgeschaltet. Für unsichtbare Eingänge kann die Anzeige nicht eingeschaltet werden. In der folgenden Abbildung ist eine Komponente dargestellt, bei der die Anzeige für alle Eingangswerte, also auch für die Werte an verbundenen Eingängen, eingeschaltet ist:



- **Eingang forcen**
  In der vierten Spalte kann für jeden verbundenen Eingang das Forcen ein- (⊣⊢) und ausgeschaltet (—) werden.
- **Eingangswert**
  In der fünften Spalte wird der Eingangswert angezeigt bzw. gesetzt.

Eigenschaften der Ausgänge



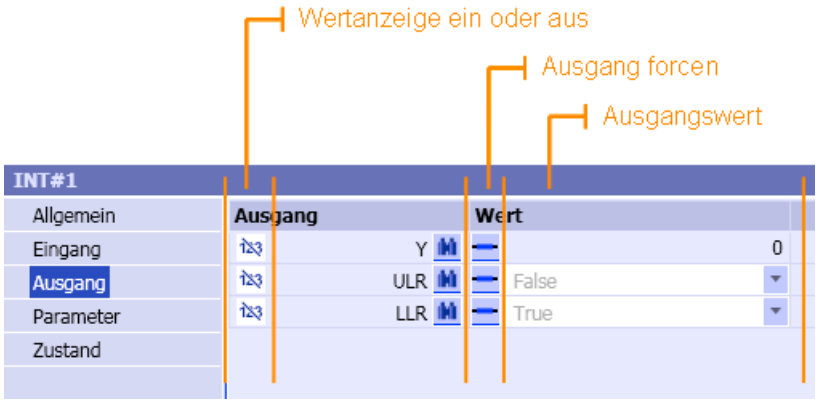
Jeder Ausgang besitzt die folgenden Eigenschaften:

- **Sichtbarkeit**
  In der ersten Spalte kann zwischen Sichtbarkeit ( ) und Unsichtbarkeit ( ) des Ausgangs umgeschaltet werden. Diese Umschaltung kann nur vorgenommen werden, wenn der Ausgang nicht über eine Verbindungslinie oder durch Aufeinanderlegen von Anschlüssen angeschlossen ist.
- **Name**
  In der zweiten Spalte wird der Name des Ausgangs rechtsbündig angezeigt.
- **Referenzen**
  In der dritten Spalte kann nach Referenzen, d. h. nach Objekten, die diesen Ausgang verwenden gesucht werden.
- **Implizite Verschaltung**
  In der vierten Spalte kann eingestellt werden, ob dieser Ausgang implizit verschaltet sein soll.

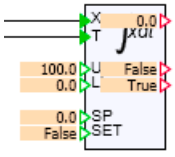
	Ausgang ist implizit verschaltet
	Ausgang ist nicht implizit verschaltet

Wenn die implizite Verschaltung ausgewählt wird, öffnet sich ein Eingabefeld. Tragen Sie hier das zu verschaltende Signal mit Quelle und Namen ein.

Bei laufender Simulation ändert sich die Darstellung der Eigenschaften.



- **Anzeige Ein/Aus**
  In der ersten Spalte kann die Anzeige des Werts am Ausgang der Komponente auf dem Diagramm ein- (123) und ausgeschaltet (123). Beim Starten der Simulation wird die Anzeige für alle Ausgänge ausgeschaltet. Für unsichtbare Ausgänge kann die Anzeige nicht eingeschaltet werden. In der folgenden Abbildung ist eine Komponente dargestellt, bei der die Anzeige für alle Ausgangswerte eingeschaltet ist.



- **Ausgang forcen**
  In der vierten Spalte kann für jeden verbundenen Ausgang das Forcen ein- (-I-) und ausgeschaltet (—) werden.
- **Ausgangswert**
  In der fünften Spalte wird der Ausgangswert angezeigt bzw. gesetzt.

Anzeige der Ein- und Ausgangswerte von Komponenten

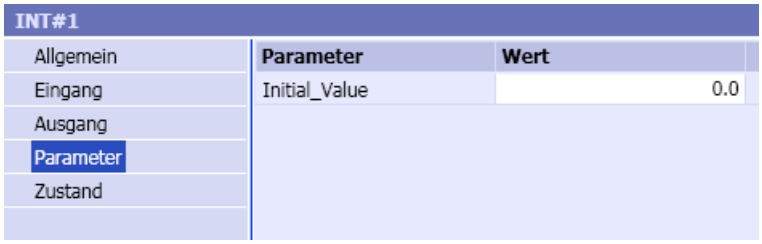
Die aktuellen Ein- und Ausgangswerte von Komponenten können bei laufender Simulation einzeln für jeden Ein- und Ausgang mit der Schaltfläche 123 im Eigenschaftsfenster der Komponente sichtbar geschaltet werden. Um alle Ein- und Ausgänge der Komponenten auf einem Diagramm mit einem Mausklick sichtbar zu schalten, können Sie den gleichen Befehl 123 in der Werkzeugleiste des Diagramms benutzen.



Wenn Sie vor Ausführung des Befehls eine oder mehrere Komponenten selektiert haben, wirkt diese Umschaltung nur auf die selektierten Komponenten. Falls keine Komponente selektiert ist, wirkt die Umschaltung auf alle Komponenten des Diagramms.

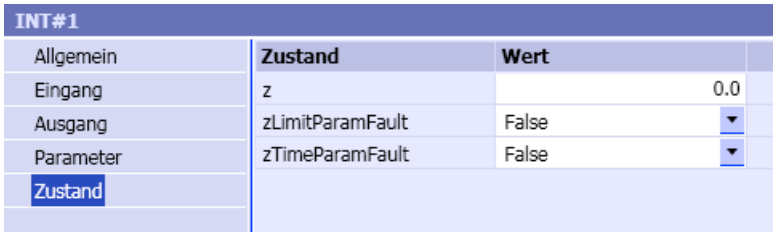
Parameter

Jeder Parameter wird mit Namen und Wert im Eigenschaftsfenster dargestellt:



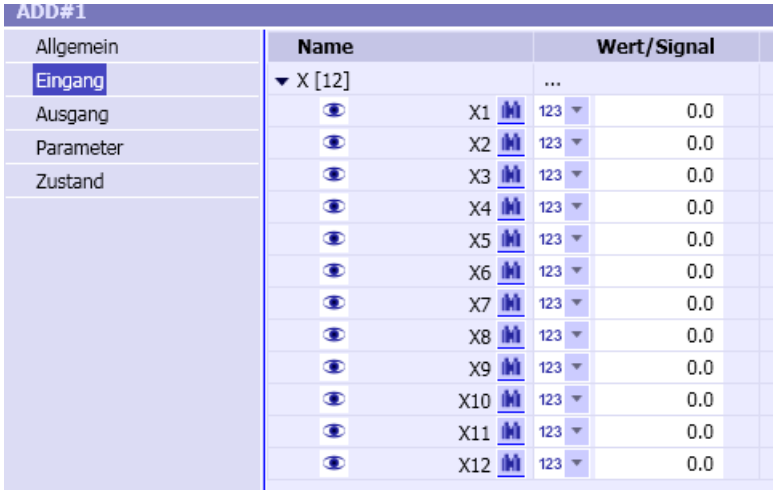
Zustände

Zustände werden mit ihrem Namen und den Initialwerten im Eigenschaftsfenster dargestellt. Wird die Simulation gestartet, dann werden für jeden Zustand die aktuellen Werte eingeblendet.



Darstellung von Vektoren im Eigenschaftsfenster

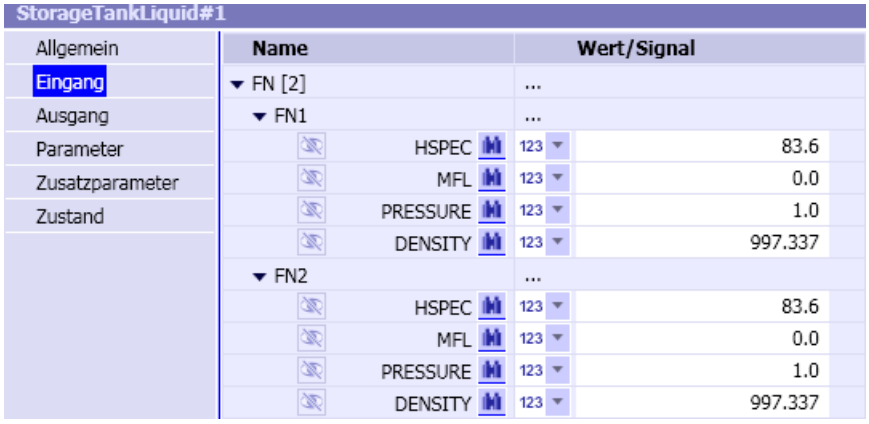
Vektoren von Ein- und Ausgängen, Parametern und Zuständen werden, wie in folgender Abbildung beispielhaft gezeigt, im Eigenschaftsfenster gruppiert und in numerisch korrekter Reihenfolge dargestellt.



Die Vektorelemente können ein- und ausgeklappt werden. In folgender Abbildung ist ein Vektor mit eingeklappten Elementen zu sehen.



Bei Anschlüssen mit komplexen Verbindungstypen wird eine weitere ein- und ausklappbare Ebene hinzugefügt:



8.1.1.10 Fehlermeldungen von Komponenten

Die Komponenten sind so implementiert, dass kritische oder unsinnige Parameter oder Eingangswerte nicht zu einem instabilen Verhalten der Komponente führen. Falls in der Parametrierung nicht erlaubte Werte angegeben werden oder falls Eingangssignale nicht im vorgesehenen Bereich liegen, wird von der Komponente eine Fehlermeldung ausgegeben.

Zusätzlich können im Fehlerfall zur Vermeidung von instabilen Ausgangswerten Ausgänge der Komponente auf einen definierten Wert gesetzt sein. Dieser Setzwert wird wieder unwirksam, wenn der Fehlerzustand aufgehoben ist.

Alle Fehlermeldungen aus Komponenten der Basisbibliothek sind der Meldekategorie *ERROR* zugeordnet.

Fehlermeldungen werden als sogenannte Kommend- und Gehend-Meldungen erzeugt. Die Kommend-Meldung wird beim Auftreten des Fehlers abgesetzt; die Gehend-Meldung wird abgesetzt, wenn der Fehlerzustand wieder aufgehoben wird. Beide Meldungen haben den gleichen Meldetext; bei Gehend-Meldungen wird der Meldetext in runde Klammern gesetzt.

Hinweis

Wenn Sie die Meldekategorie *ERROR* in Ihren eigenen Meldungen, die Sie beispielsweise mit der Komponente *Message* des Trend and Messaging Editors generieren können, verwenden, dann können Sie anhand der Meldekategorie nicht zwischen Meldungen aus Komponenten der Basisbibliothek und Ihren eigenen Meldungen unterscheiden.

8.1.1.11 Eigenschaften von Controls

Die Eigenschaften von Controls sind im Eigenschaftsfenster zugänglich. Um die Eigenschaften einer Komponente im Eigenschaftsfenster anzuzeigen, klicken Sie mit der linken oder rechten Maustaste auf die Komponente. Bei laufender Simulation können die Eigenschaften nur mit einem Klick der rechten Maustaste eingeblendet werden. Jedes Control hat

- allgemeine Eigenschaften und
- Eigenschaften für den Anschluss.

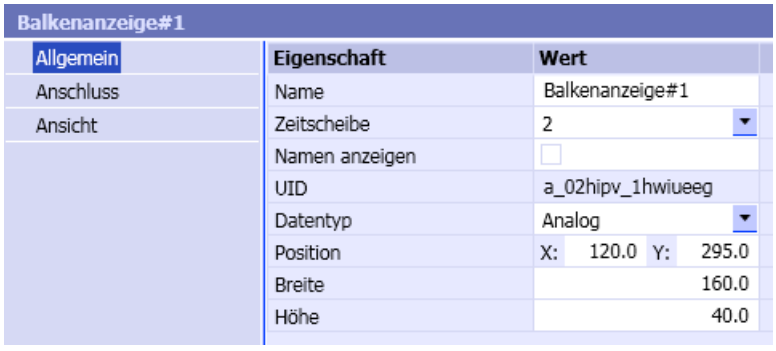
Controls, die in ihrer Darstellung verändert werden können, besitzen noch

- Eigenschaften für die Ansicht.

Weitere Informationen hierzu finden Sie im Kapitel: Controls (Seite 577).

Allgemeine Eigenschaften

Allgemeine Eigenschaften sind der Name, die Zeitscheibe, die eindeutige Kennung (UID), die Position, sowie die Breite und Höhe des Controls.



Der Name eines Controls muss eindeutig für alle im Projekt verwendeten Controls und Komponenten sein, d. h. im Projekt dürfen mehrere Controls und/oder Komponenten nicht den gleichen Namen haben. Wenn Sie ein Control aus der Bibliothek auf ein Diagramm ziehen, wird automatisch ein Name vergeben. Dieser Name wird gebildet aus der Bezeichnung des Controls und einer projektweit eindeutigen Nummer für das Control.

Das Setzen des Kontrollkästchens *Namen anzeigen* bewirkt, dass der Name des Controls auf dem Diagramm angezeigt wird.

Darüber hinaus können Controls noch spezifische weitere allgemeine Eigenschaften aufweisen. Weitere Informationen hierzu finden Sie im Kapitel: Controls (Seite 577).

Eigenschaften der Anschlüsse

Anschlüsse können sichtbar oder unsichtbar sein. In der folgenden Abbildung ist das Eigenschaftsfenster mit dem Anschluss eines Controls beispielhaft gezeigt.



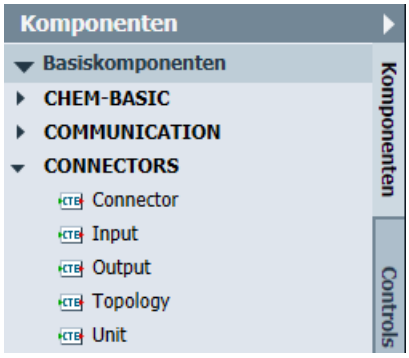
Jeder Anschluss besitzt die folgenden Eigenschaften:

- **Sichtbarkeit**
  In der ersten Spalte kann zwischen Sichtbarkeit ( ) und Unsichtbarkeit ( ) des Anschlusses umgeschaltet werden. Diese Umschaltung kann nur vorgenommen werden, wenn der Anschluss nicht über eine Verbindungslinie oder durch Aufeinanderlegen von Anschlüssen verbunden ist.
- **Name**
  In der zweiten Spalte wird der Name des Anschlusses rechtsbündig angezeigt.
- **Signal**
  In der dritten Spalte wird das verbundene Signal angezeigt bzw. kann für unsichtbare Anschlüsse gesetzt werden.

8.1.2 Konnektoren

Im Verzeichnis *CONNECTORS* der Basisbibliothek werden Konnektoren zur Verfügung gestellt:

- Ein globaler Konnektor *Connector*
- die Peripherie-Konnektoren *Input* und *Output*
- der spezielle Konnektor *Unit*
- der *Topology*-Konnektor, der nur in Zusammenhang mit speziellen SIMIT Modulen und Bibliotheken verwendet wird und daher hier nicht beschrieben wird.



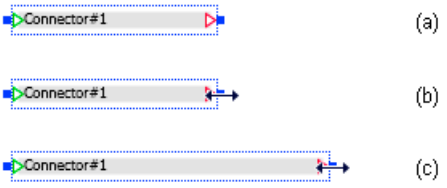
Die Konnektoren der Basisbibliothek weisen die folgenden Gemeinsamkeiten auf:

- Die Anschlüsse des globalen Konnektors der Peripherie-Konnektoren sind untypisiert. Das bedeutet, dass die Konnektoren jeweils den Verbindungstyp des Anschlusses der angeschlossenen Komponente oder des angeschlossenen Controls annehmen.

Hinweis

Eine Typprüfung, d. h. die Prüfung, ob die über Konnektoren verbundenen Anschlüsse vom gleichen Typ sind, erfolgt automatisch vor dem Starten der Simulation. Falls Konnektoren mit Anschlüssen eines falschen Typs verbunden sind, wird von der Konsistenzprüfung ein entsprechender Hinweis ausgegeben und das Starten der Simulation wird abgebrochen.

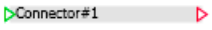
- Die Breite der Konnektorsymbole auf einem Diagramm ist einstellbar und kann damit der Länge des Konnektornamens angepasst werden. Zum Einstellen der Breite klicken Sie auf das Symbol. Es erscheint ein blauer Rahmen mit Anfassern rechts oder links (a). Wenn Sie nun den Zeiger über einem Anfasser positionieren, ändert der Mauszeiger seine Darstellung (b). Drücken Sie dann die linke Maustaste und verändern Sie die Breite mit niedergehaltener Maustaste durch eine Bewegung des Mauszeigers nach links oder rechts (c).
- Der Name des Konnektors wird im Konnektorsymbol angezeigt.



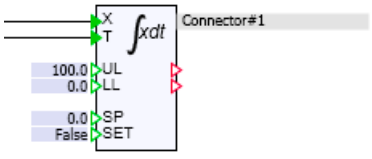
8.1.2.1 Globaler Konnektor

Der globale Konnektor *Connector* stellt Verbindungen von Komponenten und/oder Controls über Diagrammgrenzen hinweg zur Verfügung.

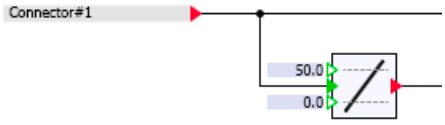
Der globale Konnektor kann sowohl als Ausgangs- als auch als Eingangskonnektor eingesetzt werden. Sein Symbol ist in folgender Abbildung dargestellt.



Wird der globale Konnektor an den Ausgang einer Komponente oder eines Controls angeschlossen, dann wird der rechte Anschluss des Konnektors eliminiert: Der Konnektor wird so zu einem Ausgangskonnektor.



Wird der globale Konnektor mit Eingängen von Komponenten und/oder Controls verbunden, dann wird der linke Anschluss des Konnektors eliminiert: Der Konnektor wird zum Eingangskonnektor.

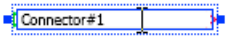


Eine Verbindung zwischen einem Ausgangskonnektor und einem oder mehreren Eingangskonnektoren wird dadurch hergestellt, dass alle Konnektoren den gleichen Namen erhalten: den Konnektornamen.

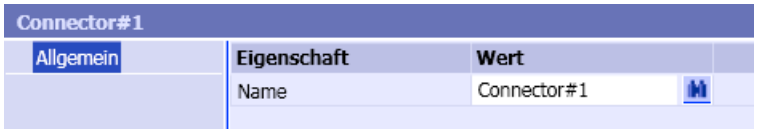
Hinweis

Globale Konnektornamen müssen im gesamten Simulationsprojekt eindeutig sein.

Wenn Sie den globalen Konnektor aus der Bibliothek auf ein Diagramm ziehen, wird der Konnektor mit einem Namen versehen, der sich aus *Connector* und einer laufenden eindeutigen Nummer zusammensetzt. Ein Konnektorname kann direkt in das Symbol eingetragen werden. Das Eingabefeld öffnet sich mit einem Doppelklick auf den Konnektor. Die Eingabe kann mit der Return-Taste oder mit einem Mausklick außerhalb des Eingabefelds abgeschlossen werden.



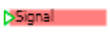
Alternativ kann der Konnektorname auch im Eigenschaftsfenster gesetzt werden. Der Konnektorname ist die einzige Eigenschaft des globalen Konnektors.



8.1.2.2 Peripherie-Konnektoren

Peripherie-Konnektoren stellen die Verbindung zu Signalen in den SIMIT-Kopplungen her. Ein Peripherie-Konnektor kann eine Verbindung zu Signalen einer jeden beliebigen SIMIT-Kopplung herstellen.

Peripherie-Konnektoren existieren als Eingangs- und Ausgangskonnektoren (*Input* und *Output*) wie nachfolgend dargestellt.



Eingangskonnektor *Input*



Ausgangskonnektor *Output*

Verbindungen von Kopplungssignalen mit Peripherie-Konnektoren sind

- Verbindung von einem oder mehreren Ausgangskonnektoren *Output* mit einem Ausgangssignal einer Kopplung oder
- Verbindung von einem Eingangssignal einer Kopplung mit einem Eingangskonnektor *Input*.

Die Verbindung wird dadurch hergestellt, dass im Eigenschaftsfenster des Peripheriekonnektors der Name der Kopplung und der Name des Kopplungssignals eingetragen werden.



Falls Sie den Kopplungsnamen auf dem Diagramm nicht dargestellt haben wollen, können Sie die Darstellung in der Option *Kopplungsnamen anzeigen* abwählen.

8.1.2.3 Topologischer Konnektor

Mit dem topologischen Konnektor können topologische Verbindungen über die jeweiligen Diagrammgrenzen hinaus realisiert werden. Dieser Konnektor ist in der CONTEC- und in der FLOWNET-Bibliothek vorhanden, die prinzipielle Funktionsweise ist dabei dieselbe.

Informationen zum topologischen Konnektor in der CONTEC-Bibliothek finden Sie im Abschnitt: Topologischer Konnektor in der CONTEC-Bibliothek (Seite 838)

Informationen zum topologischen Konnektor in der FLOWNET-Bibliothek finden Sie im Abschnitt: Topologischer Konnektor in den Bibliotheken CHEM-BASIC und FLOWNET (Seite 611).

8.1.2.4 Unit-Konnektor

Der *Unit*-Konnektor ist ein spezieller Konnektor, der nur gemeinsam mit "Komponenten für SIMATIC" oder Komponenten mit integrierter Datensatzkommunikation eingesetzt wird. Der Unit-Konnektor stellt in Zusammenhang mit diesen Komponenten Beziehungen zwischen den Komponenten und Modulen folgender Kopplungen her:

- SIMIT Unit
- Virtual Controller
- PLCSIM
- PLCSIM Advanced
- PRODAVE

Siehe auch

Verknüpfung der SIWAREXU-Komponenten mit der Kopplung (Seite 547)

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

Komponenten für SIMATIC (Seite 559)

8.1.3 Standardkomponenten

8.1.3.1 Unterscheidung der Standardkomponenten

Die Standardkomponententypen der Basisbibliothek bilden die Standardbibliothek. Sie finden sie im Verzeichnis *STANDARD.* Nach ihrer Funktion sind die Komponententypen unterteilt in Komponententypen mit

- analogen Funktionen,
- binären Funktionen,
- ganzzahligen (Integer) Funktionen,
- Konvertierungsfunktionen,
- mathematischen Funktionen und
- verschiedene Hilfsfunktionen.

8.1.3.2 Analoge Funktionen

Allgemeines zu den analogen Funktionen

Alle Komponententypen mit analogen Funktionen sind in den Verzeichnissen *AnalogBasic* und *AnalogExtended* der Standardbibliothek enthalten. Unter *AnalogBasic* sind dabei analoge Basisfunktionen eingeordnet, unter *AnalogExtended* erweiterte Analogfunktionen.

Analoge Basisfunktionen

Einleitung

Die vier grundlegenden analogen Funktionen Addition, Subtraktion, Multiplikation und Division, also die vier arithmetischen Grundoperationen, sind als Komponententypen im Verzeichnis *AnalogBasic* der Standardbibliothek abgelegt.

ADD – Addition

Symbol



Funktion

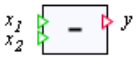
Der Komponententyp *ADD* bildet die Summe der analogen Werte an den *n* Eingängen x_1 bis x_n auf den Ausgang *y* ab:

$$y = \sum_{i=1}^{n} x_i = x_1 + x_2 + \ \ldots \ + x_n$$

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit null vorbelegt.

SUB – Subtraktion

Symbol



Funktion

Der Komponententyp *SUB* bildet die Differenz der analogen Werte an den beiden Eingängen x_1 und x_2 auf den Ausgang *y* ab gemäß

$y = x_1 - x_2$

Alle Eingänge sind mit null vorbelegt.

MUL – Multiplikation

Symbol



Funktion

Der Komponententyp *MUL* bildet das Produkt der analogen Werte an den *n* Eingängen x_1 bis x_n auf den Ausgang *y* ab:

$$y = \prod_{i=1}^{n} x_i = x_1 x_2 \ \ldots \ x_n$$

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit eins vorbelegt.

DIV – Division

Symbol



Funktion

Der Komponententyp *DIV* bildet den Quotienten der analogen Werte an den Eingängen x_1 und x_2 auf den Ausgang *y* ab gemäß

$y = x_1 / x_2$

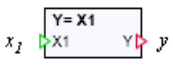
Der Eingang x_1 ist mit null, der Divisoreingang x_2 ist mit eins vorbelegt.

Der Wert des Divisors x_2 darf nicht null werden. Falls beim Durchführen der Simulation der Divisor null wird, wird die Fehlermeldung "*DIV: division by zero*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf null gesetzt.

Erweiterte analoge Funktionen

AFormula – analoge Formelkomponente

Symbol



Funktion

Der Komponententyp *AFormula* ermöglicht die Verwendung von expliziten algebraischen Funktionen. Diese Funktion *f* berechnet einen Wert in Abhängigkeit von den *n* Eingangswerten x_i. Der Funktionswert wird dem Ausgang *y* zugewiesen:

$y = f(x_1, \ldots, x_n)$

Die Anzahl *n* der Eingänge kann frei von 1 bis 32 variiert werden. Im Formelausdruck können nur die Eingänge benutzt werden, die gemäß der aktuell eingestellten Anzahl zur Verfügung stehen.

Die folgenden Operatoren sind in Formelausdrücken verwendbar:

Tabelle 8-1 Operatoren in Formelausdrücken der Komponente AFormula

Operator	Funktion
+	Addition
−	Subtraktion
/	Division
*	Multiplikation
(	Öffnende Klammer
)	Schließende Klammer
Zahlenkonstante	Gleitkommazahlen, auch in Exponentialschreibweise
Funktionsaufrufe	Mathematische Standardfunktionen

Hinweis

Verwenden Sie bei der Eingabe von Gleitkommazahlen den Punkt als Dezimaltrennzeichen.

Hinweis

In der Formelkomponente wird nicht überprüft, ob die Argumente des Formelausdrucks gültige Werte aufweisen. Tritt während der Simulation in der Berechnung des Formelausdrucks eine Division durch null auf, dann hat der Ausgang *y* den Wert **Inf**. Ist ein Argument eines Formelausdrucks während der Simulation unbestimmt, wie beispielsweise die Division null durch null, dann hat der Ausgang *y* den Wert **NaN** (not a number).

Diese irregulären Ausgangswerte setzen sich in alle von diesem Ausgang abhängige Werte fort. Ihre Simulation gerät damit in einen nicht definierten Zustand. Sorgen Sie daher zur Vermeidung dieser Situation dafür, dass die Eingänge der Formelkomponente nur Werte annehmen, die die Gültigkeit der Argumente im Formelausdruck sichern.

In der Formelkomponente wird nicht überprüft, ob im angegebenen Formelausdruck alle eingestellten Eingänge verwendet werden.

Als mathematische Standardfunktionen stehen die mathematischen Funktionen, wie in folgender Tabelle gelistet, zur Verfügung.

Tabelle 8-2 Mathematische Funktionen in Formelausdrücken der Komponente AFormula

Formelausdruck	Funktion
sqrt(x)	$y = \sqrt{x}$; $x \geq 0$
fabs(x)	$y = \lvert x \rvert$
exp(x)	$y = e^x$
pow(x, z)	$y = X^z$;
log(x)	Natürlicher Logarithmus: $y = \ln(x)$; $x > 0$
log10(x)	Dekadischer Logarithmus: $y = \lg(x)$; $x > 0$

Formelausdruck	Funktion
ceil(x)	Kleinste ganze Zahl größer oder gleich *x*
floor(x)	Größte ganze Zahl kleiner oder gleich *x*
rand()	ganzzahliger Zufallswert *y,* 0 ≤ *y* ≤ 32767
sin(x)	*y* = sin(*x*); Winkel *x* im Bogenmaß
cos(x)	*y* = cos(*x*); Winkel *x* im Bogenmaß
tan(x)	*y* = tan(*x*); Winkel *x* im Bogenmaß; *x* ≠±(2*n*+1)π/2
asin(x)	*y* = arcsin(*x*); −π/2 ≤ *y* ≤ π/2
acos(x)	*y* = arccos(*x*); 0 ≤*y* ≤ π
atan(x)	*y* = arctan(*x*); −π/2 ≤ *y* ≤ π/2
atan2(z, x)	*y* = arctan(*x* / *z*); −π ≤ *z* ≤ π
sinh(x)	*y* = sinh(*x*); Winkel *x* im Bogenmaß
cosh(x)	*y* = cosh(*x*); Winkel *x* im Bogenmaß
tanh(x)	*y* = tanh(*x*); Winkel *x* im Bogenmaß

Hinweis

Zufallszahlen *y* können Sie mit der Funktion *y* = rand() in einem definierten Bereich YMIN ≤ *y* ≤YMAX mit dem Formelausdruck YMIN + rand() * (YMAX−YMIN) / 32767.0 erzeugen.

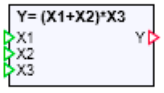
Parameter

Parametername	Beschreibung	Einheit	Standardwert
Formula	Formelausdruck, der den Ausgang *y* in Abhängigkeit von den Eingängen x_i berechnet.	–	X1

Beispiel für einen Baustein mit drei Eingängen X1 bis X3:

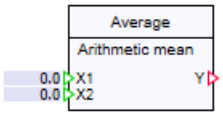
- (X1+X2)*X3

Der Formelausdruck wird im Symbol der Komponente angezeigt.



Average – Mittelwert

Symbol



Funktion

Der Komponententyp *Average* bildet den Mittelwert der n Eingänge X_1 bis X_n auf dem Ausgang *Y* ab.

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit "0" vorbelegt.

Arithmetic mean

Das arithmetische Mittel berechnet sich, indem alle Eingänge X_n summiert werden und durch die Anzahl *n* geteilt werden.

$$Y = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

Median

Alle Eingänge X_n werden (aufsteigend) nach ihrem Wert geordnet. Der Median entspricht genau dem Wert in der Mitte.

$$Y = \begin{cases} x_{\frac{n+1}{2}} & n \text{ ungerade} \\ \frac{1}{2}\left(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}\right) & n \text{ gerade} \end{cases}$$

Geometric mean

Das geometrische Mittel berechnet sich, indem alle Eingänge X_n multipliziert werden und von diesem Produkt die *n*-te Wurzel berechnet wird.

$$Y = \sqrt[n]{\prod_{i=1}^{n} X_i} = \sqrt[n]{X_1 \cdot X_2 \cdots X_n}$$

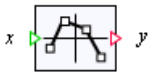
Falls das Produkt negativ wird, wird die Fehlermeldung *"Average – geometric mean: the product of all inputs is negative"* (Meldekategorie *ERROR*) erzeugt, und der Ausgang *Y* auf "0" gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
CalculationType	Art des Mittelwertes; *Arithmetic mean*, *Median* oder *Geometric mean*	–	*Arithmetic mean*

Characteristic – Kennlinie

Symbol



Funktion

Mit dem Komponententyp Characteristic kann eine Abbildung des Eingangswertes *x* auf den Ausgangswert *y* durch eine Kennlinie definiert werden:

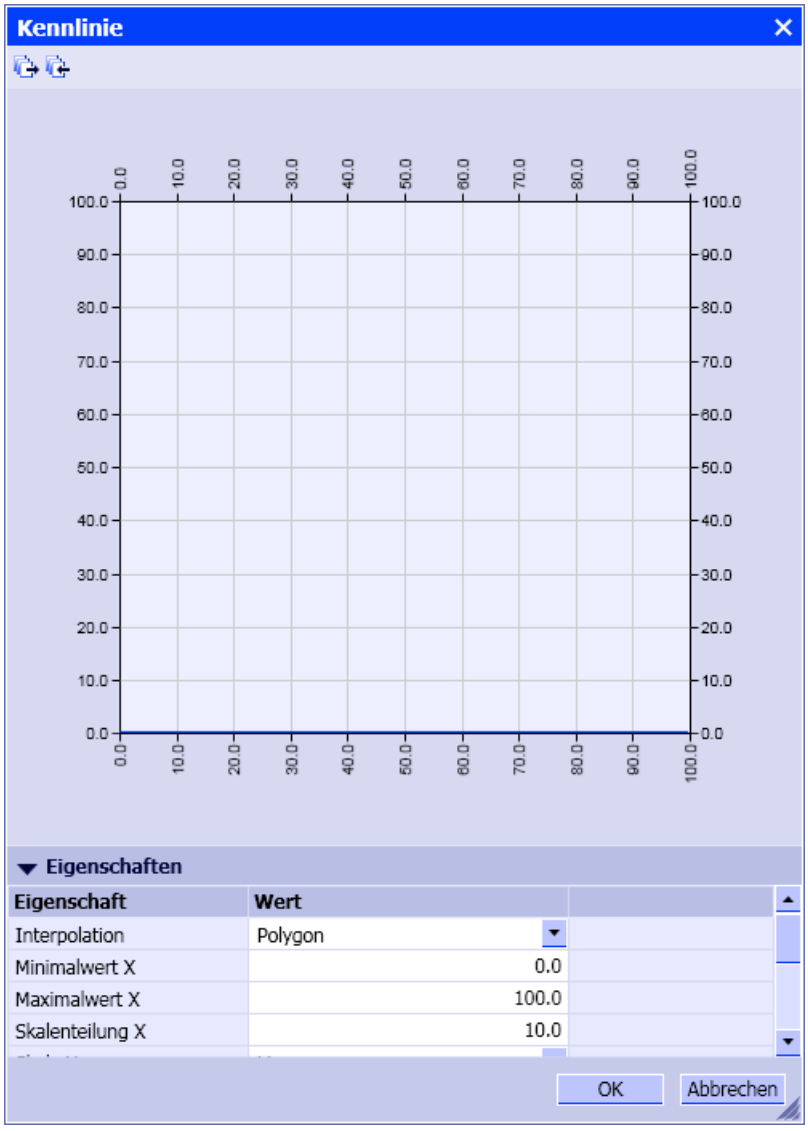
$y = f_K(x)$

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Characteristic	Definiert die zu verwendende Kennlinie. Öffnen Sie mit den Kennlinieneditor.	–	–

Kennlinie definieren

Eine Kennlinie kann mit diesem Editor durch die Vorgabe von *n* Stützpunkten (x_i, y_i), $i = 1, \ldots, n$ und durch die Wahl der *Interpolation* zwischen diesen Stützpunkten definiert werden.



Die Anzahl *n* der Stützpunkte ist beliebig und Sie können zwischen einer konstanten und einer linearen Interpolation wählen – die Kennlinie wird damit also als *Treppenkurve* oder als *Polygonzug* angesetzt. Voreingestellt ist der *Polygonzug*.

Außerhalb des Stützstellenbereichs (Interpolationsbereichs) werden die Ausgangswerte *y* extrapoliert.

Für jede der beiden Achsen, d. h. für die horizontale X-Achse und vertikale Y-Achse des Kennliniendiagramms, können jeweils folgende Größen in den Eigenschaften des Diagramms eingestellt werden:

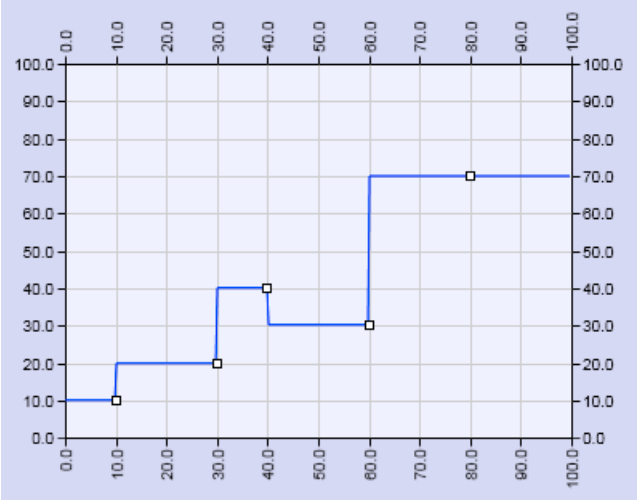
- der *Minimalwert* und der *Maximalwert*,
- die *Skalenteilung* und
- die *Skala*.

Voreingestellt ist sind beide Achsen gleich mit einer linearen Skala, einer Skalenteilung von zehn bei einem Minimalwert null und Maximalwert hundert.

Die *Skala* kann *linear* oder *logarithmisch* eingestellt werden.

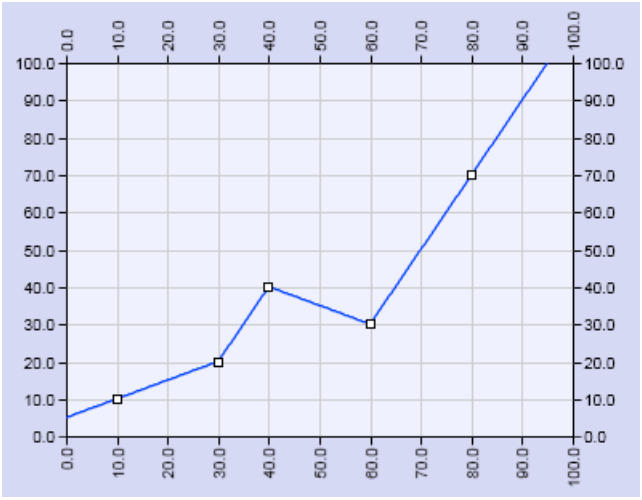
Die Kennlinienfunktion für n Stützpunkte ist somit gegeben als 0.0 für $n = 0$, y_1 für $n = 1$ und für $n > 1$ für die Treppenkurve durch

$$y = \begin{cases} y_1 \text{ für } x \le x_1, \\ y_i \text{ für } x_{i-1} < x \le x_i, \ i = 2,\ldots,n \\ y_n \text{ für } x > x_n \end{cases}$$



und für den Polygonzug durch

$$y = \begin{cases} y_1 + \dfrac{y_2 - y_1}{x_2 - x_1}(x - x_1) \text{ für } x \le x_1 \\ y_{i-1} + \dfrac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_{i-1}) \text{ für } x_{i-1} < x \le x_i, \ i = 2,\ldots,n \\ y_n + \dfrac{y_n - y_{n-1}}{x_n - x_{n-1}}(x - x_n) \text{ für } x > x_n \end{cases}$$



Stützpunkte einfügen:

Um einen Stützpunkt in das Diagramm einzufügen, führen Sie den Cursor an die gewünschte Position. Die aktuellen Koordinaten werden laufend an der Cursorposition eingeblendet. Durch einen Doppelklick fügen Sie dann einen Stützpunkt an der aktuellen Position ein.

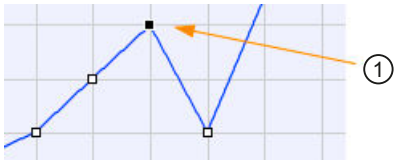




Alternativ können Sie einen Stützpunkt auch über das Kontextmenü an der aktuellen Position einfügen.

Stützpunkte löschen:

Zum Löschen eines Stützpunktes selektieren Sie den Stützpunkt. Der selektierte Stützpunkt ändert seine Darstellung.



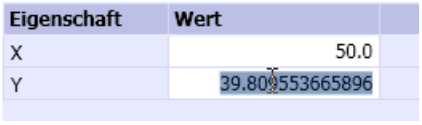
① Selektierter Stützpunkt

Den selektierten Stützpunkt können Sie mit <Entf> oder über das Kontextmenü löschen.

Koordinaten der Stützpunkte ändern

Selektieren Sie einen Stützpunkt mit einem Klick der linken Maustaste und verschieben Sie ihn mit gedrückter linker Maustaste auf die gewünschte Position. Sie können den Stützpunkt lediglich zwischen den X-Koordinaten seines linken und rechten Nachbarstützpunktes verschieben.

Alternativ können Sie die beiden Koordinaten des selektierten Stützpunktes in den Eigenschaften ändern. Öffnen Sie die Eingabe durch einen Mausklick in das Eingabefeld und geben Sie jeweils den gewünschten Wert ein. Eine Eingabe wird mit der *Return*-Taste abgeschlossen.



Stützpunkte importieren und exportieren

Sie können Stützpunkte auch aus einer Datei importieren. Mit dem Befehl aus der Icon-Leiste des Kennlinieneditors öffnet sich der Dialog zur Auswahl einer Excel-Datei im csv-Format. In dieser Stützpunktdatei stehen die Stützpunkte zeilenweise als Wertepaare mit ihren beiden Koordinaten: X-Koordinate Trennzeichen Y-Koordinate. Wenn als Dezimalzeichen der Punkt "." verwendet wird, ist das Komma "," das Trennzeichen. Wenn als Dezimalzeichen das Komma "," verwendet wird, ist der Strichpunkt ";" das Trennzeichen.

Mit dem Befehl aus der Icon-Leiste des Kennlinieneditors können Sie die aktuellen Stützpunkte auch in eine csv-Datei exportieren und dort beispielsweise weiter bearbeiten. Es wird eine Datei geschrieben, die alle Stützpunkte als Wertepaare (X-Koordinate und Y-Koordinate) enthält. Das Dezimalzeichen ist das Komma ",", das Trennzeichen ist der Strichpunkt ";".

Compare – Vergleichsfunktionen

Symbol



Funktion

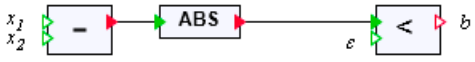
Der Komponententyp *Compare* vergleicht die analogen Eingänge x_1 und x_2. Der binäre Ausgang *b* wird auf 1 gesetzt, wenn der Vergleichsausdruck wahr ist – andernfalls wird *b* auf 0 gesetzt.

Der gewählte Vergleichsoperator wird im Symbol der Komponente dargestellt.

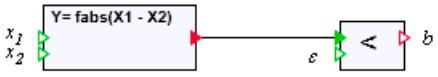
Um die Vergleichsoperatoren ">=" und "<=" vom rechten Rand des Symbols absetzen zu können, kann die Breite des Symbols verändert werden.



Da analoge Größen in SIMIT auf Variablen vom Typ double abgebildet werden, ist ein unmittelbarer Vergleich auf Gleichheit oder Ungleichheit nicht sinnvoll. Gleichheit von double-Größen ist nur innerhalb der durch den Rechner vorgegebenen Genauigkeit (Maschinengenauigkeit) gegeben. Gleichheit lässt sich beispielsweise mit dem in folgender Abbildung dargestellten Modell prüfen.



Alternativ kann ein funktional identisches Modell beispielsweise unter Verwendung des Formelbausteins *AFormula* angesetzt werden.



Von den beiden Größen x_1 und x_2 wird die Differenz gebildet. Der Betrag dieser Differenz wird dann mit einer vorgebbaren positiven Schranke ε verglichen:

$$b = \begin{cases} 1, \text{ falls } & |x_1 - x_2| < \varepsilon \\ 0, \text{ sonst} & \end{cases}$$

Für die Prüfung auf Ungleichheit ist in den beiden skizzierten Modellen nur in der Vergleichs-Komponente der Vergleich größer einzustellen.

Hinweis

Komponenten zur Prüfung auf Ungleichheit bzw. Gleichheit können Sie anhand der obigen Erläuterungen selbst erstellen.

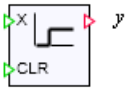
Sie können beispielsweise mit dem Makrokomponenteneditor entsprechende Makrokomponenten anlegen oder mit dem Component Type Editor (CTE) eine um diese Vergleiche erweiterte Vergleichskomponente erstellen.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Comparison	Legt die Art des Vergleichs fest: • Vergleich kleiner (<), also $b = \begin{cases} 1, \text{ falls } & x_1 < x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich kleiner gleich (<=), also $b = \begin{cases} 1, \text{ falls } & x_1 \leq x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich größer (>), also $b = \begin{cases} 1, \text{ falls } & x_1 > x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich größer gleich (>=), also $b = \begin{cases} 1, \text{ falls } & x_1 \geq x_2 \\ 0, \text{ sonst} \end{cases}$	–	<

DeadTime – Totzeitglied

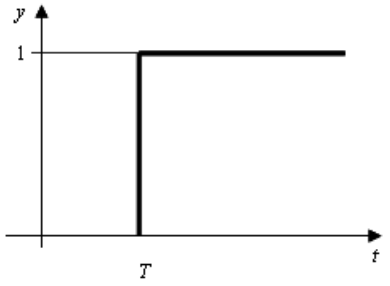
Symbol



Funktion

Mit dem Komponententyp *DeadTime* wird ein Totzeitglied zur Verfügung gestellt. Der Analogwert am Eingang *x* wird mit einer einstellbaren Verzögerung an den Ausgang *y* weitergegeben.

Eine sprungförmige Änderung des Eingangswertes *x* von null auf eins ergibt somit den Verlauf des Ausgangswerts *y* wie in folgender Abbildung dargestellt:



Hinweis

Wenn Sie die Zykluszeit Δ*t* in den Projekteigenschaften ändern, ändern sich entsprechend auch die eingestellten Totzeiten.

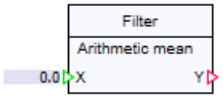
In der Komponente werden die Eingangswerte gespeichert und verzögert, entsprechend der eingestellten Totzeit, am Ausgang ausgegeben. Der Speicher wird entsprechend der parametrierten Anzahl an Verzögerungszyklen für *n* Werte angelegt. Alle Speicherstellen sind mit null vorbelegt. Über den binären Eingang *CLR* können der Ausgangswert *y* und alle Speicherstellen *n* auf null gesetzt werden.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Delay_Cycles	Verzögerungszeit *T* als ganzzahliges Vielfaches der Zykluszeit Δ*t* Maximalwert: 128 Mit dem Standardwert und einer Zykluszeit von z. B. 100 ms ergibt sich somit eine Verzögerung von einer Sekunde.	–	10

Filter

Symbol



Funktion

Der Komponententyp *Filter* bildet den zeitlichen Mittelwert des Einganges *X* auf den Ausgang *Y* ab.

Arithmetic mean

Das arithmetische Mittel berechnet sich, indem die zeitlichen Werte des Eingangs *X* summiert werden und durch die Anzahl der Zyklen *n* geteilt werden.

$$Y = \frac{1}{n}\sum_{i=0}^{n-1} X_{n-i} = \frac{X_n + X_{n-1} + \cdots + X_{n-(n-1)}}{n}$$

Median

Alle zeitlichen Werte des Eingangs *X* werden (aufsteigend) nach ihrem Wert geordnet. Der Median entspricht genau dem Wert in der Mitte.

$$Y = \begin{cases} x_{\frac{n+1}{2}} & n \text{ ungerade} \\ \frac{1}{2}\left(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}\right) & n \text{ gerade} \end{cases}$$

Geometric mean

Das geometrische Mittel berechnet sich, indem alle zeitlichen Werte des Eingangs *X* multipliziert werden und von diesem Produkt die *n*-te Wurzel berechnet wird.

$$Y = \sqrt[n]{\prod_{i=0}^{n-1} X_{n-i}} = \sqrt[n]{X_n \cdot X_{n-1} \cdots X_{nn-(n-)}}$$

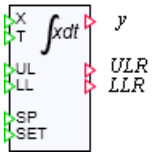
Falls das Produkt negativ wird, wird die Fehlermeldung *"Filter – geometric mean: the product Xi is negative"* (Meldekategorie *ERROR*) erzeugt, und der Ausgang *Y* auf "0" gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
CalculationType	Berechnungsmethode des Mittelwerts	–	*Arithmetic mean*
Number_Cycles	Anzahl der betrachteten Zyklen Wertebereich: 1 bis 128	–	3

INT – Integration

Symbol



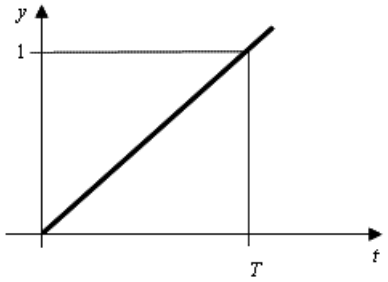
Funktion

Der Komponententyp *INT* bildet das Integral über das zeitabhängige analoge Eingangssignal *x* gemäß

$$y = \frac{1}{T}\int x \, dt$$

Der Integralwert wird dem analogen Ausgang *y* zugewiesen.

Bei einem sprungförmigen Eingangssignal der Amplitude 1 ergibt sich daher ein linear ansteigendes Ausgangssignal, wie in folgender Abbildung dargestellt.



Der Integralwert *y* ist auf ein Intervall beschränkt, das durch die beiden Grenzwerte *UL* (obere Grenze) und *LL* (untere Grenze) definiert ist:

$LL \le y \le UL$.

Die binären Ausgänge *ULR* und *LLR* zeigen an, dass der Integralwert die untere bzw. obere Grenze erreicht hat:

$$ULR = \begin{cases} 1, \text{ falls } & y \ge UL \\ 0, \text{ sonst} \end{cases}$$

und

$$LLR = \begin{cases} 1, \text{ falls } & y \le LL \\ 0, \text{ sonst} \end{cases}$$

Der untere Grenzwert muss kleiner als der obere Grenzwert sein. Wird diese Bedingung verletzt, dann wird die Fehlermeldung "*INT: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf 0 gesetzt.

Die Zeitkonstante *T* der Integration muss einen positiven Wert haben. Falls *T* nicht positiv ist, wird die Fehlermeldung "*INT: zero or negative time constant*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf null gesetzt.

Der untere Grenzwert ist mit 0, der obere Grenzwert mit 100 vorbelegt. Die Zeitkonstante *T* ist auf 1 s voreingestellt. Alle anderen Eingänge sind mit 0 vorbelegt.

Über den binären Eingang *SET* kann der Integrationswert *y* auf den Wert am Eingang *SP* gesetzt werden: falls *SET* auf eins gesetzt ist, wird der Integrationswert *y* gleich dem Wert am Eingang *SP* gesetzt. Auch in diesem Fall wird der Integrationswert durch *LL* und *UL* begrenzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Initial_Value	Initialwert des Integralwertes *y* beim Starten der Simulation	–	0.0

Bedienfenster

Im Bedienfenster wird der aktuelle Funktionswert als Prozentwert anzeigt.

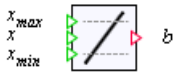
Folgende Werte können Sie ändern:

- Eingangswert *X*, auch während der Simulation
- Setpoint SP. Bestätigen Sie den Wert mit SET

Wenn die Eingänge *SP*, *SET* oder *X* per Signallinien mit anderen Komponenten verbunden sind, aktivieren Sie zur manuellen Bedienung zuerst den jeweiligen Signaltrenner.

Interval – Intervallabfrage

Symbol



Funktion

Der Komponententyp *Interval* prüft, ob ein Eingangswert *x* im abgeschlossenen Intervall [x_{min}, x_{max}] liegt. Wenn der Eingangswert im vorgegebenen Intervall liegt, dann wird der Binärausgang auf eins gesetzt, anderenfalls auf null:

$$b = \begin{cases} 1, \text{ für } & x_{min} \le x \le x_{max} \\ 0, \text{ sonst} & \end{cases}$$

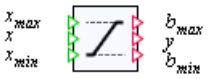
Parameter

Parametername	Beschreibung	Einheit	Standardwert
x_{min}	Untere Intervallgrenze	–	0
x_{max}	Obere Intervallgrenze	–	100

Die obere Intervallgrenze darf nicht kleiner als die untere Intervallgrenze sein. Falls beim Durchführen der Simulation die obere Intervallgrenze kleiner als die untere wird, wird die Fehlermeldung "*Interval: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *b* wird auf null gesetzt.

Limiter – Begrenzung

Symbol



Funktion

Der Komponententyp *Limiter* bildet einen auf den Bereich von x_{min} bis x_{max} begrenzten Eingangswert auf den Ausgang *y* ab:

$$y = \begin{cases} x_{max} \text{ für } x \geq x_{max} \\ x \quad \text{für } x_{min} < x < x_{max} \\ x_{min} \text{ für } x \leq x_{min} \end{cases}$$

Die binären Ausgänge b_{min} und b_{max} werden auf eins gesetzt, wenn die Begrenzung wirksam wird, also

$$b_{max} = \begin{cases} 1, \text{ falls } x \geq x_{max} \\ 0, \text{ sonst} \end{cases}$$

und

$$b_{min} = \begin{cases} 1, \text{ falls } x \leq x_{min} \\ 0, \text{ sonst} \end{cases}$$

Parameter

Parametername	Beschreibung	Einheit	Standardwert
x_{min}	Untere Begrenzung	–	0
x_{max}	Obere Begrenzung	–	100

Die obere Begrenzung darf nicht kleiner als die untere Begrenzung sein. Falls beim Durchführen der Simulation die untere Begrenzung gleich oder größer als die obere wird, wird die Fehlermeldung "*Limiter: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf null gesetzt.

MinMax – Minimal- und Maximalwertauswahl

Symbol



Funktion

Der Komponententyp *MinMax* bildet das Minimum oder Maximum der *n* Eingänge x_1 bis x_n auf den Ausgang *y* ab. Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit 0 vorbelegt.

Die für eine Komponente eingestellte Abbildung *MIN* oder *MAX* wird im Komponentensymbol wie in folgender Abbildung angezeigt:

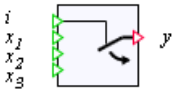


Parameter

Parametername	Beschreibung	Einheit	Standardwert
MinMax	Art des zu berechnenden Extremums	–	*MIN*

Multiplexer

Symbol



Funktion

Der Komponententyp *Multiplexer* schaltet einen der *n* verschiedenen Eingänge x_i in Abhängigkeit vom Wert am ganzzahligen (Integer) Selektionseingang *i* auf den Ausgang *y* durch:

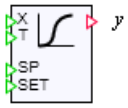
$y = x_i$ für $1 \leq i \leq n$.

Die Anzahl *n* der Eingänge x_i ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit "0" vorbelegt.

Der Wert am Selektionseingang *i* wird intern auf 1 bis *n* begrenzt, d. h. für Werte kleiner als 1 wird *i* auf "1" gesetzt, für Werte größer *n* wird *i* auf den Wert *n* gesetzt. Damit ist in der Voreinstellung der erste Eingang x_i auf den Ausgang *y* durchgeschaltet.

PTn – Verzögerung n-ter Ordnung

Symbol



Funktion

Mit dem Komponententyp *PTn* wird eine Verzögerung n-ter Ordnung zur Verfügung gestellt.

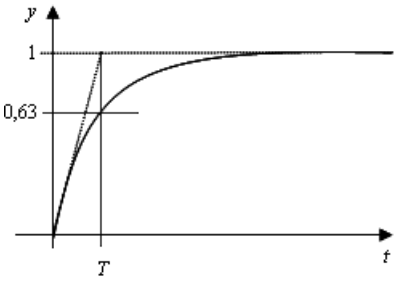
Bei einer Verzögerung erster Ordnung folgt der Funktionswert *y* am Ausgang dem Wert *x* am Eingang verzögert entsprechend der Differentialgleichung:

$$\frac{dy}{dt} = \frac{1}{T}(x - y)$$

Eine sprungförmige Änderung des Eingangswertes *x* von 0 auf 1 ergibt somit einen exponentiellen Verlauf des Ausgangswerts *y* (Sprungantwort) gemäß

$y = 1 - e^{-t/T}$

wie in folgender Abbildung dargestellt.



Bei Verzögerungen höherer Ordnung wird der Ausgangswert *y* aus der Verkettung von Verzögerungen erster Ordnung gebildet:

$$\frac{dz_i}{dt} = \frac{1}{T}(z_{i-1} - z_i),\ i = 1, \ldots, n$$

$y = z_n$

Dabei sind z_i, $i = 1,\ldots,n$, die *n* Zustände der Verzögerung *n*-ter Ordnung. Bildlich gesehen entspricht eine Verzögerung *n*-ter Ordnung der Hintereinanderschaltung von *n* Verzögerungen erster Ordnung.

Um zu vermeiden, dass die Komponente für zu kleine Verzögerungszeiten instabiles Verhalten annimmt, wird die Verzögerungszeitkonstante *T* auf Werte begrenzt, die größer oder gleich der Zykluszeit der Komponente sind. Sind die Werte am Eingang *T* kleiner als die Zykluszeit, dann wird die Fehlermeldung "*PTn: delay time below cycle time*" (Meldekategorie *ERROR*) erzeugt.

Die Verzögerungszeit ist mit 1 s vorbelegt. Alle anderen Eingänge der Komponente sind mit 0 vorbelegt.

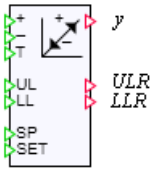
Über den binären Eingang *SET* können der Ausgangswert y und die Zustandswerte z_i, $i = 1,...,n$, auf den Wert am Eingang *SP* gesetzt werden: falls *SET* auf 1 gesetzt ist, werden diese Werte gleich dem Wert am Eingang *SP* gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
n	Ordnung der Verzögerung Maximalwert: 32	–	1
Initial_Value	Initialwert für die Zustände der Verzögerungsfunktion	–	0.0

Ramp – Rampenfunktion

Symbol



Funktion

Der Komponententyp *Ramp* inkrementiert oder dekrementiert seinen Funktionswert *y* in jedem Zeitschritt der Simulation um den Wert

$$\Delta y = (UL - LL)\frac{\Delta t}{T}$$

wobei Δt die Schrittweite der Simulation ist. Der Rampenwert *y* wird um Δy inkrementiert, wenn der Eingang "+" (*UP*) mit 1 belegt ist. Ist der Eingang "−" (*DOWN*) mit 1 belegt, wird der Rampenwert *y* um Δy dekrementiert. Sind beide Eingänge "+" und "−" mit 1 belegt, dann wird der Rampenwert *y* nicht verändert.

Der Rampenwert ist auf ein Intervall beschränkt, das durch die beiden Grenzwerte *UL* (obere Grenze) und *LL* (untere Grenze) definiert ist:

$LL \le y \le UL$.

Die binären Ausgänge *ULR* und *LLR* zeigen an, dass der Rampenwert die untere bzw. obere Grenze erreicht hat:

$$ULR = \begin{cases} 1, \text{ falls } & y \ge UL \\ 0, \text{ sonst} & \end{cases}$$

und

$$LLR = \begin{cases} 1, \text{ falls } & y \le LL \\ 0, \text{ sonst} \end{cases}$$

Der untere Grenzwert muss kleiner als der obere Grenzwert sein. Wird diese Bedingung verletzt, dann wird die Fehlermeldung "*RAMP: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf 0 gesetzt.

Die Zeitkonstante *T* muss einen positiven Wert haben. Falls *T* nicht positiv ist, wird die Fehlermeldung "*Ramp: zero or negativ time constant*" (Meldekategorie *ERROR*) erzeugt und der Ausgang *y* wird auf 0 gesetzt.

Die untere Grenze ist mit 0, die obere Grenze ist mit 100 vorbelegt. Die Zeitkonstante ist mit 10 Sekunden voreingestellt. Alle anderen Eingänge sind mit 0 vorbelegt.

Über den binären Eingang *SET* kann der Rampenwert *y* auf den Wert am Eingang *SP* gesetzt werden: falls *SET* auf 1 gesetzt ist, wird der Rampenwert *y* gleich dem Wert am Eingang *SP* gesetzt. Auch hier wird der Rampenwert durch *LL* und *UL* begrenzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Initial_Value	Initialwert der Rampenfunktion	–	0.0

Bedienfenster

Im Bedienfenster wird der aktuelle Funktionswert als Prozentwert anzeigt.

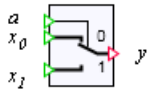
Folgende Werte können Sie ändern:

- Rampenfunktionswert über die Taster "Down" und "Up", auch während der Simulation
- Setpoint SP. Bestätigen Sie den Wert mit SET

Wenn die Eingänge *SP*, *SET*, "Down" oder "Up" per Signallinien mit anderen Komponenten verbunden sind, aktivieren Sie zur manuellen Bedienung zuerst den jeweiligen Signaltrenner.

Selection – Analogschalter

Symbol



Funktion

Der Komponententyp *Selection* schaltet einen der beiden Eingänge x_0 oder x_1 in Abhängigkeit von der Belegung des Binäreingangs *a* auf den Ausgang y durch.

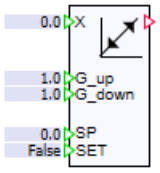
Ist der Auswahleingang $a = 0$, so wird der Eingang x_0 durchgeschaltet; ist der Auswahleingang $a = 1$, so wird der Eingang x_1 durchgeschaltet:

$$y = \begin{cases} x_0, \text{ falls } & a = 0 \\ x_1, \text{ falls } & a = 1 \end{cases}$$

Alle Eingänge sind mit 0 vorbelegt.

Tracking – Nachführung

Symbol



Funktion

Der Komponententyp *Tracking* führt den Ausgangswert *Y* dem Eingangswert *X* mit einem einstellbaren Gradienten nach. Sie können den Gradienten für die positive Steigung (*G_up*) und die negative Steigung (*G_down*) getrennt einstellen. Die Einheit der Gradienten ist 1/ Sekunde. Falls der Ausgang *Y* dem Eingang *X* entspricht, bleibt der Ausgangswert *Y* konstant.

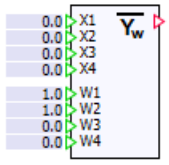
Über den binären Eingang *SET* können Sie den Ausgangswert *Y* auf den Wert am Eingang *SP* setzen. Falls *SET* auf True gesetzt ist, wird der Ausgangswert *Y* gleich dem Wert am Eingang *SP* gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Initial_Value	Initialwert der Trackingfunktion	–	0.0

WeightedAverage – Gewichtetes arithmetisches Mittel

Symbol



Funktion

Der Komponententyp *Weighted Average* bildet das gewichtete arithmetische Mittel aus den analogen Eingängen X_1, X_2, X_3 und X_4 mit den Gewichten W_1, W_2, W_3 und W_4 auf dem Ausgang Y ab.

$$Y = \frac{X_1 W_1 + X_2 W_2 + X_3 W_3 + X_4 W_4}{W_1 + W_2 + W_3 + W_4}$$

Die Summe von W_1, W_2, W_3 und W_4 darf nicht "0" sein. Falls beim Durchführen der Simulation die Summe gleich "0" wird, wird die Fehlermeldung "*Weighted_Average:sum of* W_1, W_2, W_3 und W_4 *is zero – division by zero*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang Y auf "0" gesetzt.

Die Eingänge X_1, X_2, X_3, X_4, W_3 und W_4 sind mit "0" vorbelegt. Die Eingänge W_1 und W_2 sind mit 1.0 vorbelegt.

8.1.3.3 Ganzzahlige Funktionen

Ganzzahlige Funktionen als Komponententypen

Alle Komponententypen mit ganzzahligen Funktionen sind in den Verzeichnissen *IntegerBasic* und *IntegerExtended* der Standardbibliothek enthalten.

Die vier grundlegenden ganzzahligen Funktionen Addition, Subtraktion, Multiplikation und Division, also die vier arithmetischen Grundoperationen, sind als Komponententypen im Verzeichnis *IntegerBasic* der Standardbibliothek abgelegt.

Die Symbole dieser Komponententypen sind zur Unterscheidung von den analogen Komponententypen in blau gehalten.

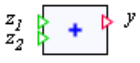
Im Verzeichnis *IntegerExtended* der Standardbibliothek sind weitere ganzzahlige Funktionen als Komponententypen abgelegt.

Die Symbole dieser Komponententypen sind zur Unterscheidung von den analogen Komponententypen in blau gehalten.

Ganzzahlige Basisfunktionen

ADD_I – Addition

Symbol



Funktion

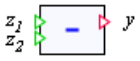
Der Komponententyp *ADD_I* bildet die Summe der ganzzahligen Werte an den *n* Eingängen x_1 bis x_n auf den Ausgang *y* ab:

$$y = \sum_{i=1}^{n} x_i = x_1 + x_2 + \ldots + x_n$$

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit null vorbelegt.

SUB_I – Subtraktion

Symbol



Funktion

Der Komponententyp *SUB_I* bildet die Differenz der ganzzahligen Werte an den beiden Eingängen x_1 und x_2 auf den Ausgang *y* ab gemäß:

$y = x_1 - x_2$

Alle Eingänge sind mit null vorbelegt.

MUL_I – Multiplikation

Symbol



Funktion

Der Komponententyp *MUL_I* bildet das Produkt der ganzzahligen Werte an den *n* Eingängen x_1 bis x_n auf den Ausgang *y* ab:

$$y = \prod_{i=1}^{n} x_i = x_1 x_2 \ldots x_n$$

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit eins vorbelegt.

DIV_I – Ganzzahlige Division

Symbol



Funktion

Der Komponententyp *DIV_I* bildet den Quotienten der ganzzahligen Werte an den Eingängen x_1 und x_2 gemäß

$x_1 = y \cdot x_2 + R$

als ganzzahlige Division mit Rest. Am Ausgang *y* steht das ganzzahlige Ergebnis der Division, am Ausgang *R* der Rest der ganzzahligen Division zur Verfügung.

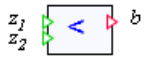
Der Dividend am Eingang x_1 ist mit null, der Divisoreingang x_2 ist mit eins vorbelegt.

Der Wert des Divisors x_2 darf nicht null werden. Falls beim Durchführen der Simulation der Divisor null wird, wird die Fehlermeldung "*DIV_I: division by zero*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* und Rest *R* werden auf null gesetzt.

Erweiterte ganzzahlige Funktionen

Compare_I – Vergleichsfunktionen

Symbol



Funktion

Der Komponententyp *Compare_I* vergleicht die ganzzahligen Eingänge x_1 und x_2. Der binäre Ausgang *b* wird auf 1 gesetzt, wenn der Vergleichsausdruck wahr ist – anderenfalls wird *b* auf 0 gesetzt.

Der gewählte Vergleichsoperator wird im Symbol der Komponente dargestellt. Um die Vergleichsoperatoren "<=", ">=" und "<>" vom rechten Rand des Symbols abzusetzen, verändern Sie die Breite des Symbols.

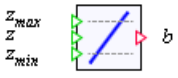


Parameter

Parametername	Beschreibung	Einheit	Standardwert
Comparison	Legt die Art des Vergleichs fest: • Vergleich "kleiner" (<), also $b = \begin{cases} 1, \text{ falls } & x_1 < x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich "kleiner gleich" (<=), also $b = \begin{cases} 1, \text{ falls } & x_1 \leq x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich "größer" (>), also $b = \begin{cases} 1, \text{ falls } & x_1 > x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich "größer gleich" (>=), also $b = \begin{cases} 1, \text{ falls } & x_1 \geq x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich "gleich" (=), also $b = \begin{cases} 1, \text{ falls } & x_1 = x_2 \\ 0, \text{ sonst} \end{cases}$ • Vergleich "ungleich" (<>), also $b = \begin{cases} 0, \text{ falls } & x_1 \neq x_2 \\ 1, \text{ sonst} \end{cases}$	–	<

Interval_I – Intervallabfrage

Symbol



Funktion

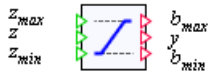
Der Komponententyp *Interval_I* prüft, ob ein ganzzahliger Eingangswert *x* im abgeschlossenen Intervall [x_{min}, x_{max}] liegt. Liegt der Eingangswert im vorgegebenen Intervall, dann wird der Binärausgang auf eins gesetzt, andernfalls auf null:

$$b = \begin{cases} 1, \text{ für } & x_{min} \le x \le x_{max} \\ 0, \text{ sonst} & \end{cases}$$

Die obere Intervallgrenze darf nicht kleiner als die untere Intervallgrenze sein. Falls beim Durchführen der Simulation die obere Intervallgrenze kleiner als die untere wird, wird die Fehlermeldung "*Interval_I: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *b* wird auf null gesetzt.

Limiter_I – Begrenzung

Symbol



Funktion

Der Komponententyp *Limiter_I* bildet einen auf den Bereich von x_{min} bis x_{max} begrenzten Eingangswert auf den Ausgang *y* ab:

$$y = \begin{cases} x_{max} \text{ für } & x \ge x_{max} \\ x \quad \text{für } & x_{min} < x < x_{max} \\ x_{min} \text{ für } & x \le x_{min} \end{cases}$$

Die binären Ausgänge b_{min} und b_{max} werden auf eins gesetzt, wenn die Begrenzung wirksam wird, also

$$b_{max} = \begin{cases} 1, \text{ falls } & x \ge x_{max} \\ 0, \text{ sonst} & \end{cases}$$

und

$$b_{min} = \begin{cases} 1, \text{ falls } & x \le x_{min} \\ 0, \text{ sonst} & \end{cases}$$

Die obere Begrenzung darf nicht kleiner als die untere Begrenzung sein. Falls beim Durchführen der Simulation die untere Begrenzung gleich oder größer als die obere wird, wird die Fehlermeldung "*Limiter_I: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf null gesetzt.

MinMax_I – Minimal- und Maximalwertauswahl

Symbol



Funktion

Der Komponententyp *MinMax_I* bildet das Minimum oder Maximum der *n* Eingänge z_1 bis z_n auf den Ausgang *y* ab. Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit 0 vorbelegt.

Die für eine Komponente eingestellte Abbildung *MIN* oder *MAX* wird im Komponentensymbol wie in folgender Abbildung angezeigt.

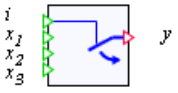


Parameter

Parametername	Beschreibung	Einheit	Standardwert
MinMax	Art des zu berechnenden Extremums	–	*MIN*

Multiplexer_I – Integermultiplexer

Symbol



Funktion

Der Komponententyp *Multiplexer_I* schaltet einen der *n* verschiedenen ganzzahligen (Integer) Eingänge x_i in Abhängigkeit vom Wert am Selektionseingang *i* auf den Ausgang *y* durch:

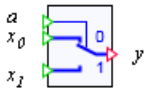
$y = x_i$ für $1 \le i \le n$.

Die Anzahl *n* der Eingänge x_i ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit 0 vorbelegt.

Der Wert am Selektionseingang *i* wird intern auf 1 bis *n* begrenzt, d. h. für Werte kleiner als 1 wird *i* auf 1 gesetzt, für Werte größer *n* wird *i* auf den Wert *n* gesetzt. Damit ist in der Voreinstellung der erste Eingang x_i auf den Ausgang *y* durchgeschaltet.

Selection_I – Integerschalter

Symbol



Funktion

Der Komponententyp *Selection_I* schaltet einen der beiden ganzzahligen (Integer) Eingänge x_0 oder x_1 in Abhängigkeit von der Belegung des Binäreingangs a auf den Ausgang y durch.

Ist der Auswahleingang $a = 0$, so wird der Eingang x_0 durchgeschaltet; ist der Auswahleingang $a = 1$, so wird der Eingang x_1 durchgeschaltet:

$$y = \begin{cases} x_0, \text{ falls } & a = 0 \\ x_1, \text{ falls } & a = 1 \end{cases}$$

Alle Eingänge sind mit 0 vorbelegt.

8.1.3.4 Mathematische Funktionen

Mathematische Funktionen als Komponententypen

Im Verzeichnis *Math* der Standardbibliothek sind die gebräuchlichsten mathematischen Funktionen in Form von Komponententypen enthalten: die Absolutwertbildung (*ABS*), das Radizieren (*SQRT*), der natürliche Logarithmus (*LN*) und die Exponentialfunktion (*EXP*) sowie die trigonometrischen Funktionen Sinus (*SIN*), Cosinus (*COS*) und Tangens (*TAN*).

Hinweis

Alternativ können Sie anstelle dieser Komponenten auch die entsprechend parametrierte Formelkomponente *AFormula* einsetzen.

Ihren Vorrat an mathematischen Funktionen dieser Art können Sie erweitern, indem Sie mit Hilfe des SIMIT-Erweiterungsmoduls Component Type Editor (CTE) entsprechende Komponententypen anlegen.

ABS – Absolutwert

Symbol



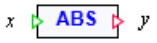
Funktion

Der Komponententyp *ABS* bildet den Absolutwert (Betrag) des Eintritts *x* auf den Ausgang *y* ab:

$y = |x|$.

ABS_I – Absolutwert ganzer Zahlen

Symbol



Das Symbol für *ABS_I* ist im Unterschied zum Komponententyp *ABS* in blau gehalten.

Funktion

Der Komponententyp *ABS_I* bildet den Absolutwert (Betrag) des ganzzahligen (Integer) Eintritts *x* auf den ganzzahligen Ausgang *y* ab:

$y = |x|$.

COS – Kosinusfunktion

Symbol



Funktion

Der Komponententyp *COS* bildet den Kosinuswert des Eintritts *x* auf den Ausgang *y* ab:

$y = \cos(x)$.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Unit	Maßeinheit des Arguments *x*: • rad (Bogenmaß) • deg (Gradmaß)	–	rad

EXP – Exponentialfunktion

Symbol



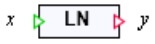
Funktion

Der Komponententyp *EXP* bildet den Exponentialwert des Eintritts *x* auf den Ausgang *y* ab:

$y = e^x$

LN – Natürlicher Logarithmus

Symbol



Funktion

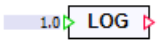
Der Komponententyp *LN* bildet den natürlichen Logarithmus des Eintritts *x* auf den Ausgang *y* ab:

$y = \ln(x)$.

Das Argument *x* muss positiv sein. Falls beim Durchführen der Simulation das Argument kleiner oder gleich null wird, wird die Fehlermeldung "*LN: invalid argument*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf null gesetzt. Das Argument *x* ist mit eins vorbelegt.

LOG – Dekadischer Logarithmus

Symbol



Funktion

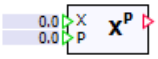
Der Komponententyp *LOG* bildet den dekadischen Logarithmus des Eingangs *X* auf den Ausgang *Y* ab.

$Y = Log_{10}X$

Das Argument *X* muss positiv sein. Falls beim Durchführen der Simulation das Argument kleiner oder gleich "0" wird, wird die Fehlermeldung "*LOG: invalid argument*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *Y* wird auf "0" gesetzt. Der Eingang *X* ist mit 1.0 vorbelegt.

POW – Potenz

Symbol



Funktion

Der Komponententyp *POW* bildet die Potenz der Basis (Eingang) *X* zum Exponent (Eingang) P auf den Ausgang *Y* ab.

$Y = X^P$

Alle Eingänge sind mit 0.0 vorbelegt.

RAND – Zufallszahlen

Symbol



Funktion

Der Komponententyp RAND gibt an seinem Ausgang Y zufällig erzeugte Zahlen aus, deren Wertebereiche und Verteilung über den Parameter *Distribution* eingestellt werden.

Hinweis

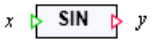
Beachten Sie, dass die Zufallszahlen berechnet werden und damit nicht unvorhersehbar sind.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Distribution	Wertebereich und Verteilung der Zufallszahlen • *Uniform distribution* Die Zufallszahlen liegen gleichverteilt im offenen Intervall (0,1). • *Normal distribution* Die Zufallszahlen sind unbegrenzt normalverteilt; der Mittelwert ist 0, die Standardabweichung 1.	–	*Uniform distribution*

SIN – Sinusfunktion

Symbol



Funktion

Der Komponententyp *SIN* bildet den Sinuswert des Eintritts *x* auf den Ausgang *y* ab:

$y = \sin(x)$.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Unit	Maßeinheit des Arguments *x*: • rad (Bogenmaß) • deg (Gradmaß)	–	rad

SQRT – Quadratwurzel

Symbol



Funktion

Der Komponententyp *SQRT* bildet die Quadratwurzel des Eintritts *x* auf den Ausgang *y* ab:

$$y = \sqrt{x}$$

Der Radikand *x* darf nicht negativ werden. Falls beim Durchführen der Simulation der Radikand negativ wird, wird die Fehlermeldung "*SQRT: invalid argument*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *y* wird auf null gesetzt.

TAN – Tangensfunktion

Symbol



Funktion

Der Komponententyp *TAN* bildet den Tangenswert des Eintritts *x* auf den Ausgang *y* ab:

$y = \tan(x)$.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Unit	Maßeinheit des Arguments *x*: • rad (Bogenmaß) • deg (Gradmaß)	–	rad

8.1.3.5 Binäre Funktionen

Binäre Funktionen als Komponententypen

Alle Funktionen zur Verarbeitung von binären Signalen sind in den Verzeichnissen *BinaryBasic* und *BinaryExtended* enthalten.

Komponententypen mit den drei grundlegenden binären Operationen Konjunktion (*AND*), Disjunktion (*OR*) und Negation (*NOT*) sowie Äquivalenz (*XNOR*) und Antivalenz (*XOR*) sind im Verzeichnis *BinaryBasic* abgelegt.

Im Verzeichnis *BinaryExtended* der Standardbibliothek sind weitere über die elementaren Binärfunktionen hinausgehende binäre (logische) Funktionen als Komponententypen abgelegt.

Grundlegende binäre Funktionen

AND – Konjunktion

Symbol



Funktion

Der Komponententyp *AND* bildet die *n* binären Werte an den Eingängen a_i als Konjunktion, also mit einer logischen (Boole'schen) "Und"-Funktion auf den Ausgang *b* ab:

$$b = \bigcap_{i=1}^{n} a_i$$

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit eins vorbelegt.

OR – Disjunktion

Symbol



Funktion

Der Komponententyp *OR* bildet die *n* binären Werte an den Eingängen a_i als Disjunktion, also mit einer logischen (Boole'schen) "Oder"-Funktion auf den Ausgang *b* ab:

$$b = \bigcup_{i=1}^{n} a_i$$

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit null vorbelegt.

NOT, NOTc – Negation

Die Negation wird in zwei Komponententypen *NOT* und *NOTc* zur Verfügung gestellt. Sie unterscheiden sich nur in der verwendeten Symbolik – ihre Funktion ist völlig identisch.

Symbol



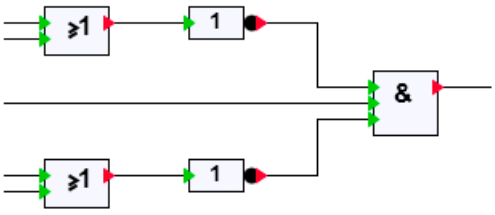
Funktion

Der Ausgang b ist gleich dem negierten Eingang a, also

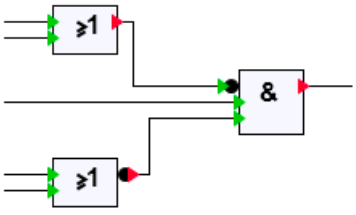
$$b = \bar{a}$$

Wie in folgender Abbildung zu sehen ist, kann bei Verwendung des Komponententyps *NOTc* die Negation übersichtlich und kompakt an den Ein- oder Ausgängen von Komponenten erfolgen.

Verwendung von "NOT"



Verwendung von "NOTc"



XOR – Antivalenz

Symbol



Funktion

Der Komponententyp *XOR* bildet die *n* binären Werte an den Eingängen a_i mit der "Exklusiv-Oder"-Funktion auf den Ausgang *b* ab: *b* ist eins, wenn eine ungerade Anzahl an Eingängen a_i eins ist. In allen anderen Fällen ist *b* null.

Für zwei Eingänge ist also

$b = a_1 \otimes a_2$

was der Antivalenz bzw. der "Ungleich"-Funktion entspricht.

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit null vorbelegt.

XNOR – Äquivalenz

Symbol



Funktion

Der Komponententyp *XNOR* bildet die *n* binären Werte an den Eingängen a_i mit der "Exklusiv-Nicht-Oder"-Funktion auf den Ausgang *b* ab: *b* ist eins, wenn eine gerade Anzahl an Eingängen a_i eins ist. In allen anderen Fällen ist *b* null.

Für zwei Eingänge ist also

$$b = \overline{a_1 \otimes a_2}$$

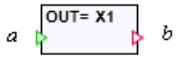
was der Äquivalenz bzw. der Identitäts-Funktion entspricht.

Die Anzahl der Eingänge *n* ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit null vorbelegt.

Erweiterte binäre Funktionen

BFormula – binäre Formelkomponente

Symbol



Funktion

Der Komponententyp *BFormula* ermöglicht die Verwendung von expliziten logischen Ausdrücken. Diese logische Funktion *f* berechnet einen Wert in Abhängigkeit von den *n* Werten an den Eingängen a_i. Der Funktionswert wird dem Ausgang *b* zugewiesen:

$b = f(a_1,...,a_n)$

Die Anzahl *n* der Eingänge kann frei von 1 bis 32 variiert werden. Im Formelausdruck können nur die Eingänge benutzt werden, die gemäß der aktuell eingestellten Anzahl zur Verfügung stehen.

Die in der folgenden Tabelle gelisteten Operatoren sind in Formelausdrücken verwendbar.

Tabelle 8-3 Zulässige Operatoren in Formelausdrücken der Komponente BFormula

Operator	Funktion
AND	Konjunktion (Und-Funktion)
OR	Disjunktion (Oder-Funktion)
NOT	Negation
(	Öffnende Klammer
)	Schließende Klammer

Hinweis

In der Formelkomponente wird nicht überprüft, ob im angegebenen Formelausdruck alle eingestellten Eingänge verwendet werden.

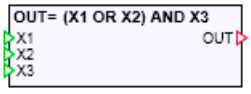
Parameter

Parametername	Beschreibung	Einheit	Standardwert
Formula	Formelausdruck, der den Ausgangswert *b* in Abhängigkeit von den Eingangswerten a_i berechnet	–	X1

Beispiel für einen Baustein mit drei Eingängen X1 bis X3:

- (X1 OR X2) AND X3

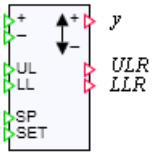
Der Formelausdruck wird im Symbol der Komponente angezeigt.



Um längere Formelausdrücke vollständig sichtbar zu machen, können Sie die Komponente verbreitern.

Counter – Auf- und Abwärtszähler

Symbol



Funktion

Der Komponententyp *Counter* bietet die Möglichkeit, die Wechsel von Binärsignalen zu zählen. Mit einem Wechsel des Binärwerts am Eingang "+" von null nach eins wird der Zählerwert am Ausgang *y* inkrementiert. Mit einem Wechsel des Binärwerts am Eingang "−" von null nach eins wird der Zählerwert am Ausgang *y* dekrementiert.

Der Zählerwert ist auf ein Intervall beschränkt, das durch die beiden Grenzwerte *UL* (obere Grenze) und *LL* (untere Grenze) definiert ist:

$LL \leq y \leq UL$.

Die binären Ausgänge *ULR* und *LLR* zeigen an, dass der Zählerwert die untere bzw. die obere Grenze erreicht hat:

$$ULR = \begin{cases} 1, \text{ falls } & y \geq UL \\ 0, \text{ sonst} \end{cases}$$

und

$$LLR = \begin{cases} 1, \text{ falls } & y \leq LL \\ 0, \text{ sonst} \end{cases}$$

Der untere Grenzwert muss kleiner als der obere Grenzwert sein. Wenn diese Bedingung verletzt wird, dann wird die Fehlermeldung "*Counter: limits do not match*" (Meldekategorie *ERROR*) erzeugt, und der Zählerwert am Ausgang *y* wird auf null gesetzt.

Über den binären Eingang *SET* kann der Zählerwert *y* auf den Wert am Eingang *SP* gesetzt werden: falls *SET* auf eins gesetzt ist, wird der Zählerwert *y* gleich dem Wert am Eingang *SP* gesetzt. Auch hier wird der Zählerwert durch *LL* und *UL* begrenzt.

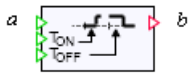
Die Grenzen sind mit null für den unteren Grenzwert und hundert für den oberen Grenzwert vorbelegt. Alle anderen Eingänge sind mit null vorbelegt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Decrement online änderbar	Wert, um den der Zählerwert bei jedem Wechsel des Binärwerts am Eingang "−" dekrementiert wird	–	1.0
Increment online änderbar	Wert, um den der Zählerwert bei jedem Wechsel des Binärwerts am Eingang "+" inkrementiert wird	–	1.0
Initial_Value	Zählerwert beim Initialisieren (Starten) der Simulation	–	1.0
TriggerUp	Art des Flankenwechsels für Inkrementieren des Zählerwerts	–	*Rising edge*
TriggerDown	Art des Flankenwechsels für Dekrementieren des Zählerwerts	–	*Rising edge*

Delay – Ein- und Ausschaltverzögerung

Symbol

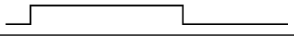
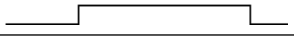


Funktion

Beim Komponententyp *Delay* wird das Binärsignal *b* am Ausgang dem Binärsignal *a* am Eingang verzögert nachgeführt.

Wenn am Eingang ein Signalwechsel von null nach eins stattfindet, wird der Ausgang nach Ablauf der Einschaltverzögerungszeit T_{ON} auf eins gesetzt. Wird das Eingangssignal vor Ablauf der Einschaltverzögerung wieder auf null gesetzt, dann bleibt das Ausgangssignal unverändert auf null. Erfolgt am Eingang ein Signalwechsel von eins nach null, dann wird der Ausgang nach Ablauf der Ausschaltverzögerungszeit T_{OFF} auf null gesetzt. Wird das Ausgangssignal vor Ablauf der Ausschaltverzögerung wieder auf eins gesetzt, dann bleibt das Ausgangssignal unverändert auf eins. Dieser Zusammenhang ist in folgender Tabelle veranschaulicht.

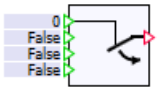
Tabelle 8-4 Signalverläufe am Ein- und Ausgang der Komponente Delay

Eingang a	
Ausgang b	

Die Verzögerungszeiten dürfen keine negativen Werte annehmen. Nimmt eine Verzögerungszeit negative Werte an, dann wird die Fehlermeldung "*Delay: on-delay time negativ value*" bzw. "*Delay: off-delay time negativ value*" (Meldekategorie *ERROR*) erzeugt, und die entsprechende Verzögerungszeit wird auf null gesetzt.

Multiplexer_B – Binärer Multiplexer

Symbol



Funktion

Der Komponententyp *Multiplexer_B* schaltet einen der n verschiedenen Eingänge X_i in Abhängigkeit vom Wert am ganzzahligen Selektionseingang *SEL* auf den Ausgang *Y* durch.

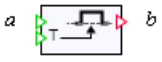
$Y = X_i$ *für* $1 \leq i \leq n$

Die Anzahl n der Eingänge X_i ist variabel und kann auf einen Wert zwischen 2 und 32 eingestellt werden. Alle Eingänge sind mit False vorbelegt.

Der Wert am Selektionseingang *SEL* wird intern auf 1 bis n begrenzt. Für Werte kleiner als 1 wird i auf 1 gesetzt, für Werte größer n wird *SEL* auf den Wert n gesetzt. Damit ist in der Voreinstellung der erste Eingang X_1 auf den Ausgang *Y* durchgeschaltet.

Pulse – Impuls

Symbol



Funktion

Der Komponententyp *Pulse* setzt den Ausgang *b*, wenn am Eingang *a* ein Flankenwechsel von 0 nach 1 stattfindet. Nach Ablauf der Zeit *T* wird der Ausgang *b* zurückgesetzt. Am Ausgang *b* wird somit ein Impuls mit der Pulsbreite *T* erzeugt. Die Pulsbreite ist über das Signal am Analogeingang *T* einstellbar.

Tabelle 8-5 Signalverläufe am Ein- und Ausgang der Komponente Pulse

	
Eingang a	
Ausgang b	

Die Pulsbreite *T* darf keine negativen Werte annehmen, anderenfalls wird die Fehlermeldung "*Puls: puls width negativ value*" (Meldekategorie *ERROR*) erzeugt, und der Ausgang *b* wird auf 0 gesetzt. Für Pulsbreiten, die kleiner als die Abtastzeit der Simulation sind, wird der Ausgangsimpuls für die Dauer eines Simulationszyklus gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Trigger online-änderbar	Art des ausgewerteten Flankenwechsels • *Rising edge*: ein Flankenwechsel von "False" nach "True" (positive Flanke) setzt den Ausgang *b* auf "True". • *Falling edge*: ein Flankenwechsel von "True" nach "False" (negative Flanke) setzt den Ausgang *b* auf "True".	–	*Rising edge*
Retriggerable online-änderbar	Art der Reaktion auf den Wechsel im Eingangssignal von "0" nach "1" • "True": der Ausgangsimpuls wird bei jedem Wechsel im Eingangssignal von "0" nach "1" erneut gestartet. • "False": der Ausgangsimpuls wird beim Wechsel im Eingangssignal von "0" nach "1" nicht erneut gestartet.	–	*False*

RS_FF – Flipflop mit Vorzugslage "Rücksetzen"

Symbol



Funktion

Der Komponententyp *RS_FF* stellt die einfachste Art eines Flipflops zur Verfügung: ein RS-Flipflop. Ist der Wert am Setzeingang *S* gleich eins, dann wird der Ausgangswert *Q* auf eins gesetzt. Ist der Eingangswert am Rücksetzeingang *R1* auf eins gesetzt, dann wird der Ausgang auf null zurückgesetzt. Der Rücksetzeingang ist dominierend, d. h. wenn beide Eingänge auf eins gesetzt sind, wird der Ausgang *Q* auf null zurückgesetzt. Der Ausgang $\overline{Q}$ nimmt immer den zum Ausgang *Q* inversen Wert an.

In der folgenden Tabelle sind die möglichen Zustände der Komponente *RS_FF* gelistet.

Tabelle 8-6 Zustandstabelle der Komponente RS_FF

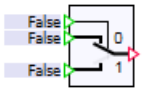
Eingang S	Eingang R1	Ausgang Q	Ausgang $\overline{Q}$
0	0	unverändert	unverändert
0	1	0	1
1	0	1	0
1	1	0	1

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Initial_State	Initialwert des Ausgangs *Q*	–	0

Selection_B – Binärschalter

Symbol



Funktion

Der Komponententyp *Selection_B* schaltet abhängig von der Belegung des Binäreingangs *SEL* einen der beiden Binäreingänge X_0 oder X_1 auf den Ausgang *Y* durch.

Ist der Auswahleingang *SEL* = False, so wird der Eingang X_0 durchgeschaltet; ist der Auswahleingang *SEL* = True, so wird der Eingang X_1 durchgeschaltet.

$$Y = \begin{cases} X_0, \text{ falls } SEL = false \\ X_1, \text{ falls } SEL = true \end{cases}$$

Alle Eingänge sind mit False vorbelegt.

SR_FF – Flipflop mit Vorzugslage "Setzen"

Symbol



Funktion

Der Komponententyp *SR_FF* bildet ein SR-Flipflop nach. Ist der Wert am Setzeingang *S1* gleich eins, dann wird der Ausgangswert *Q* auf eins gesetzt. Ist der Eingangswert am Rücksetzeingang *R* auf eins gesetzt, dann wird der Ausgang auf null zurückgesetzt. Der Setzeingang ist dominierend, d. h. wenn beide Eingänge auf eins gesetzt sind, wird der Ausgang *Q* auf eins gesetzt. Der Ausgang $\overline{Q}$ besitzt immer den zum Ausgang *Q* inversen Wert.

In der folgenden Tabelle sind die möglichen Zustände der Komponente *SR_FF* gelistet.

Tabelle 8-7 Zustandstabelle der Komponente SR_FF

Eingang S	Eingang R1	Ausgang Q	Ausgang $\overline{Q}$
0	0	unverändert	unverändert
0	1	0	1
1	0	1	0
1	1	1	0

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Initial_State	Initialwert des Ausgangs *Q*	–	0

8.1.3.6 Konvertierung von Werten

Konvertierung von Signalen als Komponententypen

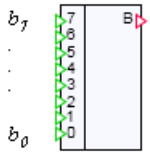
Im Verzeichnis *Conv* der Standardbibliothek sind Komponententypen zur Konvertierung von Signalen eingeordnet:

- *Bit2Byte* zur Konvertierung von Bits in einen Bytewert,
- *Byte2Bit* zur Konvertierung von Bytes in Bit,
- *Byte2Word* zur Konvertierung von Bytes in ein Wort,
- *Word2Byte* zur Konvertierung eines Worts in Bytes,
- *Byte2DWord* zur Konvertierung von Bytes in ein Doppelwort,
- *DWord2Byte* zur Konvertierung eines Doppelworts in Bytes,
- *Analog2Integer* zur Konvertierung eines Analogwerts in einen ganzzahligen Wert,
- *Integer2Analog* zur Konvertierung eines ganzzahligen Wertes in einen Analogwert,
- *Raw2Phys* zur Konvertierung eines Rohwertes in einen Analogwert,
- *Phys2Raw* zur Konvertierung eines Analogwerts in einen Rohwert,
- *Unsigned2Signed* zur Konvertierung von vorzeichenlosen Werten in vorzeichenbehaftete Werte,
- *Signed2Unsigned* zur Konvertierung von vorzeichenbehaften Werten in vorzeichenlose Werte,
- *Real2Byte* zur Wandlung einer Gleitpunktzahl in ihre binäre Repräsentation und
- *Byte2Real* zur Wandlung der binären Repräsentation einer Gleitpunktzahl in einen analogen Wert.

Anschlüsse an Komponenten für Bytes, Worte und Doppelworte sind ganzzahlige (Integer) Anschlüsse. Auch Rohwerte sind ganzzahlige (Integer) Werte.

Bit2Byte – Konvertierung von Bit in Byte

Symbol



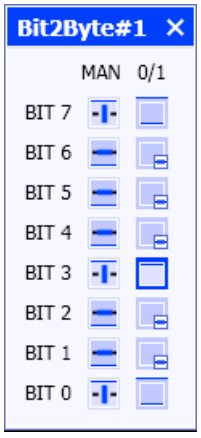
Funktion

Der Komponententyp *Bit2Byte* konvertiert die binären Eingangswerte b_i , $i = 0, ..., 7$, in einen Bytewert *B* am ganzzahligen Ausgang gemäß

$$B = \sum_{i=0}^{7} b_i \cdot 2^i$$

Bei der Umwandlung stellt b_0 somit das niedrigstwertige Bit, b_7 das höchstwertige Bit dar.

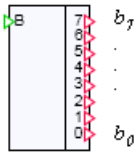
Im Bedienfenster der Komponente können die einzelnen Bits b_i bei laufender Simulation individuell gesetzt werden. Zum Setzen eines Bits öffnen Sie das Bedienfenster der Komponente. Wählen Sie jetzt die Bits aus, die Sie manuell setzen möchten, indem Sie die jeweils links angeordnete Schaltfläche betätigen. Anschließend können Sie das jeweilige Bit durch Betätigen des rechten Schalters setzen und rücksetzen, d. h. zwischen null und eins umschalten. In der folgenden Abbildung sind die Bits b_0, b_3 und b_7 zum Setzen ausgewählt, Bit b_3 ist gesetzt.



Nicht gesetzte Bits sind durch einen hellblauen Rahmen gekennzeichnet, gesetzte Bits durch einen dunkelblauen Rahmen.

Byte2Bit – Konvertierung von Byte in Bit

Symbol

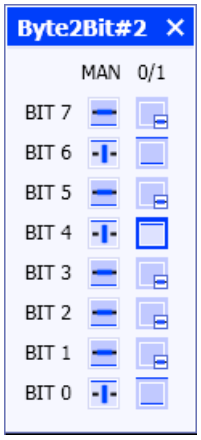


Funktion

Der Komponententyp *Byte2Bit* konvertiert einen Bytewert am ganzzahligen Eingang *B* in binäre Werte b_i , $i = 0, ... ,7$, an den Ausgängen. Bei der Umwandlung wird b_0 das niedrigstwertige Bit, b_7 das höchstwertige Bit des Bytewerts zugewiesen.

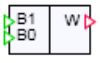
Der Eingang wird auf den Wertebereich eines Bytes, also auf den Bereich null bis 255 begrenzt. Falls beim Durchführen der Simulation der Eingangswert nicht in diesem Bereich liegt, wird die Meldung "*Byte2Bit: input not a valid byte value*" (Meldekategorie *ERROR*) erzeugt.

Im Bedienfenster der Komponente können die einzelnen Ausgangsbits b_i bei laufender Simulation individuell gesetzt werden. Zum Setzen eines Bits öffnen Sie das Bedienfenster der Komponente. Wählen Sie jetzt die Bits aus, die Sie manuell setzen möchten, indem Sie den jeweils links angeordneten Schalter betätigen. Anschließend können Sie das jeweilige Bit durch Betätigen des rechten Schalters setzen und rücksetzen, d. h. zwischen null und eins umschalten. In der folgenden Abbildung sind die Bits b_0, b_4 und b_6 zum Setzen ausgewählt, Bit b_4 ist gesetzt:



Byte2Word – Konvertierung von Byte in Wort

Symbol



Funktion

Der Komponententyp *Byte2Word* fasst zwei Bytewerte B_1, B_0 an den ganzzahligen Eingängen zu einem Wort am ganzzahligen Ausgang *W* zusammen. B_1 ist das höchstwertige Byte, B_0 das niedrigstwertige Byte.

Die Werte am Eingang werden auf den Wertebereich eines Bytes, also auf den Bereich null bis 255 begrenzt. Falls beim Durchführen der Simulation ein Eingangswert nicht in diesem Bereich liegt, wird die Meldung "*Byte2Word: input not a valid byte value*" (Meldekategorie *ERROR*) erzeugt.

Word2Byte – Konvertierung von Wort in Byte

Symbol



Funktion

Der Komponententyp *Word2Byte* zerlegt ein Wort am ganzzahligen Eingang *W* in zwei Bytewerte B_1, B_0 an den ganzzahligen Ausgängen. B_1 ist das höchstwertige Byte, B_0 das niedrigstwertige Byte.

Der Wert am Eingang wird entsprechend dem Wertebereich eines Worts auf den Bereich null bis $2^{16}-1$ begrenzt. Falls beim Durchführen der Simulation der Eingangswert nicht in diesem Bereich liegt, wird die Meldung "*Word2Byte: input not a valid word value*" (Meldekategorie *ERROR*) erzeugt.

Byte2DWord – Konvertierung von Byte in Doppelwort

Symbol



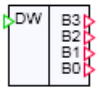
Funktion

Der Komponententyp *Byte2DWord* fasst vier Bytewerte B_i, $i = 0, \ldots, 3$, an den ganzzahligen Eingängen zu einem Doppelwort in der Reihenfolge $B_3 - B_2 - B_1 - B_0$ am ganzzahligen Ausgang *DW* zusammen. B_3 ist somit das höchstwertige Byte, B_0 das niedrigstwertige Byte.

Die Werte am Eingang werden auf den Wertebereich eines Bytes, also auf den Bereich null bis 255 begrenzt. Falls beim Durchführen der Simulation ein Eingangswert nicht in diesem Bereich liegt, wird die Meldung "*Byte2DWord: input not a valid byte value*" (Meldekategorie *ERROR*) erzeugt.

DWord2Byte – Konvertierung von Doppelwort in Byte

Symbol



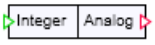
Funktion

Der Komponententyp *DWord2Byte* zerlegt ein Doppelwort am ganzzahligen Eingang *DW* in vier Bytewerte B_i, $i = 0, \ldots, 3$, an den ganzzahligen Ausgängen. B_3 ist das höchstwertige Byte, B_0 das niedrigstwertige Byte.

Der Wert am Eingang wird entsprechend dem Wertebereich eines Doppelworts auf den Bereich null bis $2^{32}-1$ begrenzt. Falls beim Durchführen der Simulation der Eingangswert nicht in diesem Bereich liegt, wird die Meldung "*DWord2Byte: input not a valid double word value*" (Meldekategorie *ERROR*) erzeugt.

Integer2Analog – Konvertierung von Integer in Analog

Symbol

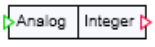


Funktion

Der Komponententyp *Integer2Analog* konvertiert einen ganzzahligen (Integer) Wert am Eingang in Analogwert am Ausgang.

Analog2Integer – Konvertierung von Analog in Integer

Symbol

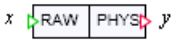


Funktion

Der Komponententyp *Analog2Integer* einen Analogwert am Eingang in einen ganzzahligen (Integer) Wert am Ausgang. Der ganzzahlige Wert ist ein gerundeter Wert.

Raw2Phys – Konvertierung von Raw in Physical

Symbol



Funktion

Der Komponententyp *Raw2Phys* überführt einen ganzzahligen Wert *x* am Eingang in einen analogen Ausgangswert *y* gemäß der einfachen linearen Transformation:

$$\frac{y - y_L}{x - x_L} = \frac{y_U - y_L}{x_U - x_L}$$

Der Eingangswert *x* kann beispielsweise ein sogenannter Rohwert eines Automatisierungssystems sein. Der Ausgangswert *y* ist dann der auf einen definierten physikalischen Wertebereich abgebildete Wert.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Phys_Lower_Limit	Untere Grenze für das Transformationsintervall des ganzzahligen Werts	–	0.0
Phys_Upper_Limit	Obere Grenze für das Transformationsintervall des ganzzahligen Werts	–	100.0
Raw_Lower_Limit	Untere Grenze für das Transformationsintervall des analogen Werts	–	−27648
Raw_Upper_Limit	Obere Grenze für das Transformationsintervall des analogen Werts	–	27648

Die Parameter können Sie während der laufenden Simulation ändern.

Phys2Raw – Konvertierung von Physical in Raw

Symbol



Funktion

Der Komponententyp *Phys2Raw* überführt einen Analogwert *x* am Eingang in einen ganzzahligen Ausgangswert *y* gemäß der einfachen linearen Transformation:

$$\frac{x - x_L}{y - y_L} = \frac{x_U - x_L}{y_U - y_L}$$

Der Eingangswert *x* wird damit beispielsweise als Messwert einer physikalischen Größe in den sogenannten Rohwert *y* eines Automatisierungssystems transformiert.

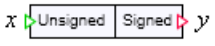
Parameter

Parametername	Beschreibung	Einheit	Standardwert
Phys_Lower_Limit	Untere Grenze für das Transformationsintervall des analogen Werts	–	0.0
Phys_Upper_Limit	Obere Grenze für das Transformationsintervall des analogen Werts	–	100.0
Raw_Lower_Limit	Untere Grenze für das Transformationsintervall des ganzzahligen Werts	–	−27648
Raw_Upper_Limit	Obere Grenze für das Transformationsintervall des ganzzahligen Werts	–	27648

Die Parameter können Sie während der laufenden Simulation ändern.

Unsigned2Signed – Konvertierung von vorzeichenlos in vorzeichenbehaftet

Symbol



Funktion

Der Komponententyp *Unsigned2Signed* wandelt einen vorzeichenlosen ganzzahligen Wert *x* am Eingang in einen vorzeichenbehafteten Wert *y* am Ausgang.

Mit dem Parameter *Width* wird festgelegt, ob der Eingangswert als Wert mit einer Datenbreite von *1 Byte*, *2 Byte* oder *4 Byte* angenommen wird. Der Ausgangswert wird entsprechend der eingestellten Datenbreite begrenzt. Die Konvertierung wird wie folgt vorgenommen:

Datenbreite: 1 Byte

$$y = \begin{cases} -1, & \text{falls } x > 2^8 - 1 \\ 0, & \text{falls } x < 0 \\ x - 2^8, & \text{falls } x > 2^7 - 1 \\ x, & \text{sonst} \end{cases}$$

Datenbreite: 2 Byte

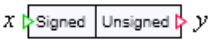
$$y = \begin{cases} -1, & \text{falls } x > 2^{16} - 1 \\ 0, & \text{falls } x < 0 \\ x - 2^{16}, & \text{falls } x > 2^{15} - 1 \\ x, & \text{sonst} \end{cases}$$

Datenbreite: 4 Byte

$$y = \begin{cases} -1, & \text{falls } x > 2^{32} - 1 \\ 0, & \text{falls } x < 0 \\ x - 2^{32}, & \text{falls } x > 2^{31} - 1 \\ x, & \text{sonst} \end{cases}$$

Signed2Unsigned – Konvertierung von vorzeichenbehaftet in vorzeichenlos

Symbol



Funktion

Der Komponententyp *Signed2Unsigned* wandelt einen vorzeichenbehafteten ganzzahligen Wert *x* am Eingang in einen vorzeichenlosen Wert *y* am Ausgang.

Mit dem Parameter *Width* wird festgelegt, ob der Ausgangswert als Wert mit einer Datenbreite von *1 Byte*, *2 Byte* oder *4 Byte* angenommen wird. Der Ausgangswert wird entsprechend der eingestellten Datenbreite begrenzt. Die Konvertierung wird wie folgt vorgenommen:

Datenbreite: 1 Byte

$$y = \begin{cases} -2^{7}, & \text{falls } x < -2^{7} \\ 2^{7} - 1, & \text{falls } x > 2^{7} - 1 \\ x + 2^{8}, & \text{falls } x < 0 \\ x, & \text{sonst} \end{cases}$$

Datenbreite: 2 Byte

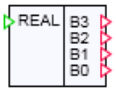
$$y = \begin{cases} -2^{15}, & \text{falls } x < -2^{15} \\ 2^{15} - 1, & \text{falls } x > 2^{15} - 1 \\ x + 2^{16}, & \text{falls } x < 0 \\ x, & \text{sonst} \end{cases}$$

Datenbreite: 4 Byte

$$y = \begin{cases} -2^{31}, & \text{falls } x < -2^{31} \\ 2^{31} - 1, & \text{falls } x > 2^{31} - 1 \\ x + 2^{32}, & \text{falls } x < 0 \\ x, & \text{sonst} \end{cases}$$

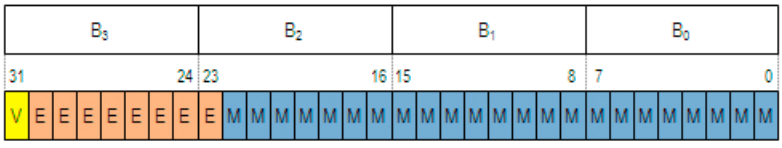
Real2Byte – Konvertierung von Real in Byte

Symbol



Funktion

Der Komponententyp *Real2Byte* wandelt den Wert eines analogen Signals am Eingang *REAL* in die binäre Repräsentation einer einfach genauen Gleitpunktzahl (Typ "single") nach IEEE 754. Die gewandelte Gleitpunktzahl wird auf die vier Bytewerte B_i, $i = 0, \ldots, 3$, an den ganzzahligen Ausgängen abgebildet, wie in der folgenden Abbildung verdeutlicht.



V	Vorzeichen
E	Exponent
M	Mantisse

Hinweis

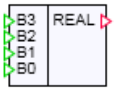
Durch die Abbildung des analogen Signals (Typ double) auf das einfach genaue Zahlenformat (Typ single) werden die Genauigkeit und der Wertebereich eingeschränkt.

Bei der hier vorliegenden Wandlung handelt es sich nicht um die Konvertierung einer Gleitpunktzahl in eine ganze Zahl (Integer). Für diese Konvertierung steht Ihnen der Komponententyp *Analog2Integer* zur Verfügung.

Wenn Sie analoge Signale an eine SIMATIC-Steuerung übertragen möchten, stellen Sie üblicherweise den Datentyp des entsprechenden Signals in der Kopplung auf *REAL*. Die Wandlung des analogen Signals erfolgt dann automatisch. Eine Komponente vom Typ *Real2Byte* benötigen Sie aber beispielsweise für den Fall, dass Sie analoge Signale direkt in Merker oder Datenbausteine einer SIMATIC-Steuerung übertragen möchten.

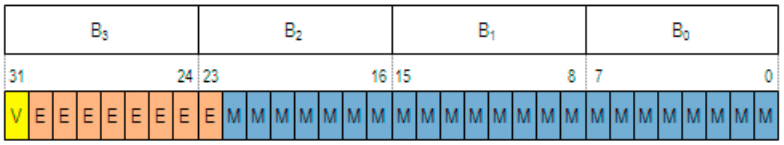
Byte2Real – Konvertierung von Byte in Real

Symbol



Funktion

Der Komponententyp *Byte2Real* wandelt die binäre Repräsentation einer Gleitpunktzahl vom Typ "single" nach IEEE 754 an den ganzzahligen Eingängen B_i, i = 0, ... ,3, in den Wert eines analogen Signals am Ausgang *REAL*, wie in der folgenden Abbildung verdeutlicht.



V	Vorzeichen
E	Exponent
M	Mantisse

Hinweis

Bei der hier vorliegenden Wandlung handelt es sich nicht um die Konvertierung einer ganzen Zahl (Integer) in eine Gleitpunktzahl. Für diese Konvertierung steht Ihnen der Komponententyp *Analog2Integer* zur Verfügung.

Wenn Sie Gleitpunktzahlen von einer SIMATIC-Steuerung erhalten möchten, stellen Sie üblicherweise den Datentyp des entsprechenden Signals in der Kopplung auf *REAL*. Die Wandlung in ein Analogsignal erfolgt dann automatisch. Eine Komponente vom Typ *Byte2Real* benötigen Sie aber beispielsweise für den Fall, dass Sie Gleitpunktzahlen direkt aus Merkern oder Datenbausteinen einer SIMATIC-Steuerung lesen möchten.

8.1.3.7 Allgemeine Komponenten im Verzeichnis Misc

Verbinder

Drei sogenannte Verbinderkomponententypen – kurz Verbinder genannt – werden zur Verfügung gestellt:

- Der analoge Verbinderkomponententyp *AConnector*,
- der binäre Verbinderkomponententyp *BConnector* und
- der ganzzahlige Verbinderkomponententyp *IConnector*.

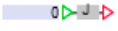
Verbinderkomponenten



Analoger Verbinder



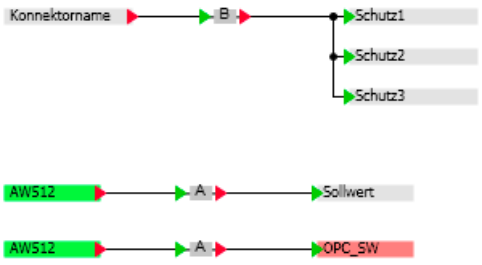
Binärer Verbinder



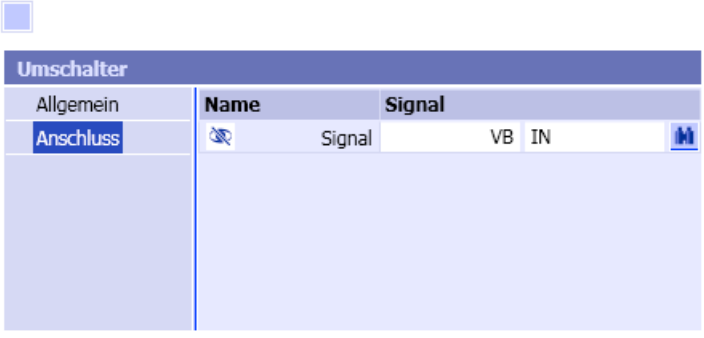
Ganzzahliger Verbinder

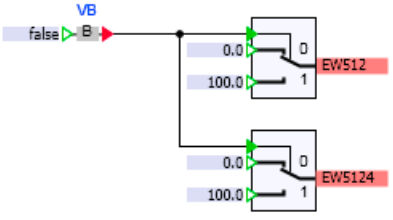
Eine Verbinderkomponente reicht den Eingangswert unverändert und ohne Verzögerung an ihren Ausgang weiter. Sie wird beispielsweise in den im Folgenden beschriebenen Fällen benötigt:

- Konnektoren (Peripheriekonnektoren oder globale Konnektoren) können nicht direkt miteinander verbunden werden. Wie in der folgenden Abbildung beispielhaft dargestellt, können Konnektoren aber mit Hilfe von Verbinderkomponenten verbunden werden.

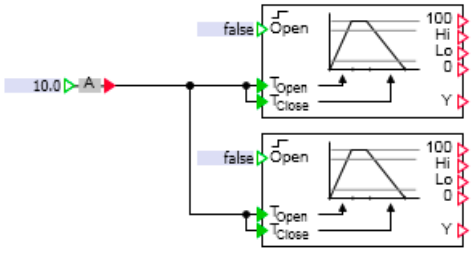


- Eingabeelemente wie beispielsweise ein Umschalter werden direkt mit dem zu setzenden Signal verknüpft. Mit Hilfe eines Verbinders kann ein Eingabeelement direkt auf mehrere Komponenten wirken. Wie in der folgenden Abbildung dargestellt, wird dazu das Eingangssignal *IN* des Verbinders *VB* mit dem Eingabeelement verknüpft.





- Komponenteneingänge können direkt mit einem Wert vorbelegt werden. Mit Hilfe eines Verbinders können mehrere Eingänge mit einem einzigen Wert vorbelegt werden. Wie in der folgenden Abbildung beispielhaft dargestellt, wird dazu ein Verbinder mit den zu setzenden Eingängen der Komponenten verbunden. Der zu setzende Wert wird dann am Eingang des Verbinders gesetzt.



Hilfskomponenten für die Makroerstellung

Drei sogenannte Hilfskomponententypen für Makros werden zur Verfügung gestellt:

- der analoge Konstantenkomponententyp *AConst*,
- der binäre Konstantenkomponententyp *BConst* und
- der ganzzahlige Konstantenkomponententyp *IConst*.

Konstantenkomponententypen für Makros

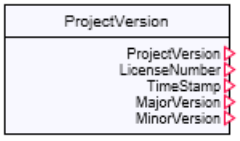
	Analoge Konstante
	Binäre Konstante
	Ganzzahlige Konstante

Parameter für eine Makrokomponente können nur aus den Parametern der in der Makrokomponente verwendeten Komponenten gebildet werden. Die Konstantenkomponententypen stellen so einen Parameter zur Verfügung, um auch Eingänge einer Komponente als Makroparameter zu setzen.

Weitere Informationen hierzu finden Sie im Abschnitt: Parameter von Makrokomponenten definieren (Seite 248)

ProjectVersion – Projektversion

Symbol



Funktion

An den Ausgängen einer Komponente dieses Typs stehen folgende ganzzahligen Werte zur Verfügung:

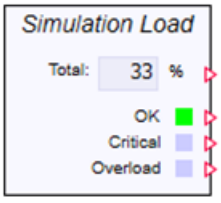
- **ProjectVersion**
  Die vollständige Versionsangabe als Zahl kodiert.
- **LicenseNumber**
  Die Lizenznummer als Zahl kodiert.
- **TimeStamp**
  Der Zeitstempel
- **MajorVersion**
  Die Hauptversion
- **MinorVersion**
  Die Unterversion

Bedienfenster

Im Bedienfenster der Komponente werden die einzelnen Versionsangaben in lesbarer Form dargestellt.

SimulationLoad – Simulationslast

Symbol



Funktion

Der Komponententyp *SimulationLoad* zeigt die aktuelle Simulationslast. Informationen zur Simulationslast finden Sie im Abschnitt: Anzeige der Simulationslast (Seite 40).

Das Symbol zeigt unter "Total" die Gesamtlast in % an. Alle Werte in dieser Komponente sind nicht begrenzt und nicht geglättet. Deswegen schwanken die Werte möglicherweise stärker als der Wert im Symbol für die laufende Simulation.

Die drei Binäranzeigen *OK*, *Critical* und *Overload* geben eine grobe Bewertung des Lastwertes nach folgender Einteilung wieder:

Last < 40 %	OK
40 % < Last < 90 %	Critical
Last > 90 %	Overload

Die Zeitscheibe, in der die Komponente zur Auswertung der Last liegt, hat nur Einfluss darauf, wie oft die Werte aktualisiert werden. SIMIT wertet aber immer die Werte aller acht möglichen Zeitscheiben aus. Der ungünstigste Wert seit der letzten Aktualisierung wird verwendet.

Bedienfenster

Diese Komponente zeigt die Lastwerte für alle acht Zeitscheiben im Bedienfenster an. Sie unterscheidet zwischen folgenden Verursachern:

- Komponenten (Components)
- Lösungsverfahren (Solvers)
- Kopplungen (Gateways)

SIMIT unterscheidet zwischen *Time* und *CPU-Ticks*.

Time

Time ist das prozentuale Verhältnis von tatsächlicher Zykluszeit zu projektierter Zykluszeit.

Das Teilmodell, das den ungünstigsten Time-Wert aufweist, ist am nächsten an der Grenze zur Überlast. Das heißt aber nicht zwingend, dass in dieser Zeitscheibe zu viel gerechnet wird.

Die Rechenlast wird auf mehrere verfügbare Rechenkerne verteilt. Meistens muss ein Rechenkern aber mehr als eine Zeitscheibe abarbeiten. Zeitscheiben mit kürzeren Zykluszeiten haben eine höhere Priorität. Das bedeutet, dass die Berechnung einer langsameren Zeitscheibe unterbrochen werden kann, um die Berechnung einer schnelleren Zeitscheibe zu ermöglichen. Dies ist wichtig, um alle Zeitscheiben im Rahmen der eingestellten Zykluszeit zu rechnen, sofern möglich.

Das kann dazu führen, dass die Berechnung in einer langsamen Zeitscheibe ständig unterbrochen wird, um eine schnellere Zeitscheibe zu berechnen und dadurch ein Lastproblem entsteht. Die schnellere Zeitscheibe hingegen wird mit höherer Priorität berechnet und kann deswegen eine geringere Last haben.

CPU-Ticks

Um den eigentlichen Verursacher einer hohen Simulationslast besser zu identifizieren, stehen zusätzlich die *CPU-Ticks* zur Verfügung. Sie geben nicht die Zeit an, die zur Berechnung benötigt wurde, sondern sind ein Maß für die aufgewendeten Rechenoperationen. Sie berücksichtigen insbesondere keine Wartezeiten wegen Unterbrechungen durch andere Zeitscheiben. Da der gemessene absolute Wert der CPU-Ticks nicht sehr aussagekräftig ist, wird er normiert:

$$ModelTicks_i' = ModelTicks_i \frac{MAX_{n=1..8}\,(Modeltime_n + Solvertime_n)}{MAX_{n=1..8}\,(Modelticks_n + Solverticks_n)}$$

$$SolverTicks_i' = SolverTicks_i \frac{MAX_{n=1..8}\,(Modeltime_n + Solvertime_n)}{MAX_{n=1..8}\,(Modelticks_n + Solverticks_n)}$$

ModelTicks$_n$': Angezeigter Wert für die Rechenlast des Komponentencodes in Zeitscheibe *n*

SolverTicks$_n$': Angezeigter Wert für die Rechenlast des Lösungsverfahrens in Zeitscheibe *n*

ModelTicks$_n$: Gemessener Wert für die Rechenoperationen des Komponentencodes in der Zeitscheibe *n*

SolverTicks$_n$: Gemessener Wert für die Rechenoperationen des Lösungsverfahrens in der Zeitscheibe *n*

ModelTime$_n$: Verhältnis der benötigten Zeit für die Berechnung des Komponentencodes zur projektierten Zykluszeit in der Zeitscheibe *n*

SolverTime$_n$: Verhältnis der benötigten Zeit für die Berechnung des Lösungsverfahrens zur projektierten Zykluszeit in der Zeitscheibe *n*

Das Verhältnis der Ticks-Werte zueinander zeigt, wo die meisten Rechenoperationen pro Zeit durchgeführt werden. Sorgen Sie in der Zeitscheibe mit den höchsten Werten für Entlastung. Verlagern Sie z. B. Teile des Simulationsmodells in eine langsamere Zeitscheibe oder stellen Sie die gesamte kritische Zeitscheibe langsamer ein.

Simulationslast durch Kopplungen

Bei der Berechnung der Ticks-Werte wird davon ausgegangen, dass die Komponenten und Lösungsverfahren ausschließlich Berechnungen durchführen und keine Datei- oder

Konsolenausgaben ausführen. Diese Kommandos benötigen zwar Zeit, lasten aber die Rechenkerne nicht aus und führen daher zu irreführenden Angaben.

Deswegen werden die Kopplungen bei der Zählung der Rechenbefehle nicht berücksichtigt. Falls eine hohe Simulationslast durch eine Kopplung verursacht wird, dann kann das zwei Ursachen haben:

- Die Kopplung wird zu häufig von höherprioren Berechnungen unterbrochen.
- Die Kopplung selbst hat einen zu großen Zeitbedarf.

Im zweiten Fall ist eine Reduzierung des Rechenaufwands im Bereich der Komponenten oder Lösungsverfahren wenig hilfreich. In der Regel muss die Kopplung dann einer langsameren Zeitscheibe zugeordnet werden, der Signalumfang der Kopplung reduziert werden oder ein Teil der Signale in eine zweite Kopplung mit langsamerer Zykluszeit verlagert werden.

Ausgangssignale

Die Komponente *SimulationLoad* besitzt zur Ansteuerung der Controls im Bedienfenster Ausgangssignale, die am Komponentensymbol nicht verschaltbar sind, aber im Eigenschaftsfenster beobachtet werden können.

Ausgangssignal	Bedeutung
$LoadModel_nTicks$	Normierter Rechenaufwand für den Komponentencode in Zeitscheibe n
$LoadModel_nTime$	Prozentuales Verhältnis der benötigten Zeit für die Berechnung des Komponentencodes zur projektierten Zykluszeit in der Zeitscheibe n
$LoadSolver_nTicks$	Normierter Rechenaufwand für die Lösungsverfahren in Zeitscheibe n
$LoadSolver_nTime$	Prozentuales Verhältnis der benötigten Zeit für die Berechnung des Lösungsverfahrens zur projektierten Zykluszeit in der Zeitscheibe n
$LoadGateway_nTime$	Prozentuales Verhältnis der benötigten Zeit für die Berechnung der Kopplungen zur projektierten Zykluszeit in der Zeitscheibe n

Verknüpfung

Auf der Verknüpfung der Komponente wird die Last X über die Zeit gemittelt an der Digitalanzeige *Load* dargestellt.

$$\text{Anzeige } Load = \frac{1}{n}\sum_{i=0}^{n-1} X_{n-i} = \frac{X_n + X_{n-1} + \cdots + X_{n-(n-1)}}{n}$$

Die Anzahl der Gesamtschritte n wird gezählt. Die Anzahl der Zeitschritte, an denen am Binärsignal *OK* "TRUE" anliegt, wird intern gezählt (n_{ok}).

Die Digitalanzeige *OK* zeigt den prozentualen Anteil der Zeitschritte, an denen die Last im Bereich OK lag.

$$\text{Anzeige } OK = \frac{n_{ok}}{n}$$

Die Anzahl der Zeitschritte, an denen am Binärsignal *Critical* "TRUE" anliegt, wird intern gezählt ($n_{critical}$).

Die Digitalanzeige *Critical* zeigt den prozentualen Anteil der Zeitschritte, an denen die Last im Bereich *Critical* lag.

$$\text{Anzeige } Critical = \frac{n_{Critical}}{n}$$

Die Anzahl der Zeitschritte, an denen am Binärsignal *Overload* "TRUE" anliegt, wird intern gezählt ($n_{overload}$).

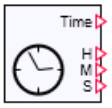
Die Digitalanzeige *Overload* zeigt den prozentualen Anteil der Zeitschritte, an denen die Last im Bereich *Overload* lag.

$$\text{Anzeige } Overload = \frac{n_{Overload}}{n}$$

Der Taster *Reset* startet die Berechnung von Neuem. Die Werte der vorherigen Zeitschritte werden ab diesem Zeitpunkt nicht mehr betrachtet und alle Zähler zurückgesetzt.

SimulationTime – Simulationszeit

Symbol



Funktion

Der Komponententyp *SimulationTime* gibt an seinen vier Ausgängen die aktuelle Simulationszeit aus. Die Ausgabe erfolgt am Ausgang *Time* als ganzzahliger (Integer) Wert in Millisekunden. An den ebenfalls ganzzahligen Ausgängen *H*, *M* und *S* steht die Simulationszeit aufgelöst in Stunden (*H*), Minuten (*M*) und Sekunden (*S*) zur Verfügung.

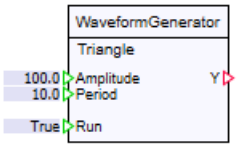
Die Simulationszeit ist die Zeit, die beim Durchführen der Simulation im Zyklus der Taktung der Simulation gezählt wird. Beim Initialisieren bzw. Starten einer Simulation wird die Simulationszeit auf null gesetzt. Simulationszeit und Realzeit laufen synchron, wenn die Simulation im Echtzeitmodus läuft. Im verlangsamten Simulationsmodus (Zeitlupe) läuft die Simulationszeit langsamer als die Realzeit und im beschleunigten Simulationsmodus (Zeitraffer) läuft die Simulationszeit schneller als die Realzeit. Wird die Simulation angehalten, dann wird auch die Simulationszeit angehalten. Wird die Simulation dann wieder fortgesetzt, dann läuft auch die Simulationszeit weiter.

Bedienfenster

Im Bedienfenster wird die Simulationszeit anzeigt. Sie können Sie die Simulationszeit mit *Reset* zurücksetzen.

WaveformGenerator – Funktionsgenerator

Symbol



Funktion

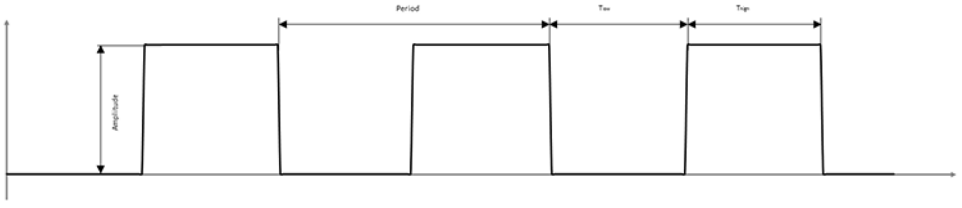
Der Komponententyp *WaveformGenerator* erzeugt periodische Signale am Ausgang *Y*. Der Eingang *Run* muss auf True gesetzt werden, damit der Ausgang *Y* dynamisch berechnet wird. Falls *Run* mit False belegt ist, bleibt der zuletzt berechnete Wert Y anstehen.

Der Eingang *Period* wird in Sekunden angegeben, die *Amplitude* ist ein absoluter Zahlenwert. Diese beiden Eingänge haben je nach Einstellung der *Waveform* eine etwas andere Bedeutung.

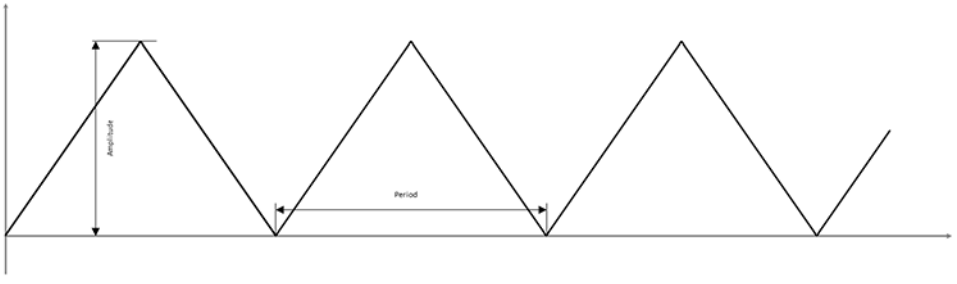
Der Eingang *SquareRate* ist nur bei der Einstellung *Square* aktiv und gibt das prozentuale Verhältnis von *Tlow* zu *Thigh* vor.

$$SquareRate\ [\%] = \frac{\mathrm{Tlow}}{Thigh}$$

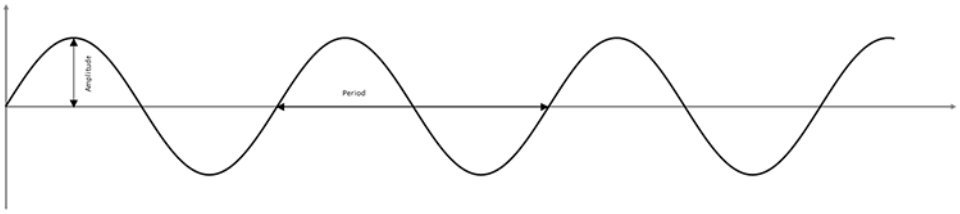
Square



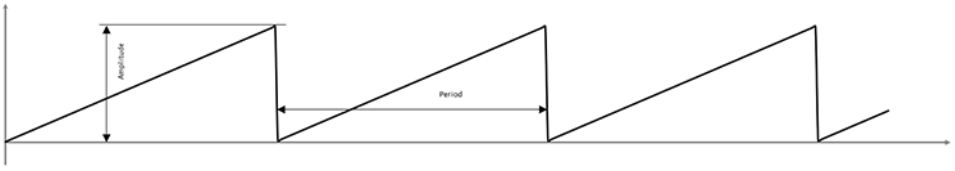
Triangle



Sine



Sawtooth



Parameter

Parametername	Beschreibung	Einheit	Standardwert
Waveform	Legt die Art der Funktion fest.	–	*Triangle*

8.1.4 Antriebskomponenten

8.1.4.1 Antriebe als Komponententypen

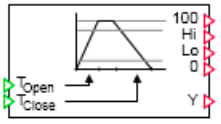
Im Verzeichnis *DRIVES* der Basisbibliothek sind Komponententypen zur Simulation von Antrieben enthalten. Diese Komponententypen bilden die Antriebsbibliothek. Es lassen sich damit allgemeine Antriebe für Ventile und Pumpen simulieren. Die Typen

- *DriveP1* und *DriveP2* sind für die Simulation von Pumpenantrieben, die Typen
- *DriveV1* bis *DriveV4* sind für die Simulation von Ventilantrieben konzipiert.

Im Unterverzeichnis *PROFIdrive* sind Komponententypen zur Simulation von dem PROFIdrive-Profil entsprechenden drehzahlgeregelten Antrieben enthalten. Komponententypen zur Simulation von Motormanagement- und Steuergeräten SIMOCODE pro finden Sie im Unterverzeichnis *SIMOCODEpro*.

8.1.4.2 Ventilantriebe

Es stehen vier Komponententypen *DriveV1*, *DriveV2*, *DriveV3* und *DriveV4* zur Verfügung, die in der Simulation als Nachbildung von Ventilantrieben eingesetzt werden können. Diesen vier Komponententypen gemeinsam sind die Ausgänge und ein Teil der Eingänge, wie in der folgenden Abbildung dargestellt.



An den beiden Analogeingängen *TOpen* und *TClose* werden die Öffnungs- bzw. Schließzeiten des Ventils, d. h. die Laufzeit des Antriebs zum vollständigen Öffnen bzw. Schließen, vorgegeben. Falls beim Durchführen der Simulation einer der beiden Eingangswerte negativ ist, wird die Meldung "*DriveVx: closing or opening time invalid value*" (Meldekategorie *ERROR*) erzeugt.

Am Analogausgang *Y* wird der aktuelle Positionswert des Antriebs als Prozentwert, d. h. im Bereich null bis hundert ausgegeben:

$0 \leq Y \leq 100$.

Der Wert null entspricht dem vollständig geschlossenen Ventil, der Wert hundert dem vollständig geöffneten Ventil.

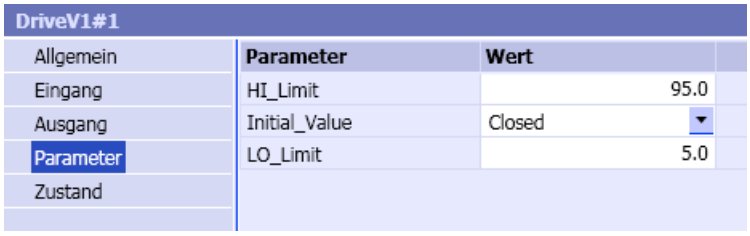
Die vier Binärausgänge *100*, *Hi*, *Lo* und *0* können als Endschalter des Antriebs interpretiert werden:

- *100* und *0* als Endschalter "Ventil vollständig auf" bzw. "Ventil vollständig zu",
- *Hi* und *Lo* als Vorendschalter "Ventil auf" bzw. "Ventil zu".

Die Endschalter sind dabei fest auf die Positionswerte null für das vollständig geschlossene Ventil und hundert für das vollständig geöffnete Ventil eingestellt. Die Binärausgänge *100* bzw. *0* werden demzufolge auf eins gesetzt, wenn das Ventil vollständig geschlossen bzw. geöffnet ist, also wenn der Positionswert *y* des Antriebs den Wert null bzw. hundert hat.

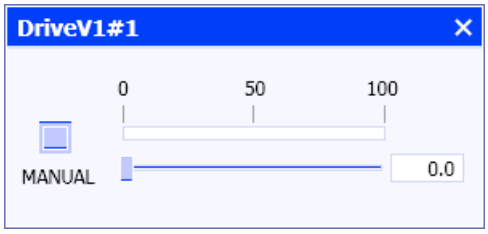
Die Vorendschalter *Hi* und *Lo* sind über die Parameter *HI_Limit* bzw. *LO_Limit* einstellbar. Ihre Werte liegen zwischen 0 und 100. Voreingestellt sind Positionswerte von fünf (*LO_Limit*) und 95 (*HI_Limit*).

Falls beim Durchführen der Simulation einer der beiden Parameterwerte negativ ist, wird die Meldung "*DriveVx: high and low parameters do not match*" (Meldekategorie *ERROR*) erzeugt.



Über den Parameter *Initial_Value* kann der Ventilantrieb auf die Anfangsposition *Closed* oder *Open* eingestellt werden. Die Vorbelegung für *Initial-Value* ist *Closed*.

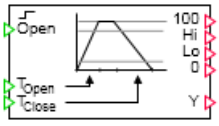
Im Bedienfenster der Komponenten ist der aktuelle Positionswert des Antriebs in Form einer Balkenanzeige visualisiert. Über den linken, mit *MANUAL* gekennzeichneten Schalter kann der Positionswert auf Vorgabe durch den ebenfalls im Bedienfenster verfügbaren Schieber umgeschaltet werden.



Die Bildung des Positionswerts wird dann nicht mehr aus den Eingängen der Komponente abgeleitet, sondern dem über den Schieber vorgegebenen Wert nachgeführt. Die vorgegebenen Öffnungs- und Schließzeiten bleiben dabei wirksam. Ebenfalls wirksam bleiben die vier Binärausgänge *100*, *Hi*, *Lo* und *0*. Sie werden auch im Fall der Nachführung des Positionswertes wie oben beschrieben gesetzt.

DriveV1 – Ventilantrieb vom Typ 1

Symbol

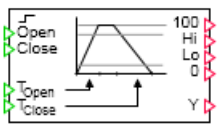


Funktion

Der Komponententyp *DriveV1* bildet eine Antriebseinheit nach, die den Positionswert *Y* am Ausgang in Abhängigkeit von einem Binärwert am Eingang *Open* setzt. Ist der Binäreingang gleich null, dann wird der Positionswert *Y* kontinuierlich mit der Schließzeit T_{Close} nach null verfahren. Ist der Binäreingang gleich eins, dann wird der Positionswert kontinuierlich mit der Öffnungszeit T_{Open} nach hundert verfahren. Der Positionswert hat damit nur die beiden stabilen Zustände null und hundert, d. h. bezogen auf die Ventilfunktion die Zustände "Ventil auf" und "Ventil zu". Mit jedem Wechsel des Binärwerts am Eingang wird der Antrieb bzw. das Ventil somit "aufgefahren" oder "zugefahren".

DriveV2 – Ventilantrieb vom Typ 2

Symbol



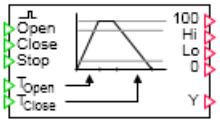
Funktion

Der Komponententyp *DriveV2* bildet eine Antriebseinheit nach, die den Positionswert *Y* am Ausgang in Abhängigkeit von zwei Binärwerten an den Eingängen *Open* und *Close* setzt. Ist der Binärwert am Eingang *Open* gleich eins, dann wird der Positionswert *Y* kontinuierlich mit der Öffnungszeit T_{Open} nach hundert verfahren. Ist der Binärwert am Eingang *Close* gleich eins, dann wird der Positionswert kontinuierlich mit der Öffnungszeit T_{Close} nach null verfahren.

Sind die Werte an beiden Eingängen gleichzeitig auf eins oder null gesetzt, dann bleibt der Positionswert unverändert. Der Positionswert ändert sich damit nur, wenn entweder der Binärwert am Eingang *Open* oder am Eingang *Close* auf eins gesetzt ist.

DriveV3 – Ventilantrieb vom Typ 3

Symbol

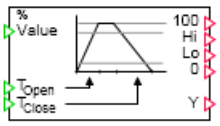


Funktion

Der Komponententyp *DriveV3* bildet eine Antriebseinheit nach, die den Positionswert *Y* am Ausgang in Abhängigkeit von Binärwerten an den drei Eingängen *Open*, *Close* und *Stop* setzt. Ändert der Binärwert am Eingang *Open* seinen Wert von null nach eins, dann wird der Positionswert *Y* kontinuierlich mit der Öffnungszeit T_{Open} nach hundert verfahren. Der Positionswert wird kontinuierlich mit der Schließzeit T_{Close} nach null verfahren wenn der Binärwert am Eingang *Close* seinen Wert von null nach eins ändert. Ändert der Binärwert am Eingang *Stop* seinen Wert von null nach eins, dann bleibt der Positionswert unverändert. Das Öffnen, Schließen und Stoppen des Antriebs wird damit über die Flanke des entsprechenden binären Eingangssignals (Signalwechsel von null nach eins) initiiert.

DriveV4 – Ventilantrieb vom Typ 4

Symbol

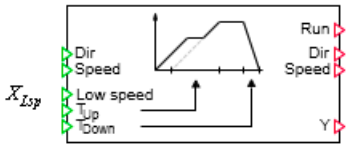


Funktion

Der Komponententyp *DriveV4* bildet eine Antriebseinheit nach, die den Positionswert *Y* am Ausgang dem Analogwert am Eingang *Value* kontinuierlich nachführt. Die Nachführung erfolgt in Richtung steigender Positionswerte mit der Öffnungszeit T_{Open} und in Richtung fallender Positionswerte mit der Schließzeit T_{Close}. Der Positionswert bleibt in jedem Fall auf Werte von null bis hundert beschränkt, auch wenn der Eingangswert außerhalb dieses Intervalls liegt.

8.1.4.3 Pumpenantriebe und Lüfterantriebe

Die beiden Komponententypen *DriveP1* und *DriveP2* können in der Simulation als Antriebe für Pumpen, Lüfter oder vergleichbare Aggregate eingesetzt werden. Den beiden Komponententypen gemeinsam sind die Ausgänge und ein Teil der Eingänge wie in der folgenden Abbildung dargestellt:



An den beiden Analogeingängen T_{Up} und T_{Down} werden die Hoch- bzw. Runterlaufzeiten des Antriebs vorgegeben. T_{Up} ist die Laufzeit des Antriebs zum Hochfahren vom Stillstand auf die Nenndrehzahl in Sekunden, T_{Down} ist die Zeit zum Abfahren des Antriebs von Nenndrehzahl bis zum Stillstand in Sekunden. Voreingestellt sind beide Zeiten mit einer Sekunde. Falls beim Durchführen der Simulation einer der beiden Eingangswerte negativ ist, wird die Meldung "*DriveVx: run-up or run-down time invalid value*" (Meldekategorie *ERROR*) erzeugt.

Am Analogausgang *Y* wird der aktuelle Drehzahlwert des Antriebs als Prozentwert, d. h. im Bereich null bis hundert ausgegeben:

$0 \leq Y \leq 100$.

Der Wert null entspricht dem stillstehenden Antrieb, der Wert hundert dem Antrieb auf Nenndrehzahl.

Über den Binäreingang *Dir* ist die Drehrichtung des Antriebs vorgebbar. Ist dieser Eingang auf null gesetzt, dann ist die positive Drehrichtung vorgegeben. Ist der Eingang auf eins gesetzt, dann ist die Drehrichtung negativ vorgegeben. Es kann über diesen Eingang also beispielsweise Rechts- und Linkslauf des Antriebs vorgegeben werden. Bei Drehung in negativer Richtung wird die Drehzahl am Ausgang *Y* als negativer Wert ausgegeben:

$-100 \leq Y \leq 0$.

Positive Werte von *Y* kennzeichnen somit Drehzahlen in positive Richtung, negative Werte Drehzahlen in negative Richtung. Die Drehrichtungsumschaltung wird nur wirksam, wenn der Antrieb steht.

Über den Binäreingang *Speed* kann die Drehzahl zwischen Nenndrehzahl (Volldrehzahl) und Teildrehzahl umgeschaltet werden. Falls *Speed* auf eins gesetzt ist, ist Nenndrehzahl angewählt, andernfalls ist Teildrehzahl angewählt. Voreinstellung für den Binäreingang *Speed* ist eins. DieTeildrehzahl wird als Zahlenwert (Prozentwert) am Analogeingang x_{LSp} vorgegeben:

$0 \leq x_{LSp} \leq 100$.

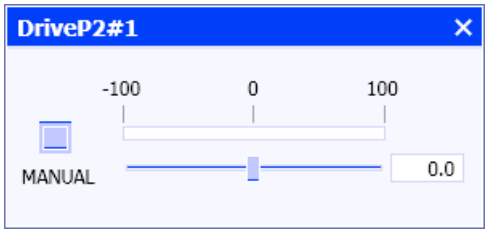
Die Voreinstellung für die Teildrehzahl ist 50, also halbe Nenndrehzahl. Falls die Teildrehzahl nicht im definierten Bereich liegt, wird die Meldung "*DrivePx: low speed invalid value*" (Meldekategorie *ERROR*) erzeugt.

Der Binärausgang *Run* ist nur dann auf eins gesetzt, wenn der Antrieb den vorgewählten Drehzahlwert in positive oder negative Drehrichtung erreicht hat, also nur, wenn der Absolutwert am Analogausgang *Y* gleich hundert (Nenndrehzahl) oder gleich der am Eingang x_{LSp} vorgegeben Teildrehzahl ist.

Am Binärausgang *Speed* steht das Rückmeldesignal für die Drehzahlanwahl zur Verfügung. Der Binärausgang wird nur auf eins gesetzt, wenn der Antrieb seine Nenndrehzahl in positive oder negative Drehrichtung erreicht hat.

Der Wert am Binärausgang *Dir* wird als Rückmeldesignal der aktuellen Drehrichtung des Antriebs gebildet. Der Binärausgang ist null, wenn der Antrieb in positive Richtung dreht; der Wert ist eins, wenn der Antrieb in negative Richtung dreht.

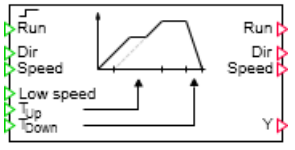
Im Bedienfenster der Komponenten ist der aktuelle Drehzahlwert des Antriebs in Form einer Balkenanzeige visualisiert. Über den linken, mit *MANUAL* gekennzeichneten Schalter kann der Drehzahlwert auf Vorgabe durch den ebenfalls im Bedienfenster verfügbaren Schieber umgeschaltet werden.



Die Bildung des Drehzahlwerts wird dann nicht mehr aus den Eingängen der Komponente abgeleitet, sondern dem über den Schieber vorgegebenen Wert nachgeführt. Die vorgegebenen Hoch- und Runterlaufzeiten bleiben dabei wirksam. Ebenfalls wirksam bleiben die Binärausgänge *Run*, *Speed* und *Dir*. Sie werden auch im Fall der Nachführung des Drehzahlwertes wie oben beschrieben gesetzt.

DriveP1 – Pumpenantrieb vom Typ 1

Symbol

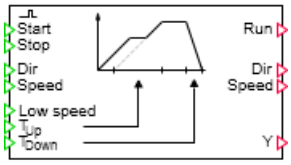


Funktion

Der Komponententyp *DriveP1* bildet eine Antriebseinheit nach, die über den Binärwert am Eingang *Run* ein- und ausgeschaltet wird. Solange der Eingangswert auf eins gesetzt ist, ist der Antrieb eingeschaltet. Wird der Eingangswert auf null gesetzt, dann wird der Antrieb ausgeschaltet.

DriveP2 – Pumpenantrieb vom Typ 2

Symbol



Funktion

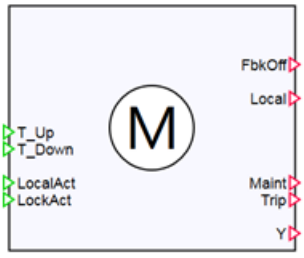
Der Komponententyp *DriveP2* bildet eine Antriebseinheit nach, die über die beiden Binäreingänge *Start* und *Stop* ein- und ausgeschaltet wird. Ändert sich der Binärwert am Eingang *Start* von null nach eins, dann wird der Antrieb eingeschaltet. Bei einer Änderung des Binärwerts am Eingang *Stop* von null nach eins wird der Antrieb ausgeschaltet. Das Ein- und Ausschalten des Antriebs wird damit über die Flanke des entsprechenden binären Eingangssignals (Signalwechsel von null nach eins) initiiert.

8.1.4.4 Motor

Gemeinsamkeiten der Motorkomponenten

Gemeinsame Anschlüsse der Komponententypen für Pumpenantriebe

Die Komponententypen *Motor, ReversibleMotor* und *TwoSpeedMotor* können in der Simulation als Antriebe für Pumpen, Lüfter oder vergleichbare Aggregate eingesetzt werden. Ein Teil der Eingänge und Ausgänge ist bei allen Komponenten gleich, wie in der folgenden Abbildung dargestellt.



Analogeingänge

An den Analogeingängen *T_UP* und *T_DOWN* werden die Hoch- bzw. Runterlaufzeiten des Antriebs vorgegeben.

T_UP ist die Laufzeit des Antriebs zum Hochfahren vom Stillstand auf die Nenndrehzahl in Sekunden, *T_DOWN* ist die Zeit zum Abfahren des Antriebs von Nenndrehzahl bis zum Stillstand in Sekunden. Die voreingestellte Zeit beträgt 1 s.

Falls beim Durchführen der Simulation einer der beiden Eingangswerte negativ ist, wird die Meldung "*Motor: run-up or run-down time invalid value*" (Meldekategorie *ERROR*) erzeugt.

Binäreingänge

Mit dem Binäreingang *LocalAct* wird der Antrieb in den lokalen Vor-Ort-Bedienmodus geschaltet. In diesem Modus kann der Antrieb ausschließlich über das Bedienfenster der Komponente gesteuert werden. Mit dem Binäreingang *LockAct* kann der Antrieb verriegelt werden, so dass weder eine lokale Vor-Ort-Bedienung noch eine Bedienung über die Steuereingänge der Komponente möglich ist. Wenn die Komponente während des Betriebs eine positive Flanke auf den Eingang *LockAct* bekommt, wird der Antrieb abgesteuert und verriegelt.

Binärausgänge

Der Binärausgang *FbkOff* ist True, wenn der Antrieb stillstehend ist (Y = 0). Der Binärausgang *Local* kann über das Bedienfenster des Antriebs geschaltet werden. Wenn der Ausgang entsprechend verschaltet ist, kann darüber der Steuerung der Vor-Ort-Betrieb mitgeteilt werden. Der Binärausgang *Maint* kann über das Bedienfenster des Antriebs geschaltet werden. Der Binärausgang *Trip* kann über das Bedienfenster des Antriebs geschaltet werden. Wenn Trip aktiviert wird, steuert der Antrieb ab und verbleibt stillstehend (Y =0).

Analogausgang

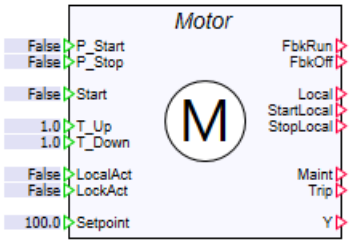
Am Analogausgang Y wird der aktuelle Drehzahlwert des Antriebs als Prozentwert, d. h. im Bereich 0 bis 100 ausgegeben:

0 ≤ Y ≤ 100 bzw. beim ReversibleMotor −100 ≤ Y ≤ 100.

Der Wert 0 entspricht einem stillstehenden Antrieb, der Wert 100 einem Antrieb auf Nenndrehzahl (bzw. −100 beim ReversibleMotor).

Motor

Symbol



Funktion

Der Komponententyp *Motor* bildet eine Antriebseinheit nach, die über den Binärwert am Eingang *Start* ein- und ausgeschaltet wird. Solange der Start-Eingangswert auf True gesetzt ist, ist der Antrieb eingeschaltet. Wenn der Start-Eingangswert auf False gesetzt wird, wird der Antrieb ausgeschaltet.

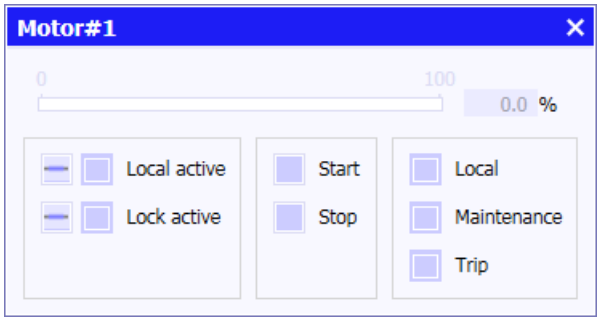
Außerdem kann die Komponente auch über die beiden Binäreingänge *P_Start* und *P_Stop* ein- und ausgeschaltet werden. Wenn sich der Binärwert am Eingang *P_Start* von False auf True ändert, wird der Antrieb eingeschaltet. Bei einer Änderung des Binärwerts am Eingang *P_Stop* von False nach True wird der Antrieb ausgeschaltet.

Über den Analogeingang *Setpoint* kann der Drehzahlwert des Antriebs als Prozentwert im Bereich 0 bis 100 angegeben werden. Standardmäßig ist die Komponente mit 100 vorbelegt. Falls der Drehzahlwert nicht in diesem definierten Bereich liegt, wird die Meldung "*Motor: Setpoint invalid value*" (Meldekategorie *ERROR*) erzeugt.

Der Binärausgang *FbkRun* ist nur dann auf True gesetzt, wenn der Antrieb den vorgewählten Drehzahlwert in positive Drehrichtung erreicht hat: Der Wert am Analogausgang *Y* und der am Eingang *Setpoint* vorgegebene Wert sind gleich.

Die Binärausgänge *StartLocal* und *StopLocal* stehen nur für die Dauer des Tastendrucks der Tasten *Start* bzw. *Stop* auf dem Bedienfenster der Komponente auf True.

Bedienfenster



Im Bedienfenster der Komponente ist der aktuelle Drehzahlwert des Antriebs in Form einer Balkenanzeige visualisiert.

In der linken Spalte werden die beiden Eingänge *LocalAct* und *LockAct* dargestellt. An dieser Stelle sehen Sie den aktuell anstehenden Zustand.

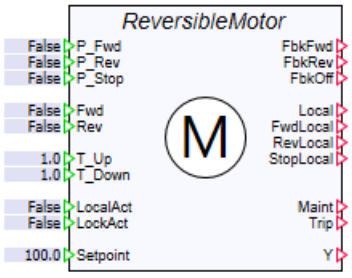
Zusätzlich haben Sie die Möglichkeit, beide Eingänge von deren verbundenen Quellen abzutrennen und den jeweiligen Wert manuell vorzugeben.

Wenn *LocalAct* (Local active) aktiv ist, wird die Bildung des Drehzahlwerts nicht mehr aus den Eingängen der Komponente abgeleitet, sondern über die Vor-Ort-Bedientaster in der mittleren Spalte *Start* und *Stop*. Die vorgegebenen Hoch- und Runterlaufzeiten bleiben dabei wirksam. Die Umschaltung von Fern auf Vor-Ort erfolgt dabei stoßfrei: Wenn der Antrieb vor der Umschaltung eingeschaltet war, ist er zunächst auch unmittelbar nach der Umschaltung eingeschaltet.

In der rechten Spalte befinden sich die Bedienschalter für *Local*, Maintenance (*Maint*) und *Trip*.

ReversibleMotor – Motor mit 2 Richtungen

Symbol



Funktion

Der Komponententyp *ReversibleMotor* bildet eine Antriebseinheit nach, die über zwei Binärwerte an den Eingängen *Fwd* und *Rev* gesteuert wird. Solange der Eingangswert *Fwd* auf True gesetzt ist, wird der Antrieb im Vorwärtsbetrieb eingeschaltet. Wenn der Eingangswert auf False gesetzt wird, wird der Antrieb ausgeschaltet. Solange der Eingangswert *Rev* auf True gesetzt ist, wird der Antrieb im Rückwärtsbetrieb eingeschaltet. Wenn der Eingangswert auf False gesetzt wird, wird der Antrieb ausgeschaltet.

Außerdem kann die Komponente auch über die drei Binäreingänge *P_Fwd, P_Rev* und *P_Stop* per Pulsbefehl angesteuert werden. Wenn sich der Binärwert am Eingang *P_Fwd* von False auf True ändert, wird der Antrieb im Vorwärtsbetrieb eingeschaltet. Bei einer Änderung des Binärwerts am Eingang *P_Stop* von False auf True wird der Antrieb ausgeschaltet. Wenn sich der Binärwert am Eingang *P_Rev* von False auf True ändert, wird der Antrieb im Rückwärtsbetrieb eingeschaltet.

Die Drehrichtungsumschaltung wird nur wirksam, wenn der Antrieb steht.

Über den Analogeingang *Setpoint* kann der Drehzahlwert des Antriebs als Prozentwert im Bereich 0 bis 100 angegeben werden. Standardmäßig ist die Komponente mit 100 vorbelegt. Falls der Drehzahlwert nicht im definierten Bereich liegt, wird die Meldung "*Motor: Setpoint invalid value*" (Meldekategorie *ERROR*) erzeugt.

Der Binärausgang *FbkFwd* ist nur dann auf True gesetzt, wenn der Antrieb den vorgewählten Drehzahlwert in positive Drehrichtung erreicht hat (Vorwärtsbetrieb): Der Wert am Analogausgang *Y* und der am Eingang *Setpoint* vorgegebene Wert sind gleich.

Der Binärausgang *FbkRev* ist nur dann auf True gesetzt, wenn der Antrieb den vorgewählten Drehzahlwert in negative Drehrichtung erreicht hat (Rückwärtsbetrieb): Der Wert am Analogausgang *Y* entspricht dem am Eingang *Setpoint* vorgegeben Wert im Negativen.

Die Binärausgänge *FwdLocal*, *RevLocal* und *StopLocal* stehen nur für die Dauer des Tastendrucks der Tasten *Forward* bzw. *Reverse* oder *Stop* auf dem Bedienfenster der Komponente auf True.

Bedienfenster



Im Bedienfenster der Komponente ist der aktuelle Drehzahlwert des Antriebs in Form einer Balkenanzeige visualisiert.

In der linken Spalte werden die beiden Eingänge *LocalAct* und *LockAct* dargestellt. An dieser Stelle sehen Sie den aktuell anstehenden Zustand.

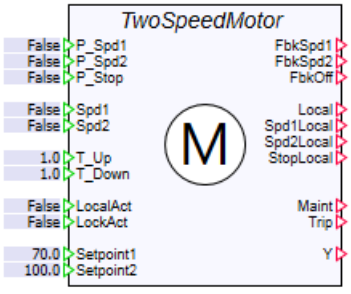
Zusätzlich haben Sie die Möglichkeit, beide Eingänge von deren verbundenen Quellen abzutrennen und den jeweiligen Wert manuell vorzugeben.

Wenn *LocalAct* (Local active) aktiv ist, wird die Bildung des Drehzahlwerts nicht mehr aus den Eingängen der Komponente abgeleitet, sondern über die Vor-Ort-Bedientaster in der mittleren Spalte *Forward, Reverse* und *Stop*. Die vorgegebenen Hoch- und Runterlaufzeiten bleiben dabei wirksam. Die Umschaltung von Fern auf Vor-Ort erfolgt dabei stoßfrei: Wenn der Antrieb vor der Umschaltung eingeschaltet war, ist er zunächst auch unmittelbar nach der Umschaltung eingeschaltet.

In der rechten Spalte befinden sich die Bedienschalter für *Local*, Maintenance (*Maint*) und *Trip*.

TwoSpeedMotor – Motor mit 2 Geschwindigkeiten

Symbol



Funktion

Der Komponententyp *TwoSpeedMotor* bildet eine Antriebseinheit nach, die über zwei Binärwerte an den Eingängen *Spd1* und *Spd2* gesteuert wird. Solange der Eingangswert *Spd1* auf True gesetzt ist, wird der Antrieb eingeschaltet und bis *Setpoint1* betrieben. Wenn der Eingangswert auf False gesetzt wird, wird der Antrieb ausgeschaltet. Solange der Eingangswert *Spd2* auf True gesetzt ist, wird der Antrieb eingeschaltet und bis *Setpoint2* betrieben. Wenn der Eingangswert auf False gesetzt wird, wird der Antrieb ausgeschaltet. Wenn beide Eingänge *Spd1* und *Spd2* gleichzeitig anstehen, wird der Antrieb mit *Spd1* verfahren.

Außerdem kann die Komponente auch über die drei Binäreingänge *P_Spd1, P_Spd2* und *P_Stop* per Pulsbefehl angesteuert werden. Wenn sich der Binärwert am Eingang *P_Spd1* von False auf True ändert, wird der Antrieb mit dem Teildrehzahlwert als Prozentwert eingeschaltet, der beim Eingang *Setpoint1* angegeben ist. Bei einer Änderung des Binärwerts am Eingang *P_Stop* von False auf True wird der Antrieb ausgeschaltet. Wenn sich der Binärwert am Eingang *P_Spd2* von False auf True ändert, wird der Antrieb mit dem Volldrehzahlwert eingeschaltet, der beim Eingang *Setpoint2* angegeben ist (max. 100). Eine direkte Umschaltung zwischen *Spd1* und *Spd2* ist möglich.

Über den Analogeingang *Setpoint1* kann der Teildrehzahlwert des Antriebs als Prozentwert im Bereich 0 bis 100 angegeben werden. Standardmäßig ist dieser Eingang mit 70 vorbelegt. Am Analogeingang *Setpoint2* kann der Volldrehzahlwert des Antriebs als Prozentwert im Bereich 0 bis 100 angegeben werden. Standardmäßig ist dieser Eingang mit 100 vorbelegt. Falls die Drehzahlwerte nicht in diesen definierten Bereichen liegen, wird die Meldung "*Motor: Setpoint invalid value*" (Meldekategorie *ERROR*) erzeugt. Beide Setpoints können unabhängig voneinander eingestellt werden.

Der Binärausgang *FbkSpd1* ist nur dann auf True gesetzt, wenn der Antrieb den vorgewählten Drehzahlwert erreicht hat: Der Wert am Analogausgang *Y* und der am Eingang *Setpoint1* vorgegebene Wert sind gleich.

Der Binärausgang *FbkSpd2* ist nur dann auf True gesetzt, wenn der Antrieb den vorgewählten Drehzahlwert erreicht hat: Der Wert am Analogausgang *Y* und der am Eingang *Setpoint2* vorgegebene Wert sind gleich.

Die Binärausgänge *Spd1Local*, *Spd2Local* und *StopLocal* stehen nur für die Dauer des Tastendrucks der Tasten *Speed 1* bzw. *Speed 2* oder *Stop* auf dem Bedienfenster der Komponente auf True.

Bedienfenster



Im Bedienfenster der Komponente ist der aktuelle Drehzahlwert des Antriebs in Form einer Balkenanzeige visualisiert.

In der linken Spalte werden die beiden Eingänge *LocalAct* und *LockAct* dargestellt. An dieser Stelle sehen Sie den aktuell anstehenden Zustand.

Zusätzlich haben Sie die Möglichkeit, beide Eingänge von deren verbundenen Quellen abzutrennen und den jeweiligen Wert manuell vorzugeben.

Wenn *LocalAct* (Local active) aktiv ist, wird die Bildung des Drehzahlwerts dann nicht mehr aus den Eingängen der Komponente abgeleitet, sondern über die Vor-Ort-Bedientaster in der mittleren Spalte *Speed 1, Speed 2* und *Stop*. Die vorgegebenen Hoch- und Runterlaufzeiten bleiben dabei wirksam. Die Umschaltung von Fern auf Vor-Ort erfolgt dabei stoßfrei: Wenn der Antrieb vor der Umschaltung mit Speed 1 eingeschaltet war, dann ist er zunächst auch unmittelbar nach der Umschaltung mit Speed 1 eingeschaltet.

In der rechten Spalte befinden sich die Bedienschalter für *Local*, Maintenance (*Maint*) und *Trip*.

8.1.4.5 PROFIdrive-Antriebe

Viele drehzahlgeregelte Antriebe werden direkt über PROFIBUS DP oder PROFINET mit der Steuerung verbunden. Um die Projektierung solcher Antriebe zu vereinheitlichen, wurde von der PROFIBUS Nutzerorganisation (PNO) ein Profil namens PROFIdrive geschaffen. Dieses Profil legt für einige wesentliche Funktionen fest, wie sich ein Antrieb, der diesem Profil entspricht, am Feldbus verhält. Die folgenden Stromrichter bzw. Umrichter aus dem Siemens-Spektrum erfüllen unter anderem die Anforderungen eines PROFIdrives:

- SIMOREG DC Master
- SIMOVERT Masterdrive
- Micromaster
- Sinamics

Allen PROFIdrive-Antrieben gemeinsam ist, dass sie über Steuerworte (STW) angesteuert werden und Rückmeldungen in Zustandsworten (ZSW) zur Verfügung stellen. Außerdem erhalten die Antriebe eine Soll-Drehzahl von der Steuerung und melden eine Ist-Drehzahl und eine Ist-Position zurück.

Die PROFIdrive-Funktionalität lässt sich grob in folgende Blöcke untergliedern:

- Zustandsautomat
- Rampengenerator für die übertragene Soll-Drehzahl
- Positions – Istwertaufbereitung
- Positionsregelung (nicht in allen PROFIdrive-Telegramme enthalten)
- Momentenbegrenzung (nur für SIEMENS spezifische Telegramme).

Der Zustandsautomat legt die Übergänge von einem Zustand in einen anderen fest. Ein Zustandsübergang wird durch ein definiertes Bit im Steuerwort 1 angestoßen. Der aktuelle Zustand ist im Zustandswort 1 erkennbar. Der Rampengenerator eines PROFIdrive dient dazu, Sprünge der Soll-Drehzahl in lineare Rampenanstiege mit einstellbarer Steigung umzuwandeln.

Über diese Grundfunktionen hinaus können PROFIdrive-Antriebe natürlich erheblich mehr Funktionen und Parametriermöglichkeiten aufweisen.

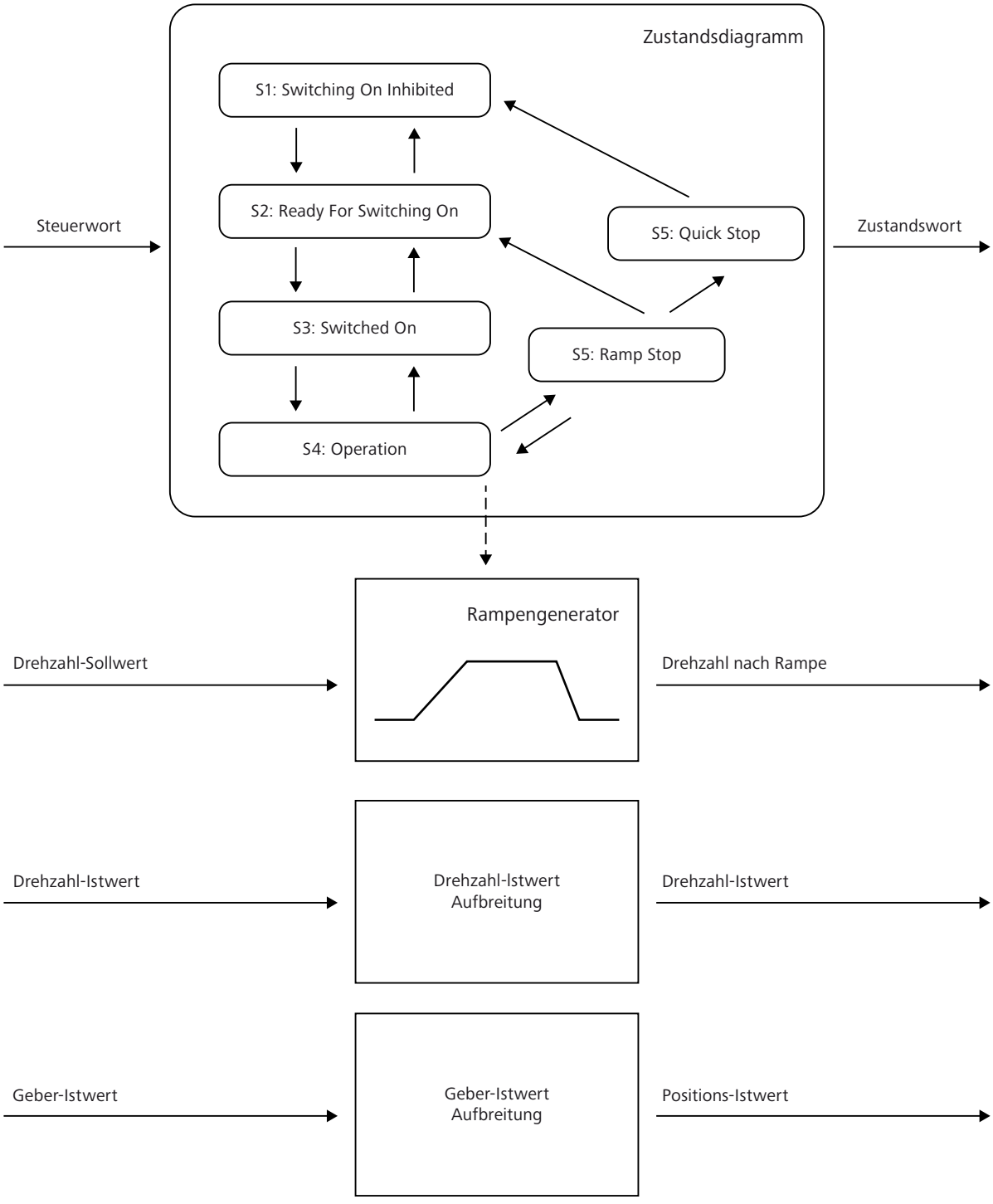
Verwendung der PROFIdrive-Bibliothek zur Simulation von PROFIdrive-Antrieben

Alle Komponententypen zur Simulation von PROFIdrive-Antrieben sind im Verzeichnis *PROFIdrive* der Antriebsbibliothek zusammengefasst:

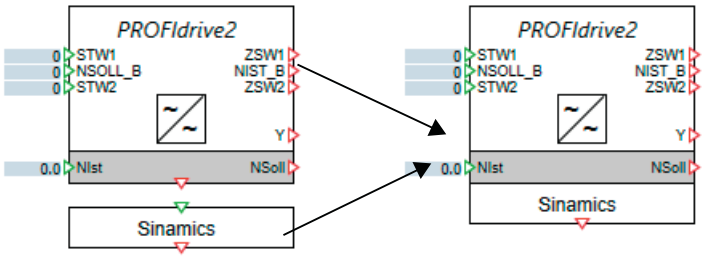
- Der PROFIdrive-Kopfkomponententyp *PROFIdrive2*
- Die Gebersimulation für rotatorische und lineare Geber
- Die Geberdatenaufbereitung
- Die Positionsregelung im Antrieb
- Die Auswertung der Momentenreduktion für SIEMENS-Geräte
- Die Unterstützung von Safety-Funktionalität im Antrieb
- Die Adaption von gerätespezifischen Daten mit der Steuerung
- die PROFIdrive-Rumpfkomponententypen *Sinamics* und *Universal*.

Diese Komponententypen bilden die PROFIdrive-Bibliothek.

Im Komponententyp *PROFIdrive* sind die in allen PROFIdrive-Antrieben enthaltenen Funktionen Zustandsautomat und Rampengenerator nachgebildet.



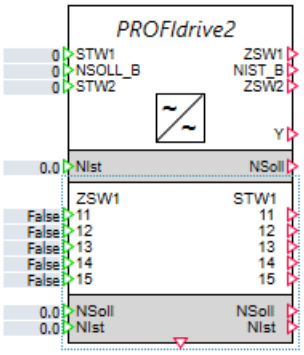
Im PROFIdrive-Profil sind nicht alle Bits des Steuer- und Zustandswortes in ihrer Bedeutung festgelegt. Die fünf Bits 11 bis 15 im Steuer- und Zustandswort 1 sind antriebsspezifisch verwendbar und haben damit eine je nach Antriebstyp unterschiedliche Bedeutung. Für die AC-Umrichter SINAMICS und Universal sind diese antriebsspezifischen Funktionen in entsprechenden Rumpfkomponenten nachgebildet. Diese antriebsspezifischen Rumpfkomponenten werden einfach an die Kopfkomponente *PROFIdrive* angesetzt, wie in der folgenden Abbildung verdeutlicht.



Die Rumpfkomponententypen der *PROFIdrive*-Bibliothek enthalten Nachbildungen verschiedener PROFIdrive-Antriebe in einer Standardparametrierung. Weitere Informationen hierzu finden Sie im Kapitel: Sinamics – Frequenzumrichter SINAMICS (Seite 523).

Die Bedeutung der antriebsspezifischen Bits kann durch Parametrierung der Umrichtersysteme verändert werden und damit von der hier nachgebildeten Standardparametrierung abweichen. Falls dies der Fall ist, können Sie den Rumpfkomponententyp *Universal* einsetzen.

Weitere Informationen hierzu finden Sie im Kapitel: Universal – Ergänzungen der PROFIdrive-Grundfunktion (Seite 518). Dieser Komponententyp stellt Ihnen die Bits 11 bis 15 des Steuerwortes 1 und des Zustandswortes 1 als Komponentenausgänge und -eingänge zur freien Verschaltung mit weiteren Komponenten der Basisbibliothek zur Verfügung.



Neben Steuer- und Zustandswort sowie Drehzahl-Sollwert und Drehzahl-Istwert können PROFIdrive-Antriebe weitere Informationen mit der Steuerung austauschen. Diese zusätzlichen E/A-Signale stehen Ihnen in SIMIT über die SIMIT Unit-Kopplung wie üblich als E/A-Signale zur Verfügung und können damit auch zur spezifischen Anpassung einer Antriebssimulation benutzt werden.

PROFIdrive-Antriebe – darstellbare PROFIdrive-Telegramme

In der Basisbibliothek sind mehrere PROFIdrive-Komponenten enthalten. Mit ihnen können folgende PROFIdrive-Standard-Telegramme, sowie SIEMENS spezifisch erweiterte Telegramme abgebildet werden.

	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6	PZD 7	PZD 8	PZD 9	PZD 10	PZD 11	PZD 12	PZD 13	PZD 14	PZD 15
TEL_1 Soll	STW 1	NSOLL_A													
TEL_1 Ist	ZSW 1	NIST_A													
TEL_2 Soll	STW 1	NSOLL_B		STW 2											
TEL_2 Ist	ZSW 1	NIST_B		ZSW 2											
TEL_3 Soll	STW 1	NSOLL_B		STW 2	G1_STW										
TEL_3 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2							
TEL_4 Soll	STW 1	NSOLL_B		STW 2	G1_STW	G2_STW									
TEL_4 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2		G2_ZSW	G2_XIST 1		G2_XIST 2		
TEL_5 Soll	STW 1	NSOLL_B		STW 2	G1_STW	XERR		KPC							
TEL_5 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2							
TEL_6 Soll	STW 1	NSOLL_B		STW 2	G1_STW	G2_STW	XERR		KPC						
TEL_6 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2		G2_ZSW	G2_XIST 1		G2_XIST 2		
TEL_102 Soll	STW 1	NSOLL_B		STW 2	M_Red	G1_STW									
TEL_102 Ist	ZSW 1	NIST_B		ZSW 2	Meld_W	G1_ZSW	G1_XIST 1		G1_XIST 2						
TEL_103 Soll	STW 1	NSOLL_B		STW 2	M_Red	G1_STW	G2_STW								
TEL_103 Ist	ZSW 1	NIST_B		ZSW 2	Meld_W	G1_ZSW	G1_XIST 1		G1_XIST 2		G2_ZSW	G2_XIST 1		G2_XIST 2	
TEL_105 Soll	STW 1	NSOLL_B		STW 2	M_Red	G1_STW	XERR		KPC						
TEL_105 Ist	ZSW 1	NIST_B		ZSW 2	Meld_W	G1_ZSW	G1_XIST 1		G1_XIST 2						
TEL_106 Soll	STW 1	NSOLL_B		STW 2	M_Red	G1_STW	G2_STW	XERR		KPC					
TEL_106 Ist	ZSW 1	NIST_B		ZSW 2	Meld_W	G1_ZSW	G1_XIST 1		G1_XIST 2		G2_ZSW	G2_XIST 1		G2_XIST 2	

Die Telegramme werden durch Aneinanderreihung der Einzelkomponenten modelliert.

In den folgenden Abschnitten werden die „Bausteine“ – die verfügbaren Einzelkomponenten näher erläutert.

Eine besondere Rolle kommt dabei der Komponente „Safety enhancement“ zu, welche die Standard-PROFIdrive-Antriebe um Safety-Funktionen erweitert.

Zusätzlich wird für die Nachbildung einer aktiven Einspeisung eine Komponente für das SIEMENS spezifische Telegramm 370 angeboten (siehe InFeed_Tel_370 (Seite 520)).

DataAdaption – Funktionalität der Datenadaption eines PROFIdrive-Antriebs

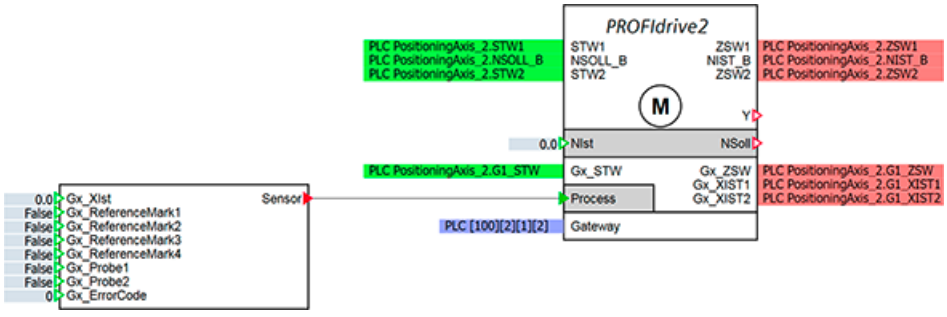
Symbol



Funktion

Technologische Objekte in der S7-1500 haben eine Funktionalität, mit der sie ihre technologischen Parameter mit den technologischen Parametern des Antriebs abgleichen. Dieser Prozess wird als Datenadaption genannt. Der Komponententyp *DataAdaption* liefert mittels azyklischer Datensatzkommunikation die notwendigen Informationen an das Technologieobjekt.

Die Komponente DataAdaption ist immer als letztes an einen simulierten Antrieb anzufügen.



Anschlüsse

Eingang	Beschreibung
Gateway	Anschluss zur Konfiguration der Antriebskomponente über einen Unit-Konnektor
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
ReferenceSpeed	Geben Sie hier die Referenzdrehzahl an.
MaxSpeed	Geben Sie hier die physikalisch maximale Drehzahl des Antriebs an
G1_EncoderSystem	Geben Sie hier die Art des Gebersystems an. 0 = rotatorisches Gebersystem 1 = lineares Gebersystem
G1_SupportsAbsolutValues	Geben Sie hier an, ob Geber 1 ein True = Absolutwertgeber False = Inkrementalgeber ist.

G1_DeterminableRevolutions	Geben Sie hier an, wie viele Geberumdrehungen für den Absolutwert unterschieden werden können.
G1_FineResolutionXist1	Geben Sie hier die Feinauflösung von G1_Xist1 an.
G1_FineResolutionXist2	Geben Sie hier die Feinauflösung von G1_Xist2 an.
G2_EncoderSystem	Geben Sie hier die Art des Gebersystems an. 0 = rotatorisches Gebersystem 1 = lineares Gebersystem
G2_SupportsAbsolutValues	Geben Sie hier an, ob Geber 2 ein True = Absolutwertgeber False = Inkrementalgeber ist.
G2_DeterminableRevolutions	Geben Sie hier an, wie viele Geberumdrehungen für den Absolutwert unterschieden werden können.
G2_FineResolutionXist1	Geben Sie hier die Feinauflösung von G2_Xist1 an.
G2_FineResolutionXist2	Geben Sie hier die Feinauflösung von G2_Xist2 an.
DSC_Configured	Geben Sie hier an, ob Sie im Telegrammverbund einen DynamicServoControl Baustein verwenden.
Torque_Nominal	Geben Sie hier das Referenzdrehmoment an
Torque_UpperLimit	Geben Sie hier die physikalisch obere Drehmomentgrenze Ihres Antriebs an
Torque_LowerLimit	Geben Sie hier die physikalisch untere Drehmomentgrenze Ihres Antriebs an
Torque_RedScale	Legen Sie hier fest auf welchen Prozentsatz die Momentenreduktion normiert werden soll. Typischerweise entspricht das Referenzdrehmoment 100 %.
DeviceIdentificationCompany	Geben Sie hier die Herstellerkennung ein. Siemens = 42
DeviceIdentificationDeviceType	Geben Sie hier die Gerätekennung ein. Sinamics S120 = 5011
Additional_Parameter	Freitext zum Angeben zusätzlicher Parameter

Zusätzliche Parameter können als Freitext in folgendem Format angegeben werden:

Parameternummer [ParameterSubIndex] = format#value

Zum Beispiel: "180 [1] = u64#10" bedeutet Parameter 180, SubIndex 1 hat den Wert 10 im Datenformat unsigned64.

Die folgenden Datenformate sind definiert:

- b – boolean
- i8 – integer8
- i16 – integer16
- i32 – integer32

- i64 – integer64
- u8 – unsigned8
- u16 – unsigned16
- u32 – unsigned32
- u64 – unsigned64
- f64 – floating point

Die folgende Tabelle zeigt die Parameter, die von der Komponente DataAdaption dem Technologieobjekt zur Verfügung gestellt werden.

Parameter	Parametertext
51	Drive Data Set DDS effective
107	Drive object type / DO type
108	Drive objects function module
180	Number of Drive Data Sets
505	Selecting the system of units / Unit sys select
922	IF1 PROFIdrive PZD telegram selection
924 SubIndex 0	ZSW bit pulses enabled.Signal number
924 SubIndex 1	ZSW bit pulses enabled.Bit position
925	PROFIdrive clock synchronous sign-of-life tolerance
950 SubIndex 0	Number of fault situation
950 SubIndex 1	Number of fault messages in a fault situation
964 SubIndex 0	Device identification.Company
964 SubIndex 1	Device identification.Device type
964 SubIndex 2	Device identification.Firmware version
964 SubIndex 3	Device identification.Firmware date (year)
964 SubIndex 4	Device identification.Firmware date (day/month)
964 SubIndex 5	Device identification.Number of drive objects
964 SubIndex 6	Device identification.Firmware patch/hot fix
965	PROFIdrive profile number
975 SubIndex 1	Drive object identification.Drive object type
975 SubIndex 5	Drive object identification.drive object type class
978 SubIndex 0	List of drive objects
978 SubIndex 1	List of drive objects
978 SubIndex 2	List of drive objects
978 SubIndex 3	List of drive objects
978 SubIndex 4	List of drive objects
978 SubIndex 5	List of drive objects
978 SubIndex 6	List of drive objects
978 SubIndex 7	List of drive objects
978 SubIndex 8	List of drive objects
978 SubIndex 9	List of drive objects
978 SubIndex 10	List of drive objects
978 SubIndex 11	List of drive objects

Parameter	Parametertext
978 SubIndex 12	List of drive objects
978 SubIndex 13	List of drive objects
978 SubIndex 14	List of drive objects
978 SubIndex 15	List of drive objects
978 SubIndex 16	List of drive objects
978 SubIndex 17	List of drive objects
978 SubIndex 18	List of drive objects
978 SubIndex 19	List of drive objects
978 SubIndex 20	List of drive objects
978 SubIndex 21	List of drive objects
978 SubIndex 22	List of drive objects
978 SubIndex 23	List of drive objects
978 SubIndex 24	List of drive objects
979 SubIndex 0	PROFIdrive encoder format [0] = Header
979 SubIndex 1	PROFIdrive encoder format [1] = Type encoder 1
979 SubIndex 2	PROFIdrive encoder format [2] = Resolution enc 1
979 SubIndex 3	PROFIdrive encoder format [3] = Shift factor G1_XIST1
979 SubIndex 4	PROFIdrive encoder format [4] = Shift factor G1_XIST2
979 SubIndex 5	PROFIdrive encoder format [5] = Distinguishable revolutions encoder 1
979 SubIndex 6	PROFIdrive encoder format [6] = Reserved
979 SubIndex 7	PROFIdrive encoder format [7] = Reserved
979 SubIndex 8	PROFIdrive encoder format [8] = Reserved
979 SubIndex 9	PROFIdrive encoder format [9] = Reserved
979 SubIndex 10	PROFIdrive encoder format [10] = Reserved
979 SubIndex 11	PROFIdrive encoder format [11] = Type encoder 2
979 SubIndex 12	PROFIdrive encoder format [12] = Resolution enc 2
979 SubIndex 13	PROFIdrive encoder format [13] = Shift factor G2_XIST1
979 SubIndex 14	PROFIdrive encoder format [14] = Shift factor G2_XIST2
979 SubIndex 15	PROFIdrive encoder format [15] = Distinguishable revolutions encoder 2
979 SubIndex 16	PROFIdrive encoder format [16] = Reserved
979 SubIndex 17	PROFIdrive encoder format [17] = Reserved
979 SubIndex 18	PROFIdrive encoder format [18] = Reserved
979 SubIndex 19	PROFIdrive encoder format [19] = Reserved
979 SubIndex 20	PROFIdrive encoder format [20] = Reserved
979 SubIndex 21	PROFIdrive encoder format [21] = Type encoder 3
979 SubIndex 22	PROFIdrive encoder format [22] = Resolution enc 3
979 SubIndex 23	PROFIdrive encoder format [23] = Shift factor G3_XIST1
979 SubIndex 24	PROFIdrive encoder format [24] = Shift factor G3_XIST2
979 SubIndex 25	PROFIdrive encoder format [25] = Distinguishable revolutions encoder 3
979 SubIndex 26	PROFIdrive encoder format [26] = Reserved
979 SubIndex 27	PROFIdrive encoder format [27] = Reserved

Parameter	Parametertext
979 SubIndex 28	PROFIdrive encoder format [28] = Reserved
979 SubIndex 29	PROFIdrive encoder format [29] = Reserved
979 SubIndex 30	PROFIdrive encoder format [30] = Reserved
1082	Maximum speed / n_max
1520	Torque limit upper/motoring / M_max upper/mot
1521	Torque limit lower/regenerative / M_max lower/regen
1544	Travel to fixed stop evaluation torque reduction / TfS M_red eval
2000	Reference speed reference frequency / n_ref f_ref
2003	Reference torque / M_ref
2079	IF1 PROFIdrive PZD telegram selection extended / IF1 PZD telegr ext

Damit ist eine vollständige Datenadaption möglich.

DynamicServoControl – Nachbildung des Antriebsregleranteils in einem PROFIdrive-Telegramm

Symbol



Funktion

Der Komponententyp *DynamicServoControl* dient zur Anreicherung eines PROFIdrive-Telegramms nach dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) um die Funktionalität des antriebsinternen Reglers.

	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6	PZD 7	PZD 8	PZD 9
TEL_5 Soll	STW 1	NSOLL_B		STW 2	G1_STW	XERR		KPC	
TEL_5 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2	

Im PROFIdrive-Profil sind höherwertige Telegramme definiert worden, mit denen die Positionsregelung von der Steuerung in den Antrieb verlagert werden kann. Dies ist gerade dann entscheidend, wenn der Takt in dem die Technologieanwendung in der Steuerung gegenüber dem Bustakt untersetzt ist.

Die Verlagerung der Positionsregelung von der Steuerung in den Antrieb setzt zwingend ein taktsynchrones Bussystem voraus und dass die SIMIT-Applikation bus-synchron betrieben wird. Stellen Sie dazu die bus-synchrone Betriebsart (Seite 47) ein.

Anschlüsse

Eingang	Beschreibung
XERR	Regeldifferenz aus dem PROFIdrive-Telegramm
KPC	Reglerverstärkung aus dem PROFIdrive-Telegramm
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Ausgang	Beschreibung
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

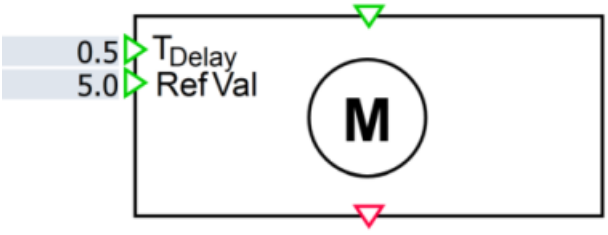
Parameter

Parameter	Beschreibung
G1_FineResolutionXist1	Geben Sie hier die Feinauflösung von G1_Xist1 an
G1_FineResolutionXist2	Geben Sie hier die Feinauflösung von G1_Xist2 an.
G2_FineResolutionXist1	Geben Sie hier die Feinauflösung von G2_Xist1 an
G2_FineResolutionXist2	Geben Sie hier die Feinauflösung von G2_Xist2 an.

Die Parameterwerte sollten konsistent zu den gleichnamigen Parametern des Sensor Bausteins sein.

GeneralDrive

Symbol

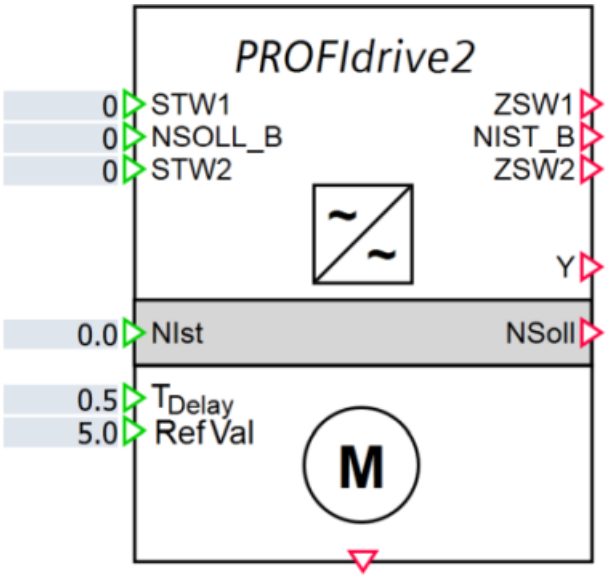


Funktion

Der Komponententyp *GeneralDrive* erweitert den PROFIdrive-Baustein um die Funktionalität eines Verzögerungsgliedes. Die *Istdrehzahlberechnung* erfolgt dabei anhand des vorgegebenen Sollwertes, so dass eine externe Verschaltung des Eingangs NIst am PROFIdrive-Baustein nicht mehr notwendig ist.

T_{Delay} ist die Verzögerungszeit, mit welcher der Drehzahlistwert dem wirksamen Sollwert (Rampenfunktionsgeneratorwert aus PROFIdrive-Baustein) über die Verzögerungsfunktion nachgeführt wird. Die Verzögerungszeit ist mit einer halben Sekunde voreingestellt.

Der Referenzwert am Analogeingang *RefVal* ist ein prozentualer Drehzahlwert. Überschreitet der Drehzahlistwert den Referenzwert, wird Bit 10 im Zustandswort (ZSW1.10) auf 1 gesetzt. Voreingestellt ist ein Referenzwert von 5 Prozent. Das Umschalten dieses Signals erfolgt über eine parametrierbare Hysterese, die ebenfalls als prozentualer Drehzahlwert im Parameter *Hysteresis* angegeben wird. Voreingestellt ist eine Hysterese von 3 Prozent.



Anschlüsse

Eingang	Beschreibung
T_{Delay}	Zeitkonstante der Verzögerungszeit
RefVal	Referenzwert mit dem Bit 10 im Zustandswort (ZSW1.10) auf eins gesetzt
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Ausgang	Beschreibung
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

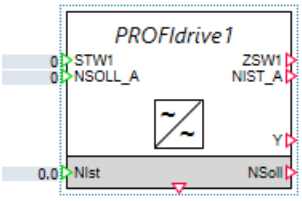
Parameter

Parameter	Beschreibung
ReferenceSpeed	Geben Sie hier die Referenzdrehzahl an.
RampTime	Geben Sie hier die Rampensteilheit für die Änderung des Drehzahlsollwertes an
EmergencyStopTime	Geben Sie hier die Steilheit an, mit der im Fehlerfall der anstehende Drehzahlsollwert abgebaut werden soll.

HysteresisSpeed	Geben Sie hier die Breite des Fensters an, welches um die Solldrehzahl gelegt wird. Ist die Istdrehzahl innerhalb dieses Fensters, wird ZSW1.Bit8 (Speed Error Within Tolerance) auf TRUE gesetzt.
SpeedThreshold	Geben Sie hier einen Vergleichswert für die Drehzahlschwelle an. Hat die Istdrehzahl diesen Vergleichswert erreicht oder überschritten, wird ZSW1.Bit10 (Speed Reached) auf TRUE gesetzt.

PROFIdrive1 – Grundfunktion des PROFIdrive-Antriebs für Drehzahlachsen

Symbol



Funktion

Der Komponententyp *PROFIdrive1* bildet den Zustandsautomaten mit dem Rampenfunktionsgenerator entsprechend dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) nach.

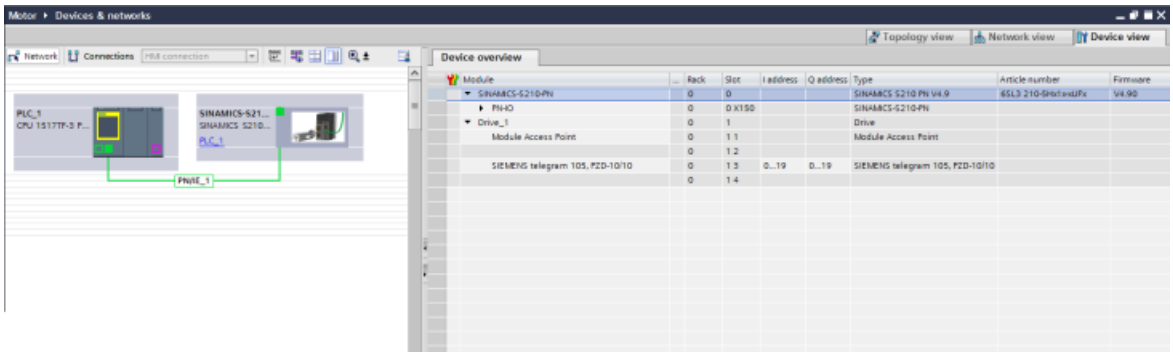
Nachgebildet werden die Funktionalitäten für Application Class 1 (AC1) "Standard Drive – Speed Control Mode".

Diese Komponente bildet damit die im Standard-PROFIdrive-Telegramm 1 definierte Funktionalität nach:

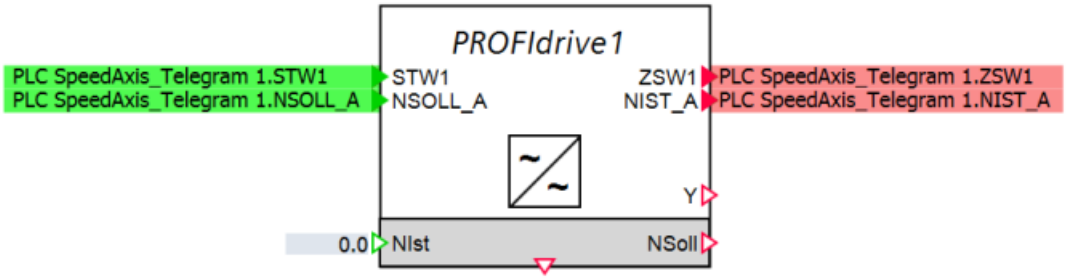
	PZD 1	PZD 2
TEL_1 Soll	STW 1	NSOLL_A
TEL_1 Ist	ZSW 1	NIST_A

Ausgangspunkt für die Konfiguration einer *PROFIdrive1*-Komponente ist der Zugriff der Steuerung auf die Ein-/Ausgangssignale des PROFIdrive-Antriebs. Für jeden Antrieb wird ein fester Adressbereich im Ein-/Ausgangsbereich der Steuerung festgelegt und einer PROFIdrive1-Komponente wie folgt zugeordnet: Das Steuerwort (erstes Prozessdatenwort im Ausgangsbereich) wird mit dem ganzzahligen (Integer) Eingang *STW1* verbunden und das Zustandswort (erstes Prozessdatenwort im Eingangsbereich) wird mit dem ganzzahligen (Integer) Ausgang *ZSW1* verbunden. Im zweiten Prozessdatenwort werden die Soll- und Istwerte für die Drehzahl übertragen. Entsprechend erfolgt die Verbindung des ganzzahligen (Integer) Eingangs *NSOLL_A* mit dem Drehzahlsollwert. Der ganzzahlige (Integer) Ausgang *NIST_A* wird mit dem Drehzahlistwertsignal der Steuerung verbunden.

In der folgenden Abbildung ist die SIMATIC-Projektierung beispielhaft für SINAMICS dargestellt.



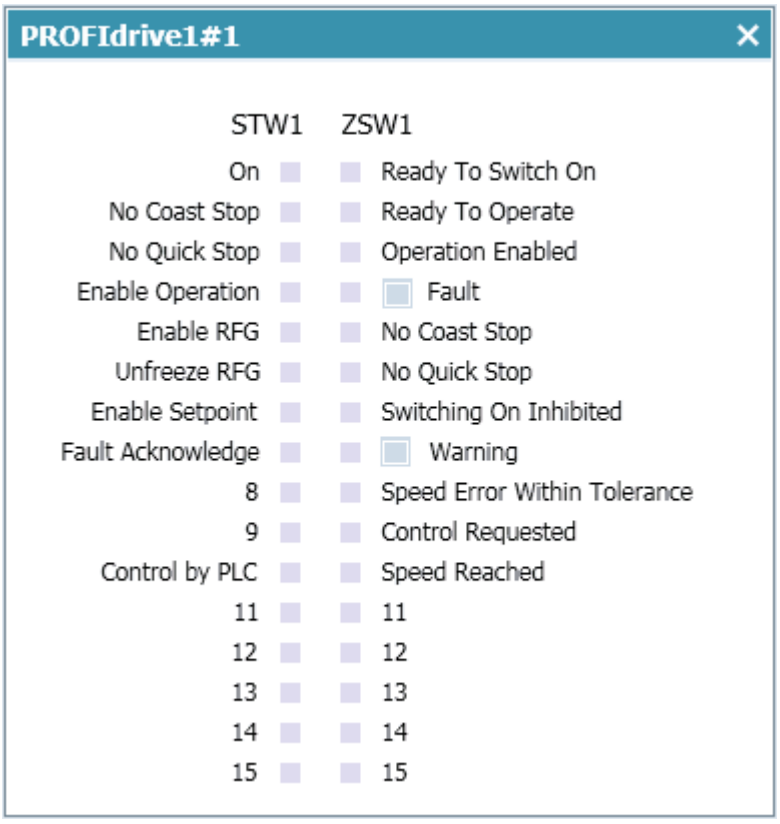
Die folgende Abbildung zeigt die *PROFIdrive1*-Komponente mit den verbundenen Prozessdaten:



Die Drehzahlwerte der Prozessdaten, also *NSOLL_A* und *NIST_A*, sind Rohwerte im Bereich null bis 16384. Der Wert 16384 entspricht der Nenndrehzahl. Weitere Prozessdaten, z. B. Momenten- oder Stromistwerte werden in der *PROFIdrive1*-Komponente nicht berücksichtigt.

Am Analogausgang *Y* wird der aktuelle Drehzahlwert des Antriebs als Prozentwert, d. h. im Bereich 0 bis 100 ausgegeben: $0 \leq Y \leq 100$. Der Wert 0 entspricht damit dem stillstehenden Antrieb, der Wert 100 der Nenndrehzahl (Drehzahlrohwert 16384).

Im Bedienfenster der Komponente werden das Steuerwort und das Zustandswort bitweise angezeigt. Zusätzlich werden der Drehzahlsollwert *NSOLL_A* und der Drehzahlistwert *NIST_A* als Prozentwerte und als Werte in Umdrehungen pro Minute (*RPM*) angezeigt. Die vierte und achte Bit im Zustandswort (ZSW1.3 – *Fault* und ZSW1.7 – *Warning*) können über einen Umschalter auf 1 gesetzt werden.

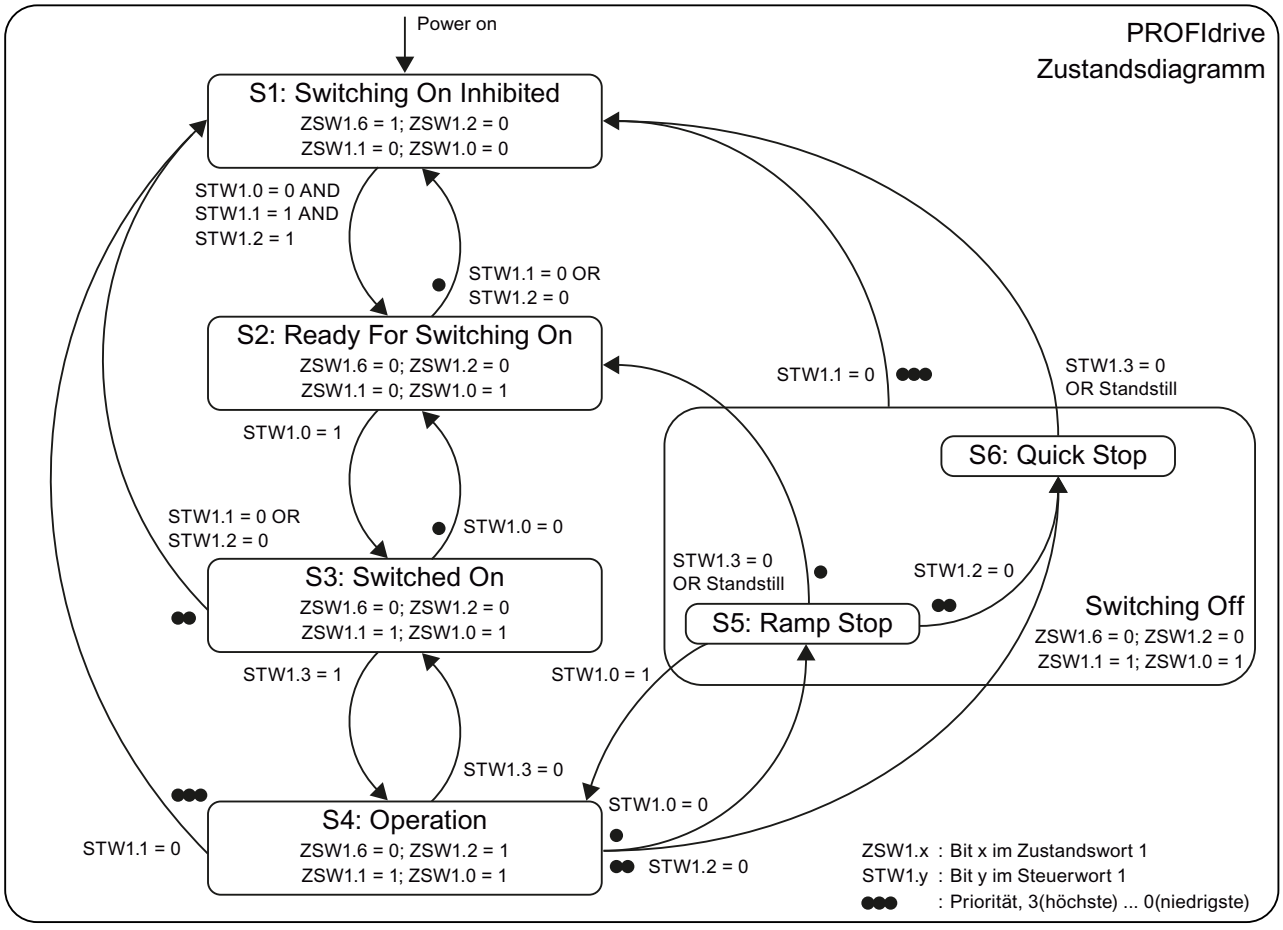


Siehe auch

Verwendung der PROFIdrive-Bibliothek zur Simulation von PROFIdrive-Antrieben (Seite 487)

Zustandsautomat

Der Zustandsautomat ist im Detail in der PROFIdrive-Spezifikation beschrieben. Der im Komponententyp *PROFIdrive1* implementierte Zustandsgraph ist in der folgenden Abbildung dargestellt:



Zur Steuerung des Zustandsautomaten (Zustandsübergänge) werden die in der Tabelle aufgeschlüsselten Steuerbits verwendet. Abhängig vom aktuellen Zustand werden die entsprechenden Zustandsbits wie in der Tabelle dargestellt gesetzt.

Tabelle 8-8 Zustandstabelle der Komponente PROFIdrive1

Zustand	State	Beschreibung	ZSW1 (Bitfolge 15 … 0)
S1: Switching On Inhibited	1	Aus	dddd dmsm a1ss a000
S2: Ready For Switching On	2	Bereit	dddd dmsm a0ss a001
S3: Switched On	3	Eingeschaltet	dddd dmsm a0ss a011
S4: Operation	4	Normaler Betrieb	dddd dmsm a0ss a111
S5: Switching Off (Ramp Stop)	5	Rampenhalt des Motors	dddd dmsm a0ss a011
S6: Switching Off (Quick Stop)	6	Schnellhalt des Motors	dddd dmsm a0ss a011

Die einzelnen Stellen im Zustandswort *ZSW1* in der Tabelle haben dabei die folgende Bedeutung:

d gerätespezifisch (Rumpfkomponente)
s direkt aus *STW1* abgeleitet
m modellabhängig (hier: Verzögerung)
a im Bedienfenster setzbar

Tabelle 8-9 Aufschlüsselung des Steuerworts

Name	STW1.Bit	Beschreibung	Verwendung
On/Off (Off1)	STW1.0	Antrieb ausschalten	Ja
No Coast Stop (Off2)	STW1.1	Kein Austrudeln des Antriebs	Ja
No Quick Stop (Off3)	STW1.2	Kein Schnellhalt des Antriebs	Ja
Enable Operation	STW1.3	Antrieb fährt auf Setpoint	Ja
Enable Ramp Generator	STW1.4	RFG wird verwendet	Ja
Unfreeze Ramp Generator	STW1.5	RFG eingefroren	Ja
Enable Setpoint	STW1.6	NSOLL als Eingang für RFG	Ja
Fault Acknowledge	STW1.7	Fehlerbestätigung der SPS	Ja
Jog 1 On	STW1.8	nicht implementiert	Nein
Jog 2 On	STW1.9	nicht implementiert	Nein
Control by PLC	STW1.10	DO IO Data gültig	Ja
Device-specific	STW1.11–15		Rumpfkomponente

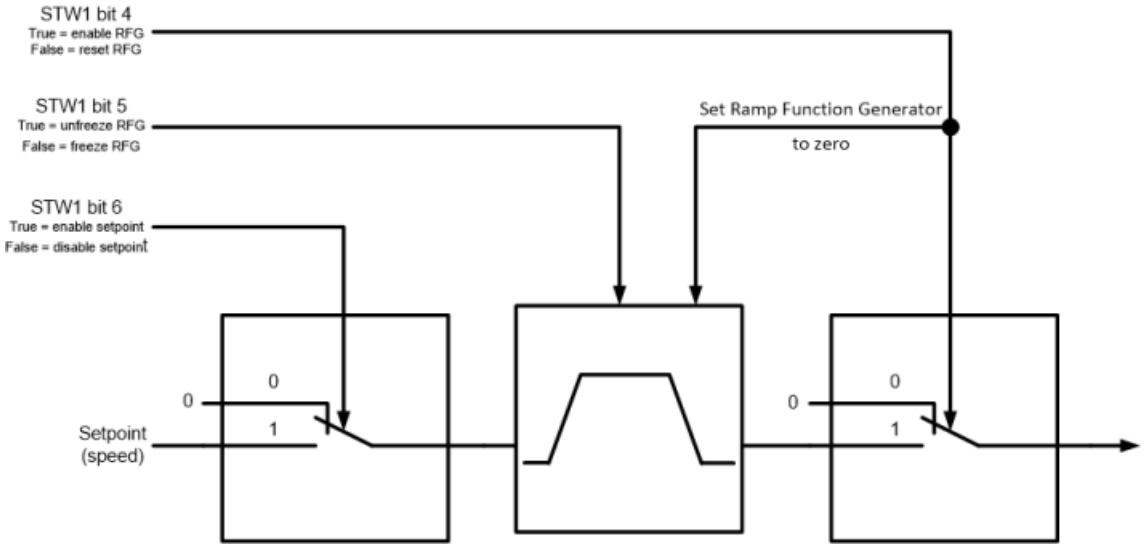
Die Bedeutung aller Bits des Zustandswortes ist in der folgenden Tabelle beschrieben.

Tabelle 8-10 Aufschlüsselung des Zustandsworts

Name	ZSW1.Bit	Beschreibung	Quelle
Ready To Switch On	ZSW1.0	Stromversorgung eingeschaltet	Zustandsautomat
Ready To Operate	ZSW1.1	keine Störung liegt vor	Zustandsautomat
Operation Enabled	ZSW1.2	Betrieb freigegeben	Zustandsautomat
Fault Present	ZSW1.3	Antrieb gestört/Außer Betrieb	Bedienfenster
Coast Stop Not Active	ZSW1.4	Kein Austrudeln des Antriebs	STW1.1
Quick Stop Not Active	ZSW1.5	Kein Schnellhalt des Antriebs	STW1.2
Switching On Inhibited	ZSW1.6	Einschaltsperre	Zustandsautomat
Warning Present	ZSW1.7	Warnung liegt vor/Außer Betrieb	Bedienfenster
Speed Error Within Tolerance	ZSW1.8	Soll-/Istwertabweichung im Toleranzbereich	PT1
Control Requested	ZSW1.9	Führung gefordert/Betrieb vor Ort	STW1.10
Speed Reached	ZSW1.10	Referenzgeschwindigkeit erreicht	PT1
Device-specific	ZSW1.11–15	Antriebsspezifisch	Zusatzkomponente/0

Rampengenerator

In Abhängigkeit der Bits 4, 5 und 6 des Steuerworts (STW1.4, STW1.5, STW1.6) wird der Rampenfunktionsgenerator (RFG) gesteuert. Bit 6 setzt den Eingang des Generators auf *NSOLL_RFG*. Daraufhin verfährt die Rampe linear mit der eingestellten Zeit *RampTime* auf diesen Sollwert, falls der Generator über Bit 5 freigegeben ist. Ansonsten wird der letzte Ausgangswert des Generators gehalten. Abschließend wird durch Bit 4 bestimmt, ob der Rampengeneratorwert ausgegeben wird. Eine Sollwertaufschaltung kann nur im Zustand S4 erfolgen. Sonst wird der Rampensollwert 0 ausgegeben.



Spezifischer Ausgang zur Erweiterung

Ein spezifischer Anschluss Drive an der Unterseite der Komponente dient zur Verbindung mit einer Rumpfkomponente oder der Erweiterung um weitere Funktionen.

Dieser Anschluss ist vom Verbindungstyp PROFIdrive. Er kann nicht mit analogen, binären oder ganzzahligen (Integer) Anschlüssen von Komponenten verbunden werden.

Verwendung des PROFIdrive1-Komponententyps ohne Erweiterung

Der Komponententyp *PROFIdrive1* realisiert die Standardfunktionen nach dem PROFIdrive-Profil. Die *PROFIdrive1*-Komponente kann daher ohne Verschaltung mit einer Rumpfkomponente zur Simulation von PROFIdrive-Antrieben mit dieser Grundfunktion verwendet werden. Antriebsspezifische Bits des Steuerwortes werden dann ignoriert und antriebsspezifische Bits des Zustandswortes werden auf null gesetzt.

Anschlüsse

Eingang	Beschreibung
STW1	Steuerwort 1 (STW1) aus dem PROFIdrive-Telegramm

NSOLL_A	Normierter Drehzahlsollwert (NSOLL_A) aus dem PROFIdrive-Telegramm
NIst	Prozesswert aktuelle Drehzahl

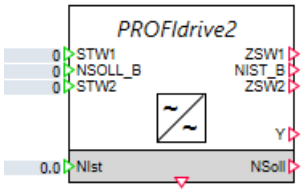
Ausgang	Beschreibung
ZSW1	Zustandswort 1 (ZSW1) aus dem PROFIdrive-Telegramm
NIST_A	Normierter Drehzahlistwert (NIST_A) aus dem PROFIdrive-Telegramm
Y	Prozesswert Drehzahlsollwertvorgabe bezogen auf die Referenzdrehzahl in %
NSoll	Prozesswert Drehzahlsollwertvorgabe
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
ReferenceSpeed	Geben Sie hier die Referenzdrehzahl an.
MaxSpeed	Geben Sie hier die physikalisch maximale Drehzahl des Antriebs an
RampTime	Geben Sie hier die Rampensteilheit für die Änderung des Drehzahlsollwertes an
EmergencyStopTime	Geben Sie hier die Steilheit an, mit der im Fehlerfall der anstehende Drehzahlsollwert abgebaut werden soll.
HysteresisSpeed	Geben Sie hier die Breite des Fensters an, welche um die Solldrehzahl gelegt wird. Ist die Istdrehzahl innerhalb dieses Fensters, wird ZSW.Bit8 (Speed Error Within Tolerance) auf TRUE gesetzt.
SpeedThreshold	Geben Sie hier einen Vergleichswert für die Drehzahlschwelle an. Hat die Istdrehzahl diesen Vergleichswert erreicht oder überschritten, wird ZSW.Bit10 (Speed Reached) auf TRUE gesetzt.

PROFIdrive2 – Grundfunktion des PROFIdrive-Antriebs für Positionsachsen

Symbol



Funktion

Der Komponententyp PROFIdrive2 bildet den Zustandsautomaten mit dem Rampenfunktionsgenerator entsprechend dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) nach.

Ergänzend zum PROFIdrive1-Baustein, stellt der PROFIdrive2-Baustein eine Überwachungsmöglichkeit in Form eines Lebenszeichensignalaustausches im STW2 und ZSW2 zur Verfügung.

Der zyklische Austausch des Lebenszeichens setzt voraus, dass die SIMIT-Applikation bus-synchron betrieben wird. Stellen Sie dazu die bus-synchrone Betriebsart ein.

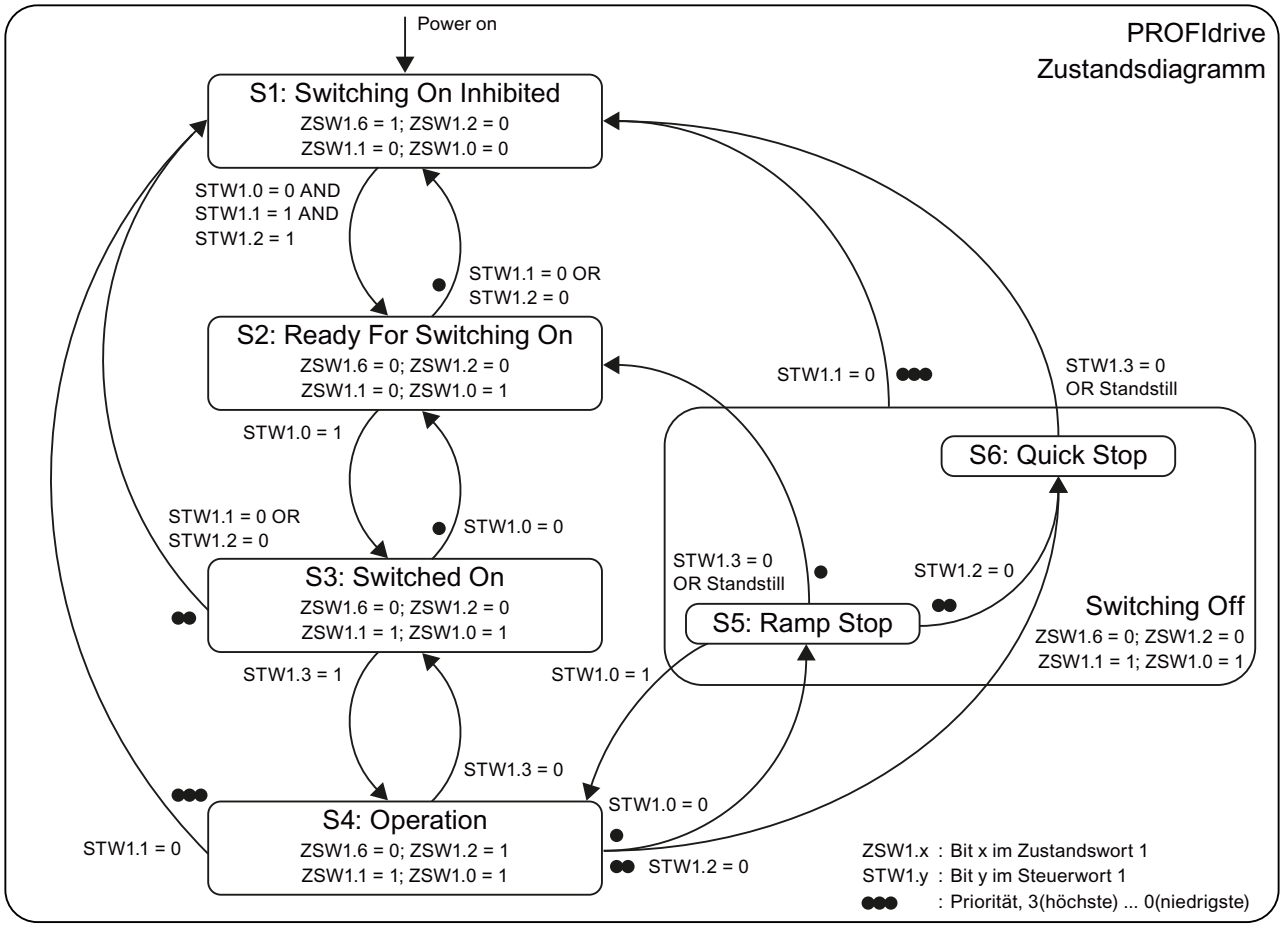
Diese Komponente bildet damit die im Standard-PROFIdrive-Telegramm 2 definierte Funktionalität nach, mit der erweitere Drehzahlachsen betrieben werden können:

	PZD 1	PZD 2	PZD 3	PZD 4
TEL_2 Soll	STW 1	NSOLL_B		STW 2
TEL_2 Ist	ZSW 1	NIST_B		ZSW 2

Zur Modellierung von Positionsachsen werden weitere Komponenten benötigt, die über den speziellen Anschluss zur Erweiterung angedockt werden können (siehe auch Spezifischer Ausgang zur Erweiterung (Seite 508)).

Zustandsautomat

Der Zustandsautomat ist im Detail in der PROFIdrive-Spezifikation beschrieben. Der im Komponententyp *PROFIdrive2* implementierte Zustandsgraph ist in der folgenden Abbildung dargestellt:



Zur Steuerung des Zustandsautomaten (Zustandsübergänge) werden die in der Tabelle aufgeschlüsselten Steuerbits verwendet. Abhängig vom aktuellen Zustand werden die entsprechenden Zustandsbits wie in der Tabelle dargestellt gesetzt.

Tabelle 8-11 Zustandstabelle der Komponente PROFIdrive2

Zustand	State	Beschreibung	ZSW1 (Bitfolge 15 … 0)
S1: Switching On Inhibited	1	Aus	dddd dmsm a1ss a000
S2: Ready For Switching On	2	Bereit	dddd dmsm a0ss a001
S3: Switched On	3	Eingeschaltet	dddd dmsm a0ss a011
S4: Operation	4	Normaler Betrieb	dddd dmsm a0ss a111
S5: Switching Off (Ramp Stop)	5	Rampenhalt des Motors	dddd dmsm a0ss a011
S6: Switching Off (Quick Stop)	6	Schnellhalt des Motors	dddd dmsm a0ss a011

Die einzelnen Stellen im Zustandswort *ZSW1* in der Tabelle haben dabei die folgende Bedeutung:

d gerätespezifisch (Rumpfkomponente)
s direkt aus *STW1* abgeleitet
m modellabhängig (hier: Verzögerung)
a im Bedienfenster setzbar

Tabelle 8-12 Aufschlüsselung des Steuerworts

Name	STW1.Bit	Beschreibung	Verwendung
On/Off (Off1)	STW1.0	Antrieb ausschalten	Ja
No Coast Stop (Off2)	STW1.1	Kein Austrudeln des Antriebs	Ja
No Quick Stop (Off3)	STW1.2	Kein Schnellhalt des Antriebs	Ja
Enable Operation	STW1.3	Antrieb fährt auf Setpoint	Ja
Enable Ramp Generator	STW1.4	RFG wird verwendet	Ja
Unfreeze Ramp Generator	STW1.5	RFG eingefroren	Ja
Enable Setpoint	STW1.6	NSOLL als Eingang für RFG	Ja
Fault Acknowledge	STW1.7	Fehlerbestätigung der SPS	Ja
Jog 1 On	STW1.8	nicht implementiert	Nein
Jog 2 On	STW1.9	nicht implementiert	Nein
Control by PLC	STW1.10	DO IO Data gültig	Ja
Device-specific	STW1.11–15		Rumpfkomponente

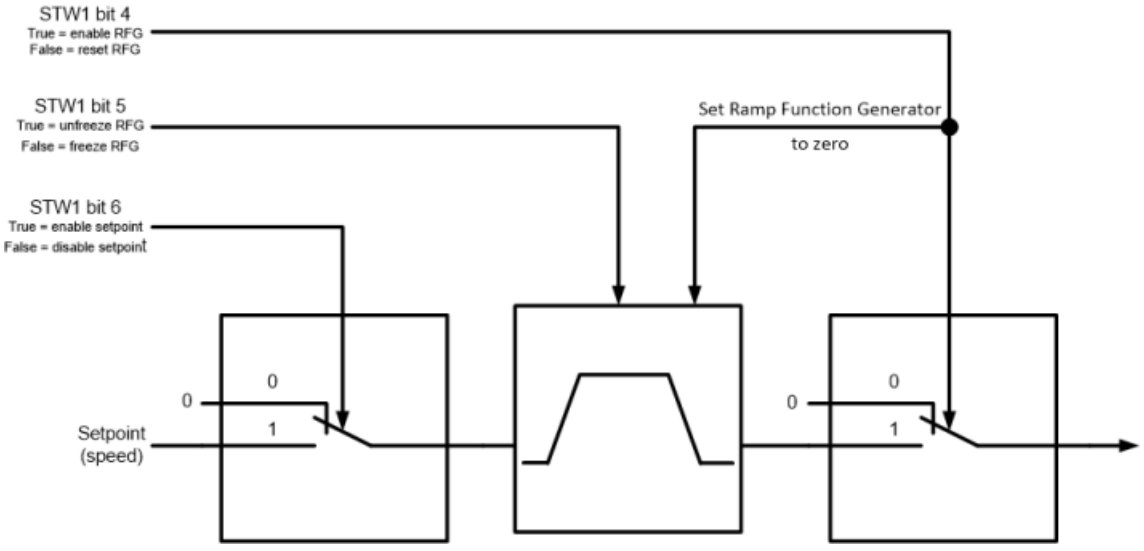
Die Bedeutung aller Bits des Zustandswortes ist in der folgenden Tabelle beschrieben.

Tabelle 8-13 Aufschlüsselung des Zustandsworts

Name	ZSW1.Bit	Beschreibung	Quelle
Ready To Switch On	ZSW1.0	Stromversorgung eingeschaltet	Zustandsautomat
Ready To Operate	ZSW1.1	keine Störung liegt vor	Zustandsautomat
Operation Enabled	ZSW1.2	Betrieb freigegeben	Zustandsautomat
Fault Present	ZSW1.3	Antrieb gestört/Außer Betrieb	Bedienfenster
Coast Stop Not Active	ZSW1.4	Kein Austrudeln des Antriebs	STW1.1
Quick Stop Not Active	ZSW1.5	Kein Schnellhalt des Antriebs	STW1.2
Switching On Inhibited	ZSW1.6	Einschaltsperre	Zustandsautomat
Warning Present	ZSW1.7	Warnung liegt vor/Außer Betrieb	Bedienfenster
Speed Error Within Tolerance	ZSW1.8	Soll-/Istwertabweichung im Toleranzbereich	PT1
Control Requested	ZSW1.9	Führung gefordert/Betrieb vor Ort	STW1.10
Speed Reached	ZSW1.10	Referenzgeschwindigkeit erreicht	PT1
Device-specific	ZSW1.11–15	Antriebsspezifisch	Zusatzkomponente/0

Rampengenerator

In Abhängigkeit der Bits 4, 5 und 6 des Steuerworts (STW1.4, STW1.5, STW1.6) wird der Rampenfunktionsgenerator (RFG) gesteuert. Bit 6 setzt den Eingang des Generators auf *NSOLL_RFG*. Daraufhin verfährt die Rampe linear mit der eingestellten Zeit *RampTime* auf diesen Sollwert, falls der Generator über Bit 5 freigegeben ist. Sonst wird der letzte Ausgangswert des Generators gehalten. Abschließend wird durch Bit 4 bestimmt, ob der Rampengeneratorwert ausgegeben wird. Eine Sollwertaufschaltung kann nur im Zustand S4 erfolgen. Sonst wird der Rampensollwert 0 ausgegeben.



Lebenszeichenaustausch über STW2 und ZSW2

In den Standard-PROFIdrive-Telegrammen 2 und höher sind die Prozessdaten STW2 und ZSW2 definiert. Über diese wird ein Lebenszeichen zwischen Steuerung und Antrieb zyklisch ausgetauscht. Fällt das Lebenszeichen in einer parametrierbaren Folge aus, wird ein Fehler erzeugt und es erfolgt eine Sicherheitsabschaltung des Antriebs.

Spezifischer Ausgang zur Erweiterung

Ein spezifischer Anschluss Drive an der Unterseite der Komponente dient zur Verbindung mit einer Rumpfkomponente oder der Erweiterung um weitere Funktionen.

Dieser Anschluss ist vom Verbindungstyp PROFIdrive. Er kann nicht mit analogen, binären oder ganzzahligen (Integer) Anschlüssen von Komponenten verbunden werden.

Verwendung des PROFIdrive2-Komponententyps ohne Erweiterung

Der Komponententyp *PROFIdrive2* realisiert die Standardfunktionen nach dem PROFIdrive-Profil. Die *PROFIdrive2*-Komponente kann daher ohne Verschaltung mit einer Rumpfkomponente zur Simulation von PROFIdrive-Antrieben mit dieser Grundfunktion verwendet werden. Antriebsspezifische Bits des Steuerwortes werden dann ignoriert und antriebsspezifische Bits des Zustandswortes werden auf null gesetzt.

Sollen höherwertige PROFIdrive-Telegramme (3, 4, 5, 6, 102, 103, 105, 106) modelliert werden, muss der Grundbaustein zwingend entsprechend erweitert werden.

Anschlüsse

Eingang	Beschreibung
STW1	Steuerwort 1 (STW1) aus dem PROFIdrive-Telegramm
NSOLL_B	Normierter Drehzahlsollwert (NSOLL_B) aus dem PROFIdrive-Telegramm
STW2	Steuerwort 2 (STW2) aus dem PROFIdrive-Telegramm
NIst	Prozesswert aktuelle Drehzahl

Ausgang	Beschreibung
ZSW1	Zustandswort 1 (ZSW1) aus dem PROFIdrive-Telegramm
NIST_B	Normierter Drehzahlistwert (NIST_B) aus dem PROFIdrive-Telegramm
ZSW2	Zustandswort 2 (ZSW2) aus dem PROFIdrive-Telegramm
Y	Prozesswert Drehzahlsollwertvorgabe bezogen auf die Referenzdrehzahl in %
NSoll	Prozesswert Drehzahlsollwertvorgabe
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
ReferenceSpeed	Geben Sie hier die Referenzdrehzahl an.
MaxSpeed	Geben Sie hier die physikalisch maximale Drehzahl des Antriebs an
RampTime	Geben Sie hier die Rampensteilheit für die Änderung des Drehzahlsollwertes an
EmergencyStopTime	Geben Sie hier die Steilheit an, mit der im Fehlerfall der anstehende Drehzahlsollwert abgebaut werden soll.
AllowedLifeSignErrors	Geben Sie hier die Anzahl der tolerierten Lebenszeichenausfälle an. (0 bedeutet, dass kein einziges Lebenszeichen ausfallen darf)
HysteresisSpeed	Geben Sie hier die Breite des Fensters an, welches um die Solldrehzahl gelegt wird. Ist die Istdrehzahl innerhalb dieses Fensters, wird ZSW1.Bit8 (Speed Error Within Tolerance) auf TRUE gesetzt.
SpeedThreshold	Geben Sie hier einen Vergleichswert für die Drehzahlschwelle an. Hat die Istdrehzhal diesen Vergleichswert erreicht oder überschritten, wird ZSW1.Bit10 (Speed Reached) auf TRUE gesetzt.

Safety30 – Anreicherung der Antriebsnachbildung mit einem PROFISafe-Telegramm 30

Symbol



Funktion

Der Komponententyp *Safety30* dient zur Anreicherung eines PROFIdrive-Telegramms nach dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) um die Funktionalität eines Safety-Telegramms 30.

Dieser Baustein bearbeitet die Nutzdaten des Safety-Telegramms 30. Die Simulation des PROFISafe-Trailers wird nicht durchgeführt. Für den Einsatz von PLCSIM_Adv ist eine Simulation der sicheren PROFISafe-Kommunikation nicht notwendig.

Anschlüsse

Eingang	Beschreibung
S_STW1	Safety Steuerwort 1
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Ausgang	Beschreibung
S_ZSW1	Safety Zustandswort 1
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
SafetyTimeout	Überwachungszeit einer aktiven Safetyfunktion in ms
STO_Enabled	Unterstützung der Safety Integrated Function Safe Torque Off (STO)
SS1_Enabled	Unterstützung der Safety Integrated Function Safe Stop 1 (SS1)
SS2_Enabled	Unterstützung der Safety Integrated Function Safe Stop 2 (SS2)
SOS_Enabled	Unterstützung der Safety Integrated Function Safe Operating Stop (SOS)
SLS_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Speed (SLS)
SLT_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Torque (SLT)
SLP_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Position (SLP)

Parameter	Beschreibung
SLA_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Acceleration (SLA)
SDI_Enabled	Unterstützung der Safety Integrated Function Safely – Safe Direction (SDI)

Safety31 – Anreicherung der Antriebsnachbildung mit einem PROFISafe-Telegramm 31

Symbol



Funktion

Der Komponententyp *Safety31* dient zur Anreicherung eines PROFIdrive-Telegramms nach dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) um die Funktionalität eines Safety-Telegramms 31.

Dieser Baustein bearbeitet die Nutzdaten des Safety-Telegramms 31. Die Simulation des PROFISafe-Trailers wird nicht durchgeführt. Für den Einsatz von PLCSIM_Adv ist eine Simulation der sicheren PROFISafe-Kommunikation nicht notwendig.

Anschlüsse

Eingang	Beschreibung
S_STW2	Safety Steuerwort 2
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Ausgang	Beschreibung
S_ZSW2	Safety Zustandswort 2
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
SafetyTimeout	Überwachungszeit einer aktiven Safetyfunktion in ms
STO_Enabled	Unterstützung der Safety Integrated Function Safe Torque Off (STO)
SS1_Enabled	Unterstützung der Safety Integrated Function Safe Stop 1 (SS1)
SS2_Enabled	Unterstützung der Safety Integrated Function Safe Stop 2 (SS2)
SOS_Enabled	Unterstützung der Safety Integrated Function Safe Operating Stop (SOS)
SLS_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Speed (SLS)

Parameter	Beschreibung
SLT_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Torque (SLT)
SLP_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Position (SLP)
SLA_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Acceleration (SLA)
SDI_Enabled	Unterstützung der Safety Integrated Function Safely – Safe Direction (SDI)

Safety32 – Anreicherung der Antriebsnachbildung mit einem PROFISafe-Telegramm 32

Symbol



Funktion

Der Komponententyp *Safety32* dient zur Anreicherung eines PROFIdrive-Telegramms nach dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) um die Funktionalität eines Safety-Telegramms 32.

Dieser Baustein bearbeitet die Nutzdaten des Safety-Telegramms 32. Die Simulation des PROFISafe-Trailers wird nicht durchgeführt. Für den Einsatz von PLCSIM_Adv ist eine Simulation der sicheren PROFISafe-Kommunikation nicht notwendig.

Anschlüsse

Eingang	Beschreibung
S_STW2	Safety Steuerwort 2
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Ausgang	Beschreibung
S_ZSW2	Safety Zustandswort 2
S_XIST32	Safety position aktueller Wert (für Safety-Telegramm 32)
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

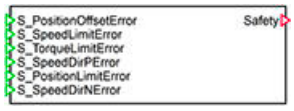
Parameter

Parameter	Beschreibung
SafetyTimeout	Überwachungszeit einer aktiven Safetyfunktion in ms
STO_Enabled	Unterstützung der Safety Integrated Function Safe Torque Off (STO)
SS1_Enabled	Unterstützung der Safety Integrated Function Safe Stop 1 (SS1)

Parameter	Beschreibung
SS2_Enabled	Unterstützung der Safety Integrated Function Safe Stop 2 (SS2)
SOS_Enabled	Unterstützung der Safety Integrated Function Safe Operating Stop (SOS)
SLS_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Speed (SLS)
SLT_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Torque (SLT)
SLP_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Position (SLP)
SLA_Enabled	Unterstützung der Safety Integrated Function Safely-Limited Acceleration (SLA)
SDI_Enabled	Unterstützung der Safety Integrated Function Safely – Safe Direction (SDI)
G1_EncoderSystem	Geben Sie hier die Art des Gebersystems an. 0 = rotatorisches Gebersystem 1 = lineares Gebersystem

SafetyProcess – Auslösen von Fehlern während aktiver Safety-Funktionen

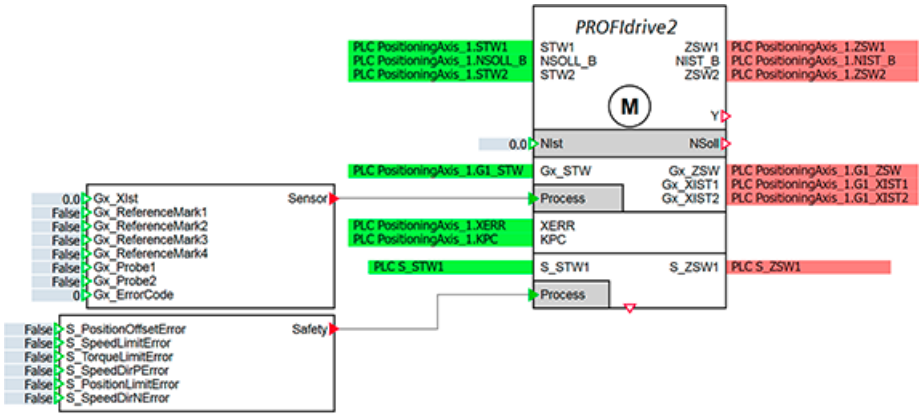
Symbol



Funktion

Der Komponententyp *SafetyProcess* dient zum Auslösen von Fehlern während aktiver antriebsinternen Safety-Funktionen.

Die Komponente muss zwingend mit dem Anschluss Process einer Komponente Safety aus dem PROFIdrive-Telegramm verbunden werden.



Anschlüsse

Eingang	Beschreibung
S_PositionOffsetError	Auslösen eines Positionsüberwachungsfehler während einer aktiven Safety Integrated Function: Safe Operating Stop (SOS)
S_SpeedLimitError	Auslösen eines Geschwindigkeitsüberwachungsfehlers während einer aktiven Safety Integrated Function: Safely-Limited Speed (SLS)
S_TorqueLimitError	Auslösen eines Momentengrenzenverletzung während einer aktiven Safety Integrated Function: Safely-Limited Torque (SLT)
S_SpeedDirPError	Auslösen einer Richtungsumkehr (negative) während einer aktiven Safety Integrated Function: Safe Direction (SDI)
S_PositionLimitError	Auslösen einer Lagebegrenzungsüberschreitung während einer aktiven Safety Integrated Function: Safely Limited Position (SLP)
S_SpeedDirNError	Auslösen einer Richtungsumkehr (positive) während einer aktiven Safety Integrated Function: Safe Direction (SDI)

Ausgang	Beschreibung
Safety	Ausgangsstruktur zur Verschaltung auf den Anschluss Process einer Komponente Safety aus dem PROFIdrive-Telegramm.

Sensor – Nachbildung des Sensoranteils in einem PROFIdrive-Telegramm

Symbol



Funktion

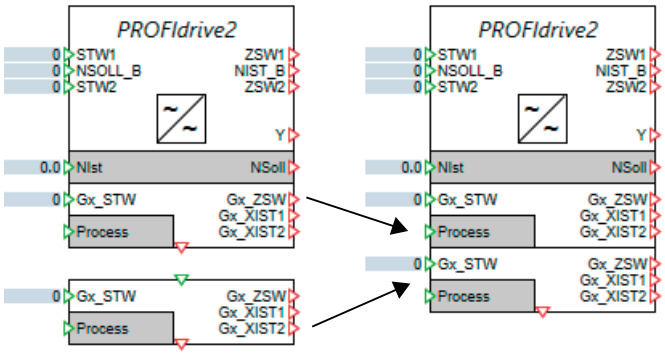
Der Komponententyp *Sensor* dient zur Anreicherung eines PROFIdrive-Telegramms nach dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) um die Funktionalität eines Gebers.

Diese Komponente bildet damit den Anteil eines Gebers im Standard-PROFIdrive-Telegramm nach:

	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6	PZD 7	PZD 8	PZD 9
TEL_3 Soll	STW 1	NSOLL_B		STW 2	G1_STW				
TEL_3 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2	

Soll ein Telegramm mit zwei Gebern nachgebildet werden, kann dies über das Andocken einer weiteren Komponente "Sensor" erfolgen.

	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6	PZD 7	PZD 8	PZD 9	PZD 10	PZD 11	PZD 12	PZD 13	PZD 14
TEL_4 Soll	STW 1	NSOLL_B		STW 2	G1_STW	G2_STW								
TEL_4 Ist	ZSW 1	NIST_B		ZSW 2	G1_ZSW	G1_XIST 1		G1_XIST 2		G2_ZSW	G2_XIST 1		G2_XIST 2	



Anschlüsse

Eingang	Beschreibung
Gx_STW	Gebersteuerwort (Gx_STW) aus dem PROFIdrive-Telegramm
Process	Struktur zum Andocken einer Gebernachbildung
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

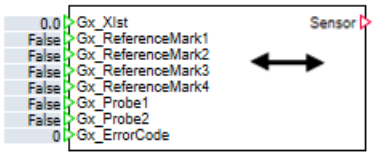
Ausgang	Beschreibung
Gx_ZSW	Geberzustandswort (Gx_ZSW) aus dem PROFIdrive-Telegramm
Gx_XIST1	Geberistwert 1 (Gx_XIST1) aus dem PROFIdrive-Telegramm
Gx_XIST2	Geberistwert 2 (Gx_XIST2) aus dem PROFIdrive-Telegramm
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
GxFineResolutionXist1	Geben Sie hier die Feinauflösung von Gx_Xist1 an
GxFineResolutionXist2	Geben Sie hier die Feinauflösung von Gx_Xist2 an.

SensorProcessLinear – Nachbildung eines linearen Gebers

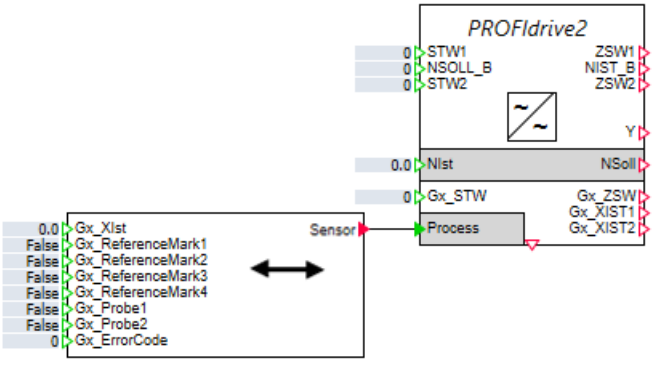
Symbol



Funktion

Der Komponententyp *SensorProcessLinear* dient zur Nachbildung der Funktionalität eines linearen Gebers. Dabei können sowohl linear absolute als auch linear inkrementelle Geber nachgebildet werden.

Die Komponente muss zwingend mit dem Anschluss "Process" einer Komponente "Sensor" (Seite 514) aus dem PROFIdrive-Telegramm verbunden werden.



Anschlüsse

Eingang	Beschreibung
Gx_XIST	Prozesswert in der physikalischen Einheit mm
Gx_ReferenceMark1…4	Geberreferenzmarken 1–4 (TRUE = Referenzmarke ausgelenkt)
Gx_Probe1..2	Messtasterauslenkung 1–2 (TRUE = Messtaster ausgelenkt)
Gx_ErrorCode	Simulierter Geberfehlercode

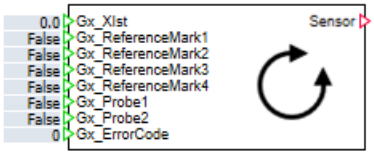
Ausgang	Beschreibung
Sensor	Ausgangsstruktur zur Verschaltung auf den Anschluss "Process" einer Komponente "Sensor" (Seite 514) aus dem PROFIdrive-Telegramm

Parameter

Parameter	Beschreibung
DistanceBetweenIncrements	Geben Sie hier den Abstand zwischen zwei Teilstrichen in der Einheit Millimeter an.
SupportsAbsoluteValues	Geben Sie hier an, ob ein absoluter (=TRUE) oder inkrementeller (=FALSE) Geber nachgebildet werden soll.

SensorProcessRotatory – Nachbildung eines rotatorischen Gebers

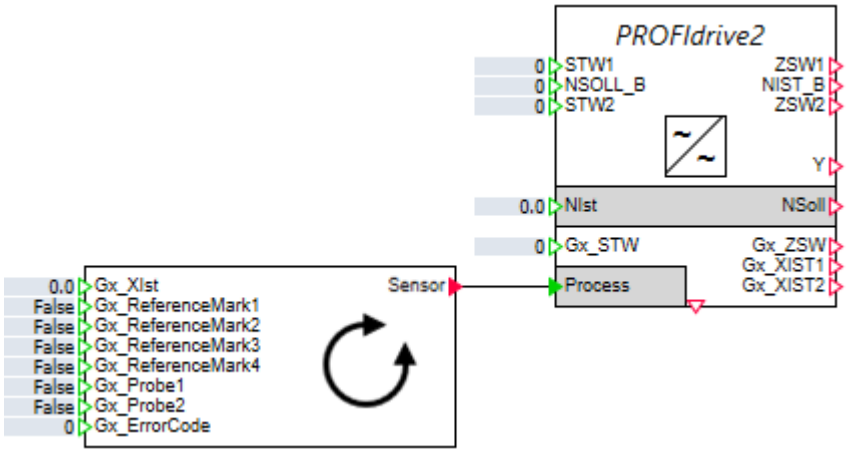
Symbol



Funktion

Der Komponententyp *SensorProcessRotatory* dient zur Nachbildung der Funktionalität eines rotatorischen Gebers. Dabei können sowohl rotatorisch absolute als auch rotatorisch inkrementelle Geber nachgebildet werden.

Die Komponente muss zwingend mit dem Anschluss "Process" einer Komponente "Sensor" (Seite 514) aus dem PROFIdrive-Telegramm verbunden werden.



Anschlüsse

Eingang	Beschreibung
Gx_XIST	Prozesswert in einer physikalischen Größe
Gx_ReferenceMark1…4	Geberreferenzmarken 1–4 (TRUE = Referenzmarke ausgelenkt)

Eingang	Beschreibung
Gx_Probe1..2	Messtasterauslenkung 1–2 (TRUE = Messtaster ausgelenkt)
Gx_ErrorCode	Simulierter Geberfehlercode

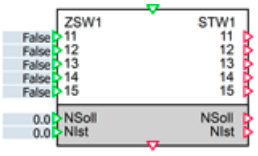
Ausgang	Beschreibung
Sensor	Ausgangsstruktur zur Verschaltung auf den Anschluss "Process" einer Komponente "Sensor" (Seite 514) aus dem PROFIdrive-Telegramm.

Parameter

Parameter	Beschreibung
IncrementsPreRevolution	Geben Sie hier die Gebereigenschaft Geberstriche pro Umdrehung an.
DeterminableRevolutions	Geben Sie hier für rotatorisch absolute Geber die Anzahl der Umdrehungen an, in denen der Absolutwert ermittelt werden kann.
DistancePerRevolution	Geben Sie hier die Strecke in der von Ihnen gewählten physikalischen Einheit an, die bei einer Geberumdrehung zurückgelegt wird (z. B. 360.0 für 360°)
SupportsAbsoluteValues	Geben Sie hier an, ob ein absoluter (=TRUE) oder inkrementeller (=FALSE) Geber nachgebildet werden soll.

Universal – Ergänzungen der PROFIdrive-Grundfunktion

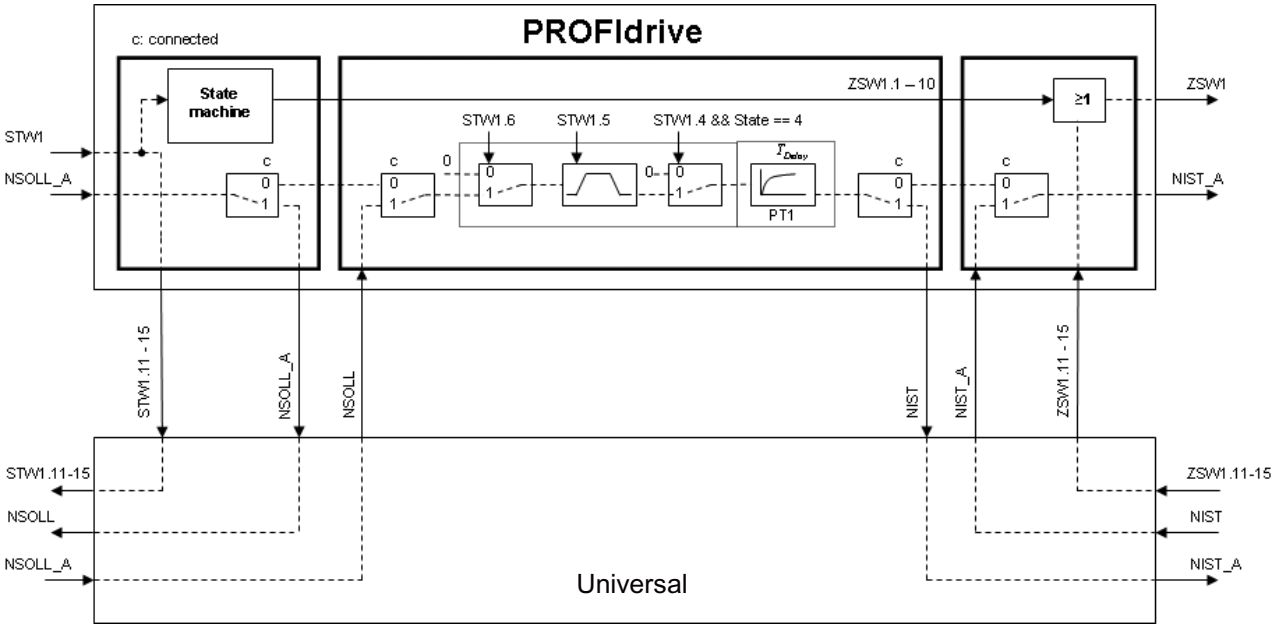
Symbol



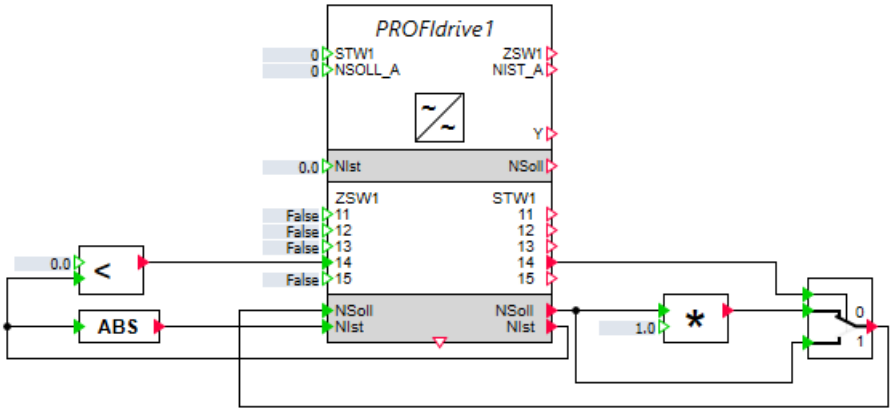
Funktion

Der Komponententyp *Universal* ist nur in Verbindung mit dem Komponententyp *PROFIdrive1* oder *PROFIdrive2* sinnvoll einsetzbar. Mit ihm können antriebsspezifische Funktionen zusätzlich zu den PROFIdrive-Grundfunktionen realisiert werden.

An den Ausgängen der Komponente *Universal* werden die Bits 11 bis 15 des Steuerworts (STW1.11–15) sowie der Drehzahlsollwert *NSOLL_A* und der Drehzahlistwert *NIST (NIST_PT1)* zur Verfügung gestellt. Über geeignete logische und arithmetische Verknüpfungen können daraus die Bits 11 bis 15 des Zustandsworts, der Drehzahlsollwert *NSOLL* (*NSOLL_RFG*) und der Drehzahlistwert *NIST_A* an den Eingängen der Komponente gesetzt werden. Die aus der Verschaltung der Komponente *Universal* mit der Komponente *PROFIdrive* resultierenden Signalverknüpfungen sind in der folgenden Abbildung dargestellt.



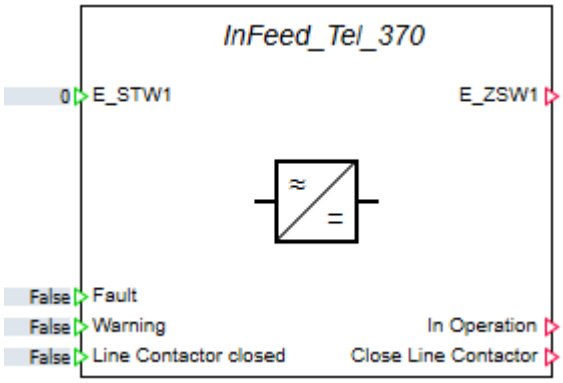
In der folgenden Abbildung ist beispielhaft dargestellt, wie die antriebsspezifischen Funktionen für den Micromaster vom Typ 3 mit Hilfe der Komponente *Universal* realisiert werden können.



Siemens

InFeed_Tel_370

Symbol



Funktion

Der Komponententyp *Infeed_Tel_370* simuliert eine elektrische Einspeiseeinheit, die durch Befehle, die vom PROFIdrive-Telegramm 370 transportiert werden, ein- und ausgeschaltet wird.

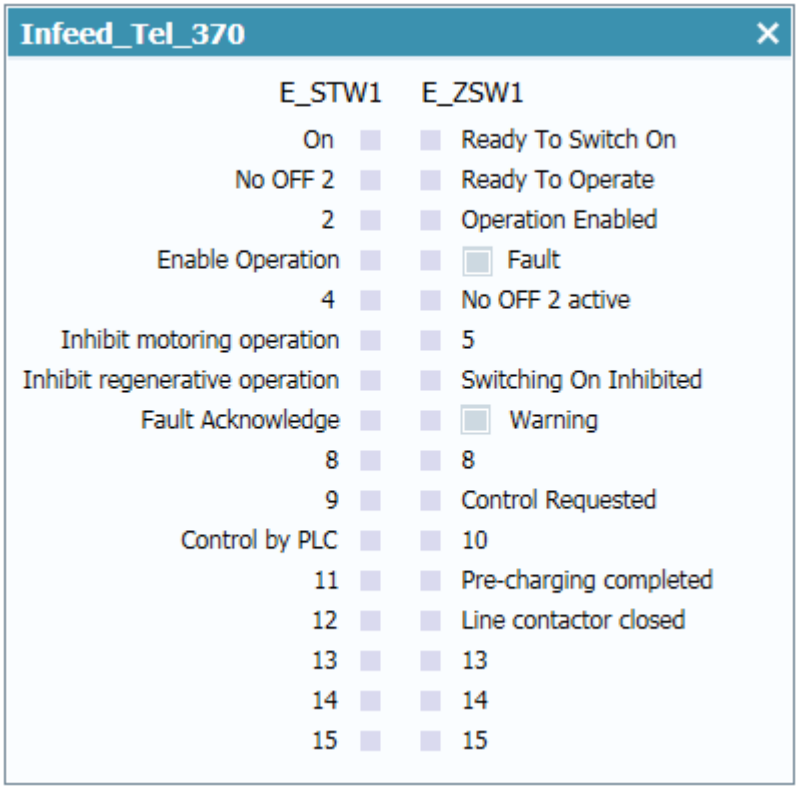
Telegrammstruktur p922=370

	Ausgangsdaten		**Eingangsdaten**	
Telegramm 370	Signal	Kommentar	Signal	Kommentar
PZD1	E_STW1		E_ZSW1	

Um eine Warn- oder Fehlersituation zu signalisieren, setzen Sie den Binäreingang *Warning* oder *Fault* von False auf True.

Wenn der Binärausgang *Close Line Contactor* auf True gesetzt ist, fordert die elektrische Einspeiseeinheit, das Linienschütz zu schließen. Diese Forderung muss zum Binäreingang *Line Contactor closed* rückgekoppelt werden. Dies geht einfach über eine direkte Verbindung oder die Verwendung eines Verzögerungsblocks, um detaillierter zu simulieren.

Dialogfeld



Das Dialogfeld "Infeed_Tel_370" zeigt den aktuellen Zustand der Steuer- und Statusworte innerhalb des Telegramms 370.

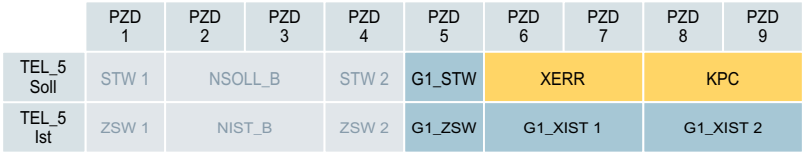
SiemensMomentumReduction – Analyse des Momentenreduktionswertes aus einem PROFIdrive-Telegramm

Symbol



Funktion

Der Komponententyp *SiemensMomentumReduction* dient zur Anreicherung eines PROFIdrive-Telegramms nach dem PROFIdrive-Profil (Technical Specification, Version 4.2, Stand: Oktober 2015 (Order No.: 3.172)) um die herstellerspezifische Funktionalität der Übertragung eines Momentenreduktionswertes.



Der Komponententyp *SiemensMomentumReduction* denormalisiert die übertragene Momentenreduktionsvorgabe.

Anschlüsse

Eingang	Beschreibung
MOMRED	Normierte Momentenreduktionsvorgabe aus dem PROFIdrive-Telegramm
DriveIn	Struktur zum Andocken an vorgelagerte PROFIdrive-Komponenten

Ausgang	Beschreibung
MELDW	Derzeit nicht verwendet und immer 0
M_RED	Reduziertes Drehmoment bezogen auf Parameter Torque_RedScale
M_Lower_Limit	Entsprechend reduzierte untere Drehmomentgrenze
M_Upper_Limit	Entsprechend reduzierte obere Drehmomentgrenze
DriveOut	Struktur zum Verschalten weiterer PROFIdrive-Komponenten

Parameter

Parameter	Beschreibung
Torque_Nominal	Geben Sie hier das Referenzdrehmoment an
Torque_UpperLimit	Geben Sie hier die physikalisch obere Drehmomentgrenze Ihres Antriebs an
Torque_LowerLimit	Geben Sie hier die physikalisch untere Drehmomentgrenze Ihres Antriebs an
Torque_RedScale	Legen Sie hier fest auf welchen Prozentsatz die Momentenreduktion normiert werden soll. Typischerweise entspricht das Referenzdrehmoment 100 %.

Sinamics – Frequenzumrichter SINAMICS

Symbol



Funktion

Im Komponententyp *Sinamics* sind die spezifischen Zusatzfunktionen für den Frequenzumrichter SINAMICS realisiert. Die Realisierung ist abgestimmt auf die im Funktionsbaustein SINA_GS enthaltenen Verarbeitungsschritte. Der Funktionsbaustein SINA_GS befindet sich in der Funktionsbaustein-Bibliothek DriveES-PCS 7.

Der Komponententyp *Sinamics* ist nur in Verbindung mit dem Komponententyp *PROFIdrive1* oder *PROFIdrive2* einsetzbar.

Tabelle 8-14 Sinamics-spezifische Auswertung des Steuerworts

STW1.11 **Sollwertinvertierung**	**NSOLL**
0	NSOLL
1	−NSOLL

Der Drehzahlistwert *NIst* und das Bit 11 des Zustandswortes werden entsprechend den in der folgenden Tabelle gelisteten Zusammenhängen gesetzt.

Tabelle 8-15 Sinamics-spezifische Zustände

NIst	**ZSW1.11** **Positive Drehrichtung**
NIst >= 0	1
NIst < 0	0

DCMaster – Stromrichter SIMOREG DC Master

Symbol



Funktion

Im Komponententyp *DCMaster* sind die spezifischen Zusatzfunktionen für den Stromrichter SIMOREG DC Master realisiert. Die Realisierung ist abgestimmt auf die im Funktionsbaustein SIMO_DC aus der Funktionsbausteinbibliothek Drive ES PCS 7 enthaltenen Verarbeitungsschritte.

Der Komponententyp *DCMaster* ist nur in Verbindung mit dem Komponententyp *PROFIdrive1* oder *PROFIdrive2* einsetzbar.

Tabelle 8-16 DCMaster-spezifische Auswertung des Steuerworts

STW1.11 Freigabe positive Drehrichtung	STW1.12 Freigabe negative Drehrichtung	NSOLL
0	0	0
0	1	-|NSOLL|
1	0	|NSOLL|
1	1	NSOLL

Der Drehzahlistwert *NIst* und die Bits 12 und 14 des Zustandsworts werden den in der folgenden Tabelle gelisteten Zusammenhängen entsprechend gesetzt.

Tabelle 8-17 DCMaster-spezifische Zustände

NIst	ZSW1.12 Anforderung Hauptschütz	ZSW1.14 Positive Drehrichtung
NIst > 0	1	1
NIst = 0	0	1
NIst < 0	1	0

Masterdrive – Frequenzumrichter SIMOVERT Masterdrive

Symbol



Funktion

Im Komponententyp *Masterdrive* sind die spezifischen Zusatzfunktionen für den Frequenzumrichter SIMOVERT Masterdrive realisiert. Die Realisierung ist abgestimmt auf die im Funktionsbaustein SIMO_MD enthaltenen Verarbeitungsschritte. Der Funktionsbaustein SIMO_MD befindet sich in der Funktionsbaustein-Bibliothek DriveES-PCS7.

Der Komponententyp *Masterdrive* ist nur in Verbindung mit dem Komponententyp *PROFIdrive1* oder *PROFIdrive2* einsetzbar.

Tabelle 8-18 Masterdrive-spezifische Auswertung des Steuerworts

STW1.11 Freigabe positive Drehrichtung	STW1.12 Freigabe negative Drehrichtung	NSOLL
0	0	0
0	1	-|NSOLL|
1	0	|NSOLL|
1	1	NSOLL

Der Drehzahlistwert *NIst* und die Bits 12 und 14 des Zustandsworts werden den in der folgenden Tabelle gelisteten Zusammenhängen entsprechend gesetzt.

Tabelle 8-19 Masterdrive-spezifische Zustände

NIst	ZSW1.12 Anforderung Hauptschütz	ZSW1.14 Positive Drehrichtung
NIst > 0	1	1
NIst = 0	0	1
NIst < 0	1	0

Micromaster3 – Frequenzumrichter MICROMASTER Typ 3

Symbol



Funktion

Im Komponententyp *Micromaster3* sind die spezifischen Zusatzfunktionen für den Frequenzumrichter MICROMASTER Typ 3 realisiert. Die Realisierung ist abgestimmt auf die im Funktionsbaustein SIMO_MM3 enthaltenen Verarbeitungsschritte. Der Funktionsbaustein SIMO_MM3 befindet sich in der Funktionsbaustein-Bibliothek DriveES-PCS7.

Der Komponententyp *Micromaster3* ist nur in Verbindung mit dem Komponententyp *PROFIdrive1* oder *PROFIdrive2* einsetzbar.

Tabelle 8-20 Micromaster3-spezifische Auswertung des Steuerworts

STW1.14 Rechtslauf	NSOLL
0	- NSOLL
1	NSOLL

Der Drehzahlistwert *NIst* und das Bit 14 des Zustandsworts werden den in der folgenden Tabelle gelisteten Zusammenhängen entsprechend gesetzt.

Tabelle 8-21 Micromaster3-spezifische Zustände

NIst	ZSW1.14 Rechtslauf
NIST >= 0	1
NIST < 0	0

Micromaster4 – Frequenzumrichter MICROMASTER Typ 4

Symbol



Funktion

Im Komponententyp *Micromaster4* sind die spezifischen Zusatzfunktionen für den Frequenzumrichter MICROMASTER Typ 4 realisiert. Die Realisierung ist abgestimmt auf die im Funktionsbaustein SIMO_MM4 enthaltenen Verarbeitungsschritte. Der Funktionsbaustein SIMO_MM4 befindet sich in der Funktionsbaustein-Bibliothek DriveES-PCS7.

Der Komponententyp *Micromaster4* ist nur in Verbindung mit dem Komponententyp *PROFIdrive1* oder *PROFIdrive2* einsetzbar.

Tabelle 8-22 Micromaster4-spezifische Auswertung des Steuerworts

STW1.11 Sollwertinvertierung	NSOLL
0	NSOLL
1	- NSOLL

Der Drehzahlistwert *NIst* und das Bit 11 des Zustandsworts werden den in der folgenden Tabelle gelisteten Zusammenhängen entsprechend gesetzt.

Tabelle 8-23 Micromaster4-spezifische Zustände

NIst	ZSW1.11 Positive Drehrichtung
NIST >= 0	1
NIST < 0	0

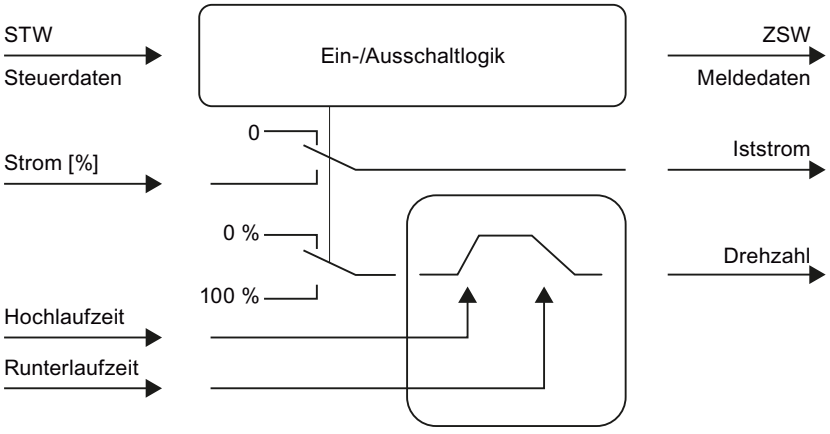
8.1.4.6 Motorsteuergeräte SIMOCODE pro

Die Motormanagement- und Steuergeräte SIMOCODE pro werden eingesetzt, um Motoren ein- und auszuschalten und die daraus resultierenden Ströme überwachen zu können. Optional können mit SIMOCODE pro weitere Messwerte erfasst und umfangreiche statistische Auswertungen abgefragt werden. Ein SIMOCODE pro-Gerät ist als individueller PROFIBUS DP-Slave mit der Steuerung verbunden.

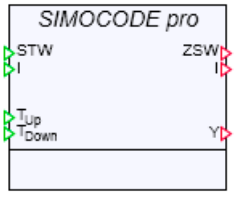
Durch entsprechende Parametrierung kann ein SIMOCODE pro ganz unterschiedliche Aufgaben übernehmen. Er kann als Direktstarter, Wendestarter, Stern- / Dreieckstarter mit oder ohne Drehrichtungsumkehr, Polumschalter mit oder ohne Drehrichtungsumkehr, sowie als Positionierer, Magnetventilantrieb, Überlastrelais oder Leistungsschalter fungieren. Im Verzeichnis *SIMOCODEpro* der Antriebsbibliothek sind Komponententypen enthalten, die die verschiedenen Steuerfunktionen eines SIMOCODE pro nachbilden. Diese Komponententypen bilden die SIMOCODEpro-Bibliothek von SIMIT.

Grundfunktionen der SIMOCODE-pro-Komponenten

In jedem Komponententyp der SIMOCODEpro-Bibliothek sind die Ein- und Ausschaltlogik sowie eine einfache Nachbildung des Motors als Grundfunktion enthalten.



Die entsprechenden Anschlüsse der *SIMOCODEpro*-Komponententypen sind in dem in der folgenden Abbildung dargestellten Symbol zu sehen.



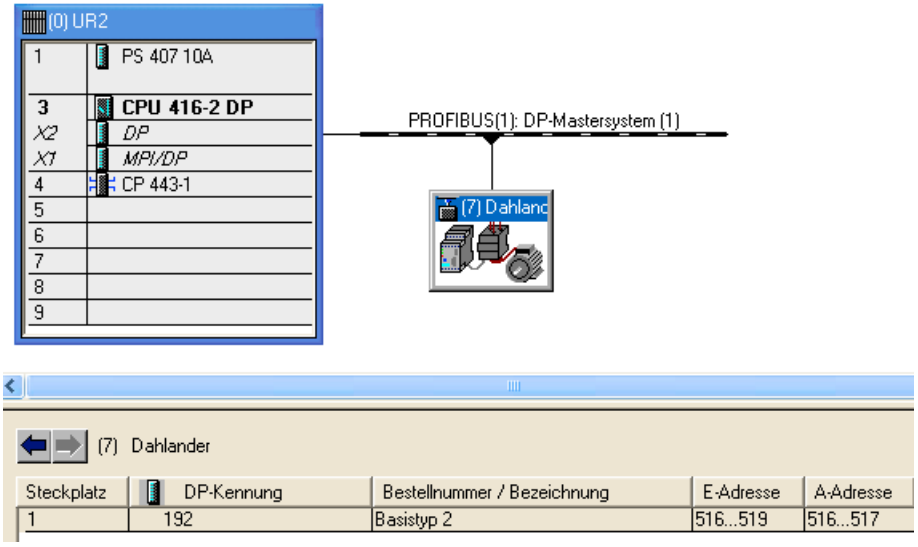
Alle SIMOCODE pro-Geräte werden über Steuerdaten angesprochen und geben ihren aktuellen Zustand über Meldedaten an die Steuerung zurück. Die Belegung der Steuer- und Meldedaten ist davon abhängig, welche Steuerfunktion realisiert ist. In den *SIMOCODEpro*-Komponententypen werden nur die zyklischen Steuer- und Meldedaten behandelt; azyklische

Daten, wie beispielsweise Statistikdaten, werden nicht berücksichtigt. Die folgenden Kopplungsmöglichkeiten der Komponenten mit der Steuerung sind möglich:

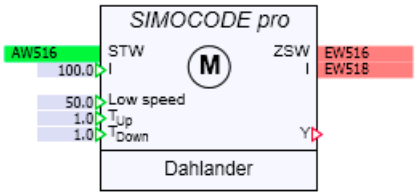
- Das Steuerdatenwort (2 Byte) im Ausgangsbereich der Steuerung wird mit dem ganzzahligen (Integer) Eingang *STW* der Komponente verbunden.
- Das erste Datenwort der Meldedaten im Eingangsbereich der Steuerung enthält die binären Rückmeldungen und wird mit dem ganzzahligen (Integer) Ausgang *ZSW* verbunden. Im zweiten Datenwort werden analoge Meldedaten, wie beispielsweise der Stromistwert an die Steuerung übertragen.

Diese Konfiguration entspricht dem SIMOCODE pro C oder beim SIMOCODE pro V dem Basistyp 2.

In der folgenden Abbildung ist die SIMATIC-Projektierung beispielhaft für einen Dahlandermotor dargestellt.



Die folgende Abbildung zeigt die *SIMOCODEpro*-Komponente mit den verbundenen Prozessdaten.



Der Iststrom *I* wird als ein auf den Einstellstrom (Motorbemessungsstrom) bezogener Prozentwert an die Steuerung übermittelt. Der am Eingang *I* anliegende Stromwert wird als Iststrom auf den Ausgang *I* durchgeschaltet, solange der Motor eingeschaltet ist. Vorbelegt ist der Eingang mit einem Wert von 100 %, d. h. der Iststrom ist gleich dem Einstellstrom. Das Ein- und Ausschalten des Motors wird aus einer Rampenfunktion abgeleitet.

Die Abhängigkeit des Stromes von der Belastung des Motors ist nicht berücksichtigt. Sie können ein solches Verhalten in der Simulation aber leicht selbst ergänzen, indem Sie beispielsweise den Strom nicht als konstanten Wert vorgeben, sondern über geeignete

Funktionen von der Motordrehzahl oder dem Prozessgeschehen abhängig machen. Weitere Informationen hierzu finden Sie im Kapitel: Individuelle Anpassungen (Seite 532).

Am Ausgang *Y* einer *SIMOCODEpro*-Komponente steht die Motordrehzahl als prozentualer Wert zur Verfügung. Die Hoch- bzw. Runterlaufzeiten des Motors werden an den beiden Analogeingängen T_{Up} und T_{Down} vorgegeben. T_{Up} ist die Laufzeit des Motors zum Hochfahren vom Stillstand auf die Nenndrehzahl in Sekunden, T_{Down} ist die Zeit zum Abfahren des Antriebs von Nenndrehzahl bis zum Stillstand in Sekunden. Voreingestellt sind beide Zeiten mit einer Sekunde. Falls beim Durchführen der Simulation einer der beiden Eingangswerte negativ ist, wird die Meldung "*x: run-up or run-down time invalid value*" (Meldekategorie *ERROR*) erzeugt.

Hinweis

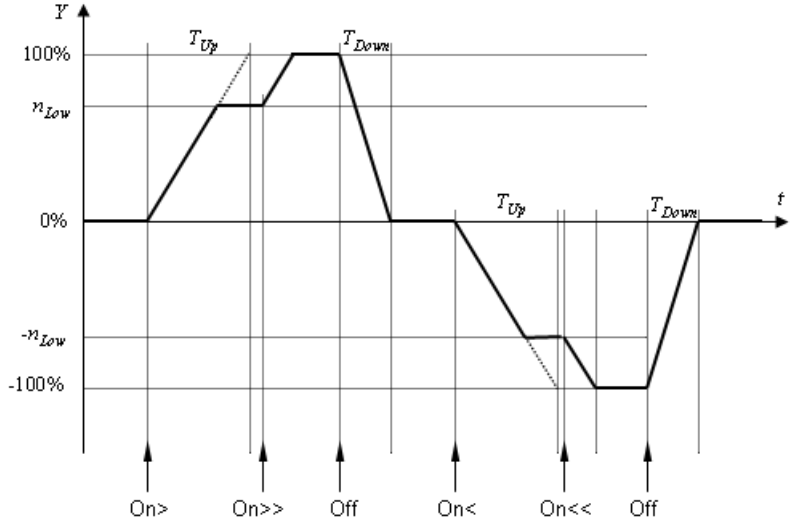
Beim Positionier- und Magnetventilantrieb haben die Hoch- und Runterlaufzeit die Bedeutung der Öffnungs- und Schließzeit, T_{Up} und T_{Down}, des Ventils.

Die Motordrehzahl wird weder von der Steuerung vorgegeben noch von SIMOCODE pro-Geräten erfasst. Der Drehzahlwert wird daher nicht an die Steuerung geführt: Er dient dazu, weiteres Simulationsverhalten aus der Motordrehzahl abzuleiten. Entsprechend ist die Einstellung der Hoch- und Runterlaufzeiten von geringer Bedeutung.

Umschaltpausen und Verriegelungszeiten werden in den *SIMOCODEpro*-Komponententypen nicht berücksichtigt. Entsprechende Rückmeldungen für die Steuerung werden nicht generiert.

Rampenfunktion

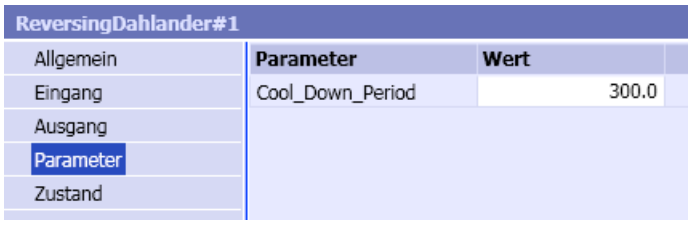
Die Motordrehzahl wird mit Hilfe einer Rampenfunktion gebildet. Die allgemeinste Form dieser Rampe für einen Antrieb mit zwei Drehzahlen in zwei Drehrichtungen ist in der folgenden Abbildung dargestellt.



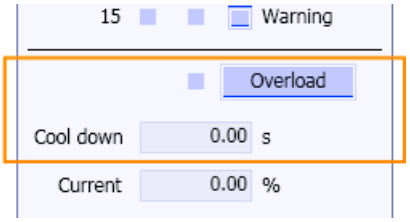
Die Anstiegs- und Abstiegszeiten der Rampe entsprechen den Hoch- und Runterlaufzeiten T_{Up} und T_{Down} der Motoren. Je nach Konfiguration des SIMOCODE pro wird so die Motordrehzahl *Y* als prozentualer Wert im Bereich von −100 % bis 100 % gebildet.

Überlastverhalten

Überlast kann mit dem Schalter *Overload* im Bedienfenster einer Komponente gesetzt werden. Bei Überlast wird der Motor abgeschaltet. Das Wiedereinschalten wird durch eine thermische Einschaltsperre (Abkühlzeit) kontrolliert. Mit dem Abschalten der Überlast beginnt die Abkühlperiode. Die Abkühldauer ist im Parameter *Cool_Down_Period* mit 300 Sekunden vorbelegt, wie in der folgenden Abbildung dargestellt.



Die verbleibende Abkühlzeit wird im Feld *Cool Down* des Bedienfensters angezeigt, wie folgende Abbildung zeigt:



Während dieser Zeit ist der Antrieb gesperrt, d. h. es ist nicht möglich den Antrieb wieder zu starten.

Durch Setzen des Notstarts (*EM-Start*, *STW.12*) wird die thermische Einschaltsperre zurückgesetzt. Der Antrieb kann sofort wieder gestartet werden.

Die Vorwarnung für Überlast (*I > 115 %*, *ZSW.11*) wird gesetzt, sobald der Eingangsstrom größer als 115 % ist.

Standardbelegungen in den Steuer- und Meldedaten

Die Bits 11 bis 15 Steuerwortes (STW.11 bis STW.15) werden mit den in der folgenden Tabelle beschriebenen Funktionen einheitlich in allen *SIMOCODEpro*-Komponententypen verwendet. Die Funktion Notstart entfällt beim Magnetventil.

Tabelle 8-24 Standardbelegungen im Steuerwort

Name	Steuerdaten	
Test1	Testfunktion: Reset nach 5 Sekunden	STW.11
EM-Start	Notstart	STW.12
Remote	Betriebsartenumschalter S1	STW.13
Reset	Gerät rücksetzen	STW.14

Die Testfunktion *Test1* setzt fünf Sekunden nach dem Setzen des Signals das SIMOCODE-pro-Gerät zurück. Der Motor wird abgeschaltet.

Das Setzen von *EM-Start* führt zum Rücksetzen der thermischen Einschaltsperre bei Überlastauslösung. Der Motor kann dann sofort wieder eingeschaltet werden.

Der Befehl *Remote* zeigt die Vorgabe der Steuerung für den Betriebsartenumschalter S1 an. Dieser Befehl hat funktional keine Auswirkungen in der Simulation.

Mit dem Befehl *Reset* wird das SIMOCODE pro-Gerät zurückgesetzt. Der Motor wird dadurch ausgeschaltet.

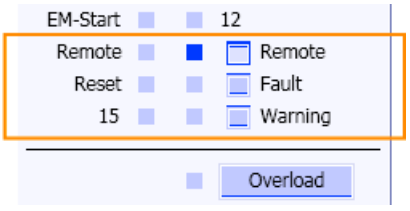
In der folgenden Tabelle sind die in allen *SIMOCODEpro*-Komponententypen einheitlichen Meldedaten zusammengestellt. Die Vorwarnung Überlast entfällt beim Magnetventil.

Tabelle 8-25 Standardbelegungen im Zustandswort

Name	Meldedaten	
I>115%	Vorwarnung Überlast	ZSW.11
Remote	Betriebsart Fern	ZSW.13
Fault	Sammelstörung	ZSW.14
Warning	Sammelwarnung	ZSW.15

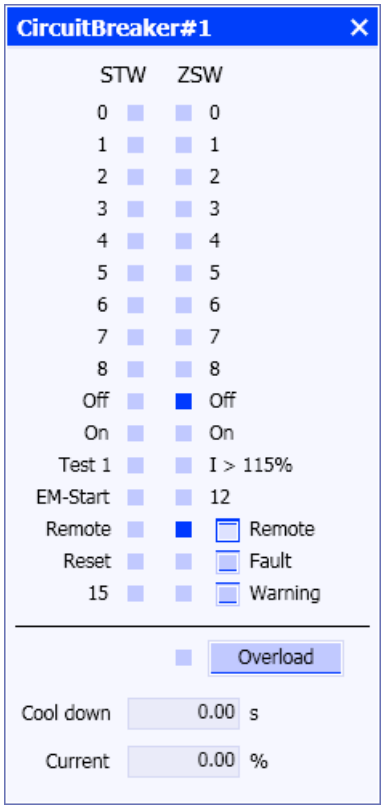
Wenn der Iststrom über den Wert von 115 % steigt, wird Bit ZSW.11 als Vorwarnsignal für Überlast gesetzt.

Die Signale *Remote*, *Fault* und *Warning* können im Bedienfenster der Komponente gesetzt werden. Das Signal *Remote* ist mit eins vorbelegt, d. h. der Schalter ist geschlossen.



Bedienfenster der SIMOCODEpro-Komponenten

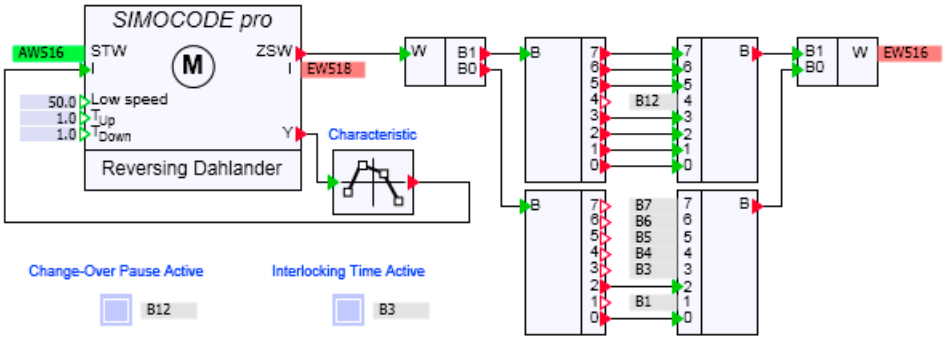
Alle *SIMOCODEpro*-Komponenten enthalten ein Bedienfenster, in dem die Signale des Steuer- und Zustandsworts angezeigt werden. Die jeweils in einer Komponente verwendeten Signale des Steuerworts sind im Bedienfenster durch Namen gekennzeichnet. Gleiches gilt für die von der Komponente beeinflussten Signale der Meldedaten. Mit Zahlen gekennzeichnete Signale haben in der Komponente keine Funktion und werden im Bedienfenster nur angezeigt. Folgende Abbildung zeigt das Bedienfenster für die Komponente vom Typ *ReversingDahlander*. Die Bedienfenster für Komponenten der anderen Typen sind entsprechend gestaltet.



Individuelle Anpassungen

Abhängig davon, wie Sie ein SIMOCODE pro-Gerät in der Automatisierung parametrieren, können auch Signale in den Steuer- und Meldedaten eine Bedeutung erhalten, die in den Komponenten der SIMOCODEpro-Bibliothek nicht belegt sind. Sie können diese Signale aber durch eine einfache Verschaltung mit anderen Komponenten aus der SIMIT-Basisbibliothek in die Simulation einbeziehen.

In dem in der folgenden Abbildung dargestellten Beispiel einer *ReversingDahlander*-Komponente können die Verriegelungszeit und die Umschaltpause manuell gesetzt werden. Beide Signale werden in der Komponente nicht gesetzt. Das Zustandswort am Ausgang der Komponente ist hier über Konvertierkomponenten (*Word2Byte*, *Byte2Bit*) in seine einzelnen Signale zerlegt worden. Die in der Komponente gesetzten Signale werden mit den über die globalen Konnektoren verknüpften Signalen wieder in ein Wort konvertiert und an den Meldeeingang *EW516* der Steuerung geführt.



Weiter ist hier auch beispielhaft gezeigt, wie aus der Drehzahl des Motors am Ausgang *Y* der *ReversingDahlander*-Komponente mit Hilfe einer Abbildung über eine geeignete Kurvenfunktion (*Characteristic*) der Strom am Eingang der *ReversingDahlander*-Komponente variabel gesetzt werden kann.

Spezifische SIMOCODE pro-Geräte

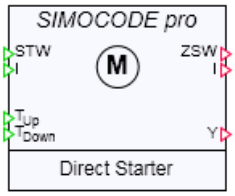
In der SIMOCODEpro-Bibliothek sind zehn verschiedene *SIMOCODEpro*-Komponententypen zu finden. Diese Komponententypen bilden die verschiedenen Steuerfunktionen eines SIMOCODE pro nach. Jede Steuerfunktion wird durch einen Komponententyp nachgebildet. Die Simulationsfunktion der Komponententypen ist abgestimmt auf die entsprechenden Ansteuervarianten (Funktionsblöcke) in PCS 7.

Tabelle 8-26 Unterstützte Steuerfunktionen des SIMOCODE pro

SIMIT-Komponententyp	Steuerfunktion	PCS 7-FB
DirectStarter	Direktstarter	SMC_DIR
ReversingStarter	Wendestarter	SMC_REV
StarDeltaStarter	Stern- / Dreieckstarter	SMC_STAR
ReversingStarDelta	Stern- / Dreieckstarter mit Drehrichtungs-umkehr	SMC_REVS
Dahlander	Dahlander / Polumschalter	SMC_DAHL
ReversingDahlander	Dahlander / Polumschalter mit Drehrich-tungsumkehr	SMC_REVD
Valve	Magnetventil	SMC_VAL
Positioner	Schieber	SMC_POS
OverloadRelay	Überlast	SMC_OVL
CircuitBreaker	Leistungsschalter	SMC_CB

DirectStarter – Direktstarter

Symbol



Funktion

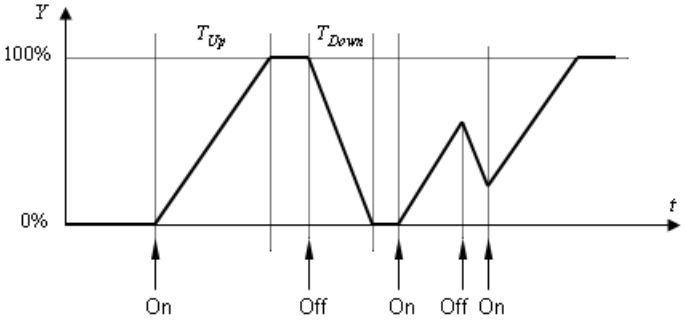
Der Komponententyp *DirectStarter* simuliert Antriebe, die eine Drehrichtung haben und direkt ein- und ausschaltbar sind. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

Tabelle 8-27 Steuer- und Meldedaten der Komponente DirectStarter

Name	Steuerdaten		Meldedaten	
Off	Antrieb ausschalten	STW.9	Antrieb ausgeschaltet	ZSW.9
On	Antrieb einschalten	STW.10	Antrieb eingeschaltet	ZSW.10

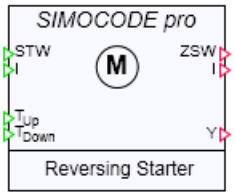
Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Befehl "Antrieb einschalten" (On, STW.10).

Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb eingeschaltet ist. Ist der Antrieb ausgeschaltet, wird er auf null gesetzt. Der Drehzahlwert am Ausgang *Y* wird als Prozentwert mit Hilfe der Rampenfunktion gebildet: 0 ≤ *Y* ≤ 100.



ReversingStarter – Wendestarter

Symbol



Funktion

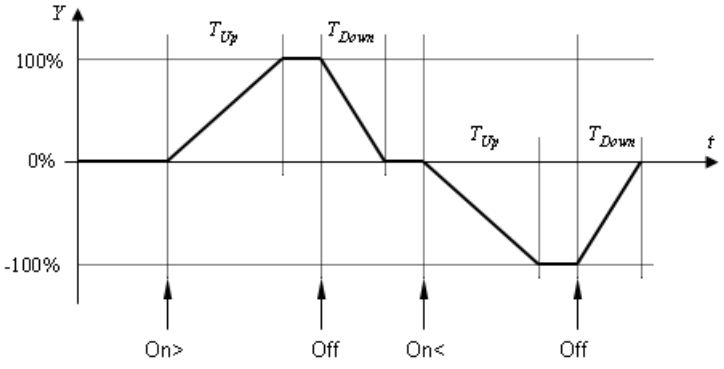
Der Komponententyp *ReversingStarter* simuliert Zweidrehrichtungsantriebe die direkt in beiden Drehrichtungen ein- und ausschaltbar sind. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

Tabelle 8-28 Steuer- und Meldedaten der Komponente DirectStarter

Name	Steuerdaten		Meldedaten	
On<	Linkslauf einschalten	STW.8	Linkslauf eingeschaltet	ZSW.8
Off	Antrieb ausschalten	STW.9	Antrieb ausgeschaltet	ZSW.9
On>	Rechtslauf einschalten	STW.10	Rechtslauf eingeschaltet	ZSW.10

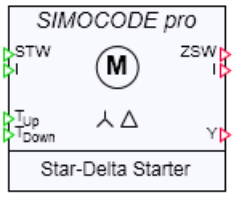
Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Befehl "Linkslauf einschalten" bzw. "Rechtslauf einschalten" (On<, STW.8 bzw. On>, STW.10). Gleichzeitige Einschaltbefehle für beide Drehrichtungen ändern den Zustand des Antriebs nicht, sie werden ignoriert.

Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb in eine der beiden Drehrichtungen eingeschaltet ist. Ist der Antrieb ausgeschaltet, wird der Stromistwert auf null gesetzt. Der Drehzahlwert am Ausgang *y* wird als Prozentwert mit Hilfe der Rampenfunktion gebildet: $-100 \leq Y \leq 100$.



StarDeltaStarter – Stern-Dreieck-Starter

Symbol



Funktion

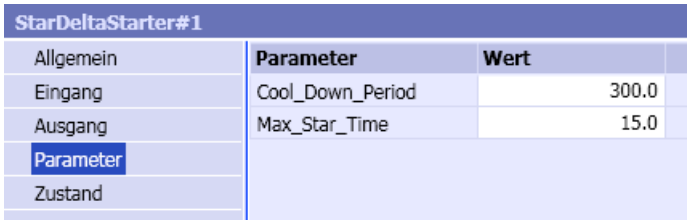
Der Komponententyp *StarDeltaStarter* simuliert Antriebe mit Stern-Dreiecks-Umschaltung. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

Tabelle 8-29 Steuer- und Meldedaten der Komponente StarDeltaStarter

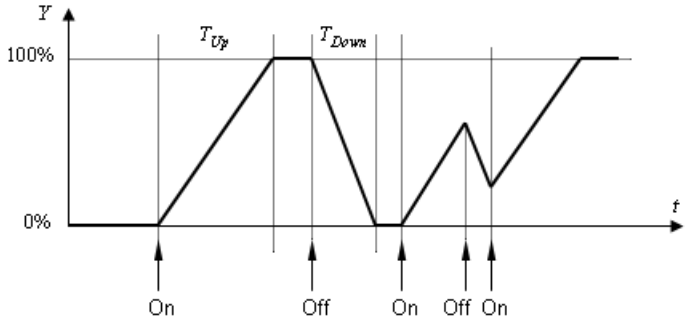
Name	Steuerdaten		Meldedaten	
Off	Antrieb ausschalten	STW.9	Antrieb ausgeschaltet	ZSW.9
On	Antrieb einschalten	STW.10	Dreiecksbetrieb eingeschaltet	ZSW.10

Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Befehl "Antrieb einschalten" (On, STW.10).

Die maximale Zeit für den Sternbetrieb wird über den Parameter *Max_Star_Time* eingestellt. Voreingestellt ist eine Zeit von 15 Sekunden. Die Umschaltung in den Dreiecksbetrieb (ZSW.10) erfolgt, wenn die maximale Sternbetriebszeit erreicht ist oder wenn am Stromeingang ein Wert kleiner als 90 % anliegt.

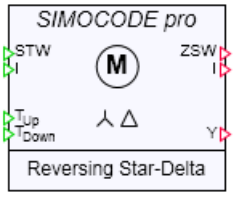


Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb eingeschaltet ist. Ist der Antrieb ausgeschaltet, wird der Stromistwert auf null gesetzt. Der Drehzahlwert am Ausgang *Y* wird als Prozentwert mit Hilfe der Rampenfunktion gebildet: 0 ≤ *Y* ≤ 100.



ReversingStarDelta – Stern-Dreieck-Starter mit Drehrichtungsumkehr

Symbol



Funktion

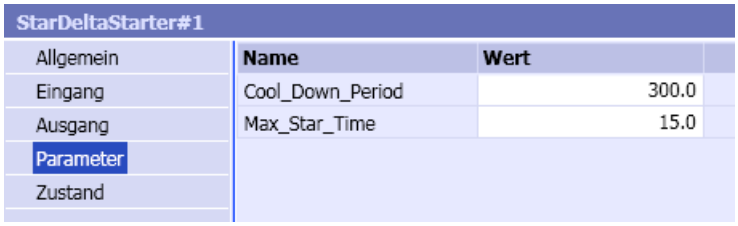
Der Komponententyp *ReversingStarDelta* simuliert Antriebe mit Stern-Dreiecks-Umschaltung in beiden Drehrichtungen. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

Tabelle 8-30 Steuer- und Meldedaten der Komponente ReversingStarDelta

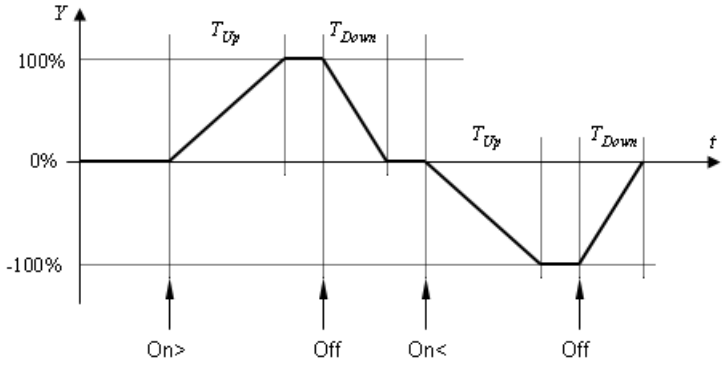
Name	Steuerdaten		Meldedaten	
On<	Linkslauf einschalten	STW.8	Dreiecksbetrieb links eingeschaltet	ZSW.8
Off	Antrieb ausschalten	STW.9	Antrieb ausgeschaltet	ZSW.9
On>	Rechtslauf einschalten	STW.10	Dreiecksbetrieb rechts eingeschaltet	ZSW.10

Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Befehl "Linkslauf einschalten" bzw. "Rechtslauf einschalten" (On<, STW.8 bzw. On>, STW.10). Gleichzeitige Einschaltbefehle für beide Drehrichtungen ändern den Zustand des Antriebs nicht, sie werden ignoriert.

Die maximale Zeit für den Sternbetrieb wird über den Parameter *Max_Star_Time* eingestellt. Voreingestellt ist eine Zeit von 15 Sekunden. Die drehrichtungsabhängige Umschaltung in den Dreiecksbetrieb (ZSW.10 bzw. ZSW.8) erfolgt, wenn die maximale Sternbetriebszeit erreicht ist oder wenn am Stromeingang ein Wert kleiner als 90 % anliegt.

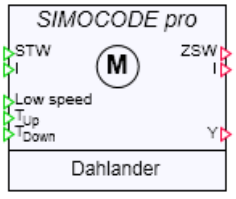


Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb in eine der beiden Drehrichtungen eingeschaltet ist. Ist der Antrieb ausgeschaltet, wird der Stromistwert auf null gesetzt. Der Drehzahlwert am Ausgang *y* wird als Prozentwert mit Hilfe der Rampenfunktion gebildet: −100 ≤ *Y* ≤ 100.



Dahlander – Dahlander-Starter oder Polumschalter

Symbol



Funktion

Der Komponententyp *Dahlander* simuliert Antriebe mit einer Drehrichtung und zwei Drehzahlen: Volldrehzahl und Teildrehzahl. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

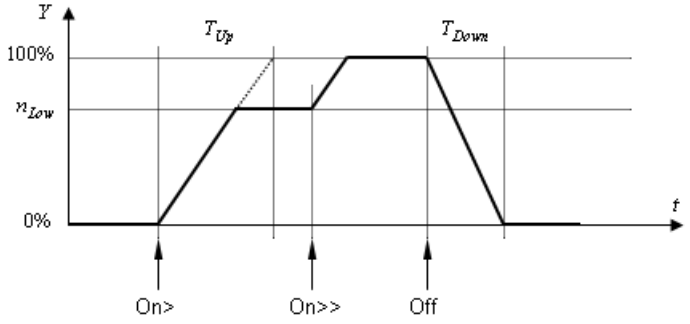
Tabelle 8-31 Steuer- und Meldedaten der Komponente Dahlander

Name	Steuerdaten		Meldedaten	
On>>	Antrieb auf volle Drehzahl schalten	STW.8	Antrieb auf volle Drehzahl geschaltet	ZSW.8
Off	Antrieb ausschalten	STW.9	Antrieb ausgeschaltet	ZSW.9
On>	Antrieb auf Teildrehzahl schalten	STW.10	Antrieb auf Teildrehzahl geschaltet	ZSW.10

Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor den Befehlen "Antrieb auf volle Drehzahl schalten" (On>>, STW.8) oder "Antrieb auf Teildrehzahl schalten" (On>, STW.10). Gleichzeitige Einschaltbefehle für beide Drehzahlen ändern den Zustand des Antriebs nicht, sie werden ignoriert.

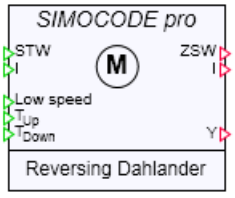
Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb auf eine der beiden Drehzahlen geschaltet ist. Ist der Antrieb ausgeschaltet, wird der Stromistwert auf null gesetzt.

Die Teildrehzahl n_{Low} wird als Prozentwert am Eingang *Low speed* gesetzt. Voreingestellt ist eine Teildrehzahl von 50 %. Der Drehzahlwert am Ausgang *Y* wird als Prozentwert mit Hilfe der Rampenfunktion gebildet: $0 \leq Y \leq 100$.



ReversingDahlander – Dahlander-Starter oder Polumschalter mit Drehrichtungsumkehr

Symbol



Funktion

Der Komponententyp *ReversingDahlander* simuliert Antriebe mit zwei Drehzahlen in zwei Drehrichtungen. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

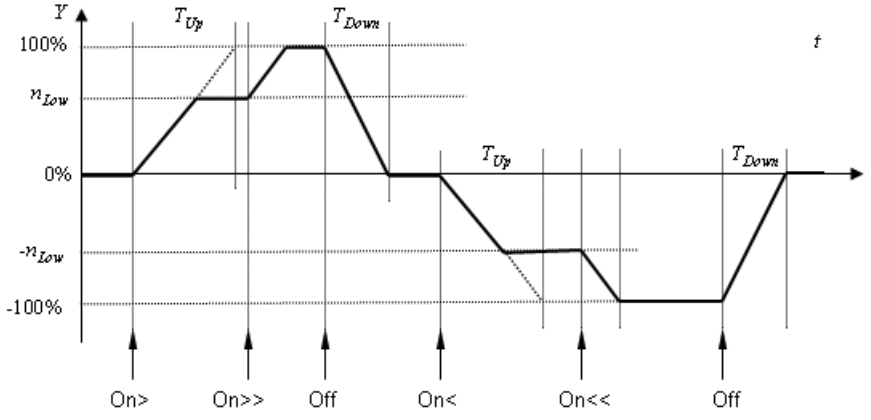
Tabelle 8-32 Steuer- und Meldedaten der Komponente ReversingDahlander

Name	Steuerdaten		Meldedaten	
On<<	Antrieb auf Linkslauf mit voller Drehzahl schalten	STW.0	Antrieb auf Linkslauf mit voller Drehzahl geschaltet	ZSW.0
On<	Antrieb auf Linkslauf mit Teildrehzahl schalten	STW.2	Antrieb auf Linkslauf mit Teildrehzahl geschaltet	ZSW.2
On>>	Antrieb auf Rechtslauf mit voller Drehzahl schalten	STW.8	Antrieb auf Rechtslauf mit voller Drehzahl geschaltet	ZSW.8
On>	Antrieb auf Rechtslauf mit Teildrehzahl schalten	STW.10	Antrieb auf Rechtslauf mit Teildrehzahl geschaltet	ZSW.10
Off	Antrieb ausschalten	STW.9	Antrieb ausgeschaltet	ZSW.9

Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor den Einschaltbefehlen "Linkslauf mit voller Drehzahl" (On<<, STW.0), "Linkslauf mit Teildrehzahl" (ON<, STW.2), "Rechtslauf mit voller Drehzahl" (On>>, STW. 8) und "Rechtslauf auf Teildrehzahl" (On>, STW.10). Mehrere gleichzeitige Einschaltbefehle ändern den Zustand des Antriebs nicht, sie werden ignoriert.

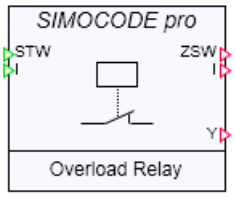
Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb eingeschaltet ist. Ist der Antrieb ausgeschaltet, wird der Stromistwert auf null gesetzt.

Die Teildrehzahl n_{Low} wird als Prozentwert am Eingang *Low speed* gesetzt. Voreingestellt ist eine Teildrehzahl von 50 %. Der Drehzahlwert am Ausgang *Y* wird als Prozentwert mit Hilfe der Rampenfunktion gebildet: $-100 \leq Y \leq 100$.



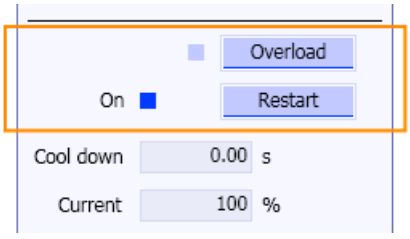
OverloadRelay – Überlastrelais

Symbol

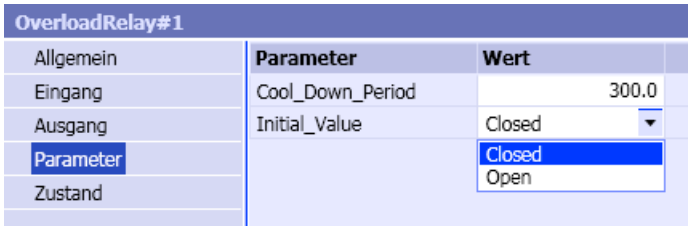


Funktion

Der Komponententyp *OverloadRelay* simuliert Antriebe mit Überlastüberwachung. Mit dem Befehl *Overload* wird Überlast ausgelöst, d. h. der Antrieb wird ausgeschaltet, mit dem Befehl *Restart* wird der Antrieb eingeschaltet. Die beiden Befehle können ausschließlich im Bedienfenster der Komponente gesetzt werden.



Mit dem Parameter *Initial_Value* kann der Antrieb als eingeschaltet oder ausgeschaltet initialisiert werden: Eingeschaltet mit dem Wert *Closed*, ausgeschaltet mit dem Wert *Open*. Voreinstellung für *Initial_Value* ist *Closed*.

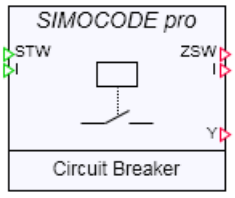


Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Antrieb eingeschaltet ist. Ist der Antrieb ausgeschaltet, wird der Stromistwert auf null gesetzt.

Am Ausgang *Y* wird der Wert hundert für den eingeschalteten Antrieb und der Wert null für den ausgeschalteten Antrieb ausgegeben.

CircuitBreaker – Leistungsschalter

Symbol



Funktion

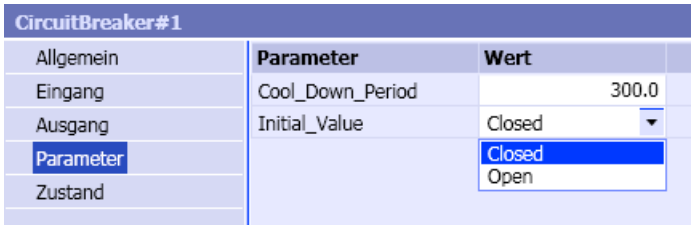
Der Komponententyp *CircuitBreaker* simuliert Antriebe mit Schaltverhalten. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

Tabelle 8-33 Steuer- und Meldedaten der Komponente CircuitBreaker

Name	Steuerdaten		Meldedaten	
Off	Schalter öffnen	STW.9	Schalter öffnet / offen	ZSW.9
On	Schalter schließen	STW.10	Schalter schließt / geschlossen	ZSW.10

Die Befehle "Schalter öffnen" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Schließbefehl (On, STW.10).

Mit dem Parameter *Initial_Value* kann der Schalter als geschlossen oder geöffnet initialisiert werden: Geschlossen mit dem Wert *Closed*, geöffnet mit dem Wert *Open*. Voreinstellung für *Initial_Value* ist *Closed*.

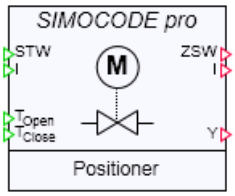


Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, sobald der Schalter schließt. Wird der Schalter geöffnet, dann wird der Stromistwert auf null gesetzt.

Am Ausgang *Y* wird der Wert hundert für den eingeschalteten Antrieb und der Wert null für den ausgeschalteten Antrieb ausgegeben.

Positioner – Schieber/Positionierer

Symbol



Funktion

Der Komponententyp *Positioner* simuliert Positionierantriebe für Schieber, Stellventile etc. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

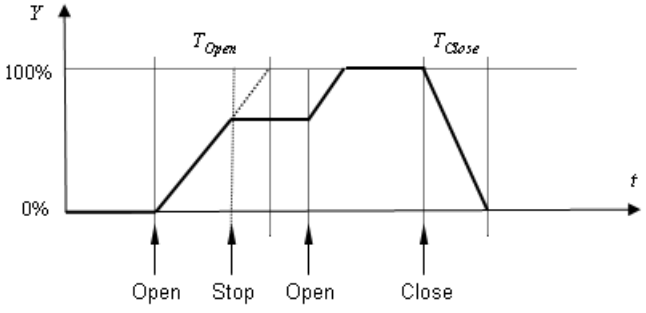
Die Befehle "Antrieb ausschalten" (Off, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Öffnungs- und Schließbefehl (Open, STW.10 und Close, STW.8). Gleichzeitig gesetzte Öffnungs- und Schließbefehle ändern den Zustand des Antriebs nicht, sie werden ignoriert.

Tabelle 8-34 Steuer- und Meldedaten der Komponente Positioner

Name	Steuerdaten		Meldedaten	
Close	Schieber schließen	STW.8	Schieber geschlossen	ZSW.8
			Schieber schließt	ZSW.2
Stop	Schieber anhalten	STW.9	Schieber steht	ZSW.9
Open	Schieber öffnen	STW.10	Schieber offen	ZSW.10
			Schieber öffnet	ZSW.0

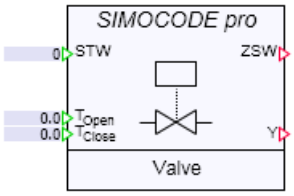
Der Stromistwert am Ausgang *I* wird gleich dem Wert am Stromeingang *I* gesetzt, solange der Schieber öffnet (ZSW.0) oder schließt (ZSW.2). Der Stromistwert wird auf null gesetzt, wenn der Schieber steht und wenn er geschlossen oder geöffnet ist.

Der Positionswert am Ausgang *Y* wird als Prozentwert mit Hilfe einer Rampenfunktion gebildet: $0 \leq Y \leq 100$. Der Wert null entspricht dem geschlossenen Schieber, der Wert einhundert dem geöffneten Schieber. Die Öffnungs- und Schließzeiten des Schiebers sind mit jeweils einer Sekunde voreingestellt.



Valve – Magnetventil

Symbol



Funktion

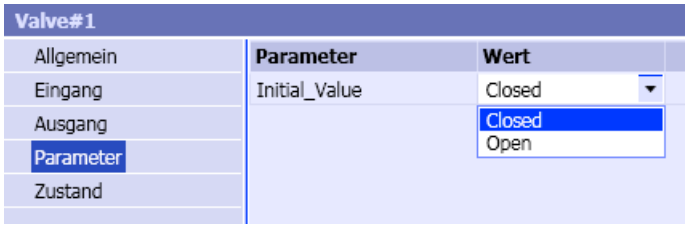
Der Komponententyp *Valve* simuliert Antriebe für Magnetventile. In der folgenden Tabelle sind die für diesen Anwendungsfall relevanten Steuerdaten *STW* und Meldedaten *ZSW* zusammengefasst.

Tabelle 8-35 Steuer- und Meldedaten der Komponente Valve

Name	Steuerdaten		Meldedaten	
Close	Ventil schließen	STW.9	Ventil schließt / geschlossen	ZSW.9
Open	Ventil öffnen	STW.10	Ventil öffnet / geöffnet	ZSW.10

Die Befehle "Ventil schließen" (Close, STW.9) und "Rücksetzen" (RESET, STW.14) haben Vorrang vor dem Öffnungsbefehl (Open, STW.10 und Close, STW.8).

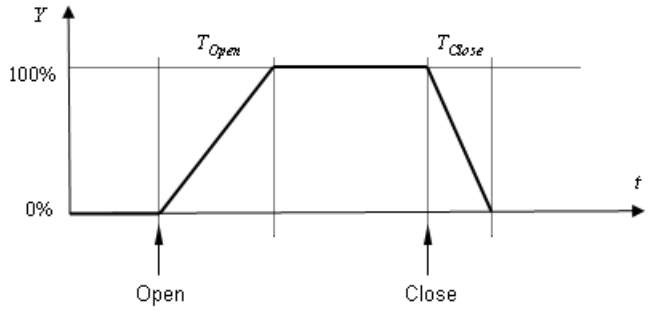
Mit dem Parameter *Initial_Value* kann das Ventil als geschlossen oder geöffnet initialisiert werden: Geschlossen mit dem Wert *Closed*, geöffnet mit dem Wert *Open*. Voreinstellung für *Initial_Value* ist *Closed*.



Der Stellungswert am Ausgang *Y* wird als Prozentwert mit Hilfe einer Rampenfunktion gebildet:

$0 \leq Y \leq 100$

Der Wert null entspricht dem geschlossenen Ventil, der Wert einhundert dem geöffneten Ventil.



Die Öffnungs- und Schließzeiten des Magnetventils sind mit null vorbelegt, d. h. das Öffnen und Schließen erfolgt sprungförmig.

Die Komponente *Valve* besitzt keine Stromüberwachungsfunktion und damit auch kein Überlastverhalten, sowie weder Stromeingang noch Stromausgang.

8.1.5 Sensorkomponenten

8.1.5.1 SIWAREXU-Komponenten

Das Wägesystem SIWAREX U wird u.a. zur Messung von Füllständen an Silos und Bunkern, bei der Überwachung von Kranlasten und bei der Überlastsicherung von industriellen Aufzügen eingesetzt. In allen diesen Anwendungen werden mit Sensoren wie Wägezellen oder Kraftaufnehmern Gewichte erfasst und als Messwerte an die Steuerung übertragen. Druckempfindliche Sensoren liefern dazu eine dem Gewicht proportionale Spannung, die über einen Analog/Digitalwandler in einen Zahlenwert gewandelt, aufbereitet und an die Steuerung gegeben wird. Wandlung und Aufbereitung des Spannungssignals erfolgt mit Hilfe des SIWAREX U-Moduls. SIWAREX U-Module können über ET 200M an den PROFIBUS DP angeschlossen werden oder als Modul der SIMATIC S7-300 eingesetzt werden.

Ziel der Simulation mit *SIWAREXU*-Komponenten ist es, Messwerte, die in der realen Anlage mit dem SIWAREX U-Wägesystem an die Steuerung übertragen werden, als Werte einer Simulation an die Steuerung zu übertragen. Im Ordner *SIWAREX* der Basisbibliothek sind zwei Komponententypen zur Simulation des Wägemoduls SIWAREX U enthalten:

- *SIWAREXU1*
- *SIWAREXU2*

Mit diesen beiden Komponententypen werden Grundfunktionen der ein- und zweikanaligen Variante des Wägemoduls SIWAREX U nachgebildet. Die Funktion ist für beide Komponententypen *SIWAREXU1* und *SIWAREXU2* gleich. Der Typ *SIWAREXU2* enthält im Vergleich zum Typ *SIWAREXU1* einen weiteren Messkanal. Die folgenden Detailbeschreibungen für einen Kanal des Typs SIWAREXU1 gelten damit entsprechend für beide Kanäle des Typs *SIWAREXU2*-Komponententyps.

Symbol

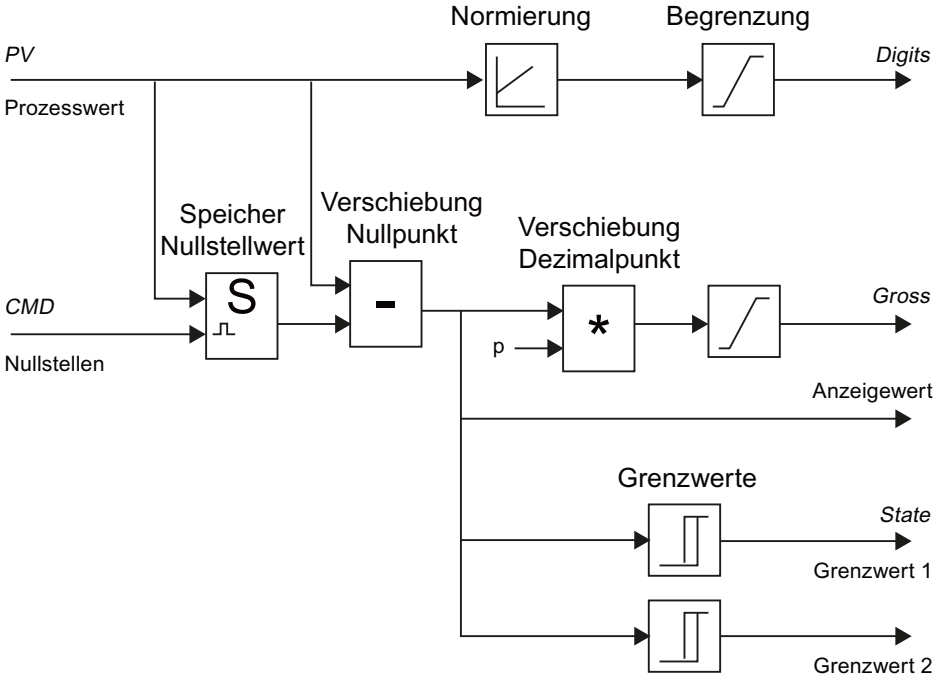


Funktion

Die SIWAREX U-Komponenten simulieren die folgenden Funktionen des Wägesystems:

- lineare Kennlinie des Wägesystems (Justiergerade),
- Nullpunktverschiebung,
- Dezimalpunktverschiebung,
- zwei parametrierbare Grenzwerte.

Ein Funktionsdiagramm für einen Messkanal des Komponententyps ist in der folgenden Abbildung dargestellt.



Im Wägesystem SIWAREX U wird das zu messende Gewicht zunächst über einen Drucksensor in eine elektrische Spannung und dann mit Hilfe eines Analog/Digitalwandlers in einen Zahlenwert gewandelt. In der Simulation liegt der Zahlenwert eines Gewichts bereits als Ergebnis einer Modellberechnung vor. Die elektrische Signalübertragung zwischen Wägesensor und SIWAREX U-Modul ist daher für die Simulation unerheblich: Der berechnete Gewichtswert wird direkt am Eingang *PV* einer *SIWAREXU*-Komponente als physikalische Messgröße angelegt.

Die im SIWAREX U-Modul enthaltene Tiefpassfilterung und gleitende Mittelwertbildung sind als Maßnahmen zur Unterdrückung von Störungen des elektrischen Spannungssignals in der

Simulation nicht erforderlich. Beide Funktionen sind daher nicht in den Komponententypen enthalten.

In *Digits* steht der Gewichtswert als normierter Zahlenwert, begrenzt auf den Bereich von null bis 65.535 zur Verfügung. Die Normierung erfolgt mit Hilfe der linearen Kennlinie des Wägesystems (Justiergeraden). Des Weiteren wird der Gewichtswert nach eventueller Dezimalpunkt- und Nullpunktverschiebung in *Gross* als ganzzahliger (Integer) Wert begrenzt auf den Bereich von −32.768 bis +32.767 gebildet.

Siehe auch

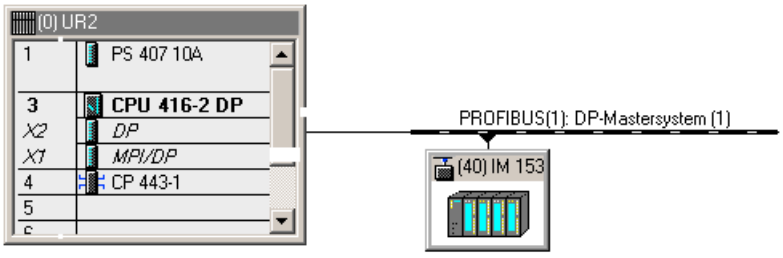
Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

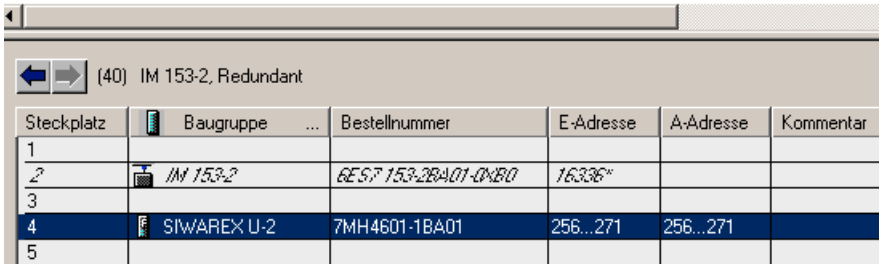
Verknüpfung der SIWAREXU-Komponenten mit der Kopplung

Die beiden Komponententypen *SIWAREXU1* und *SIWAREXU2* sind auf die Kommunikationsart SFC/SFB/FB für SIMATIC S7/PCS 7 abgestimmt. Die gesamte Kommunikation zwischen Steuerung und *SIWAREXU*-Komponenten erfolgt damit wie in der realen Anlage ausschließlich über Datensätze. Die *SIWAREXU*-Komponenten besitzen deshalb auch keine Ein-/Ausgänge zur Verbindung mit Signalen in der Kopplung. Sie werden mit den Datensätzen verknüpft.

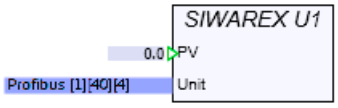
Die Datensatzkommunikation wird von den Kopplungstypen "SIMIT Unit" und "Virtual Controller" verwendet. Wenn Sie in einen solchen Kopplungstyp eine Hardware-Konfiguration importieren, die SIWAREX U-Module enthält, werden für jedes SIWAREX U-Modul automatisch die für die Kommunikation mit der Steuerung relevanten Datensätze in der Kopplung erzeugt. Diese Datensätze werden mit Hilfe des *Unit*-Konnektors mit *SIWAREXU*-Komponenten verknüpft.

In der folgenden Abbildung ist beispielhaft die SIMATIC-Projektierung für ein SIWAREX U-Modul gezeigt: Der Slave mit der PROFIBUS-Adresse 40 am Subsystem 1 enthält ein SIWAREX U-Modul im Steckplatz (Slot) 4.





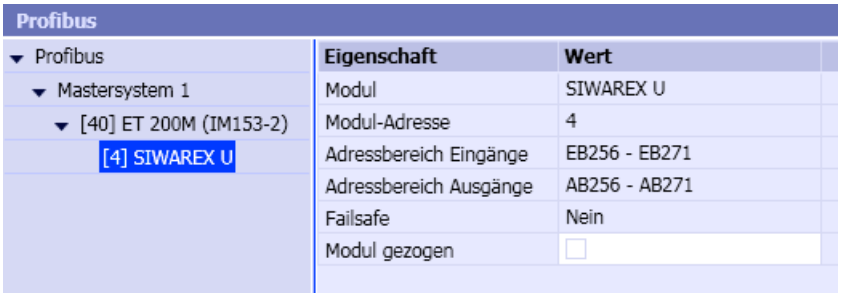
Die zugeordnete *SIWAREXU*-Komponente ist – wie in der folgenden Abbildung dargestellt – an ihrem *Unit*-Eingang mit einem *Unit*-Konnektor versehen.



Sie verknüpfen die Komponente mit den Datensätzen in der Kopplung einfach dadurch, dass Sie im Eigenschaftsfenster des Unit-Konnektors den Namen der *Kopplung* und unter Adressierung die Subsystem-Nummer, die Slave-Nummer und Slot-Nummer in der Form [#][#][#] eintragen, für das skizzierte Beispiel also [1][40][4] (s. folgende Abbildung).



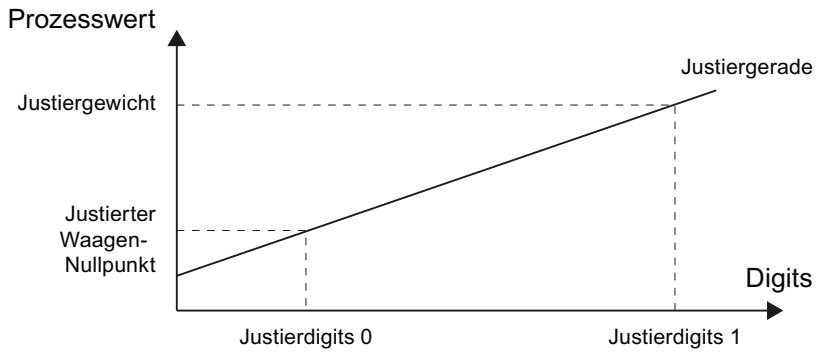
Eine zweite Möglichkeit, die *SIWAREX*-Komponente mit den Datensätzen zu verknüpfen, wird über die SIMIT Unit-Kopplung selbst zur Verfügung gestellt. Öffnen Sie dazu die SIMIT Unit-Kopplung und das Eigenschaftsfenster der Kopplung und selektieren Sie im linken Navigationsbereich das mit der Komponente zu verknüpfende SIWAREX U-Modul durch einen Klick mit der linken Maustaste auf den Moduleintrag.



Ziehen Sie per "Drag & Drop", d. h. mit gedrückter linker Maustaste, das Modul auf das zuvor geöffnete Diagramm, das die Komponente enthält. Das Modul wird dort als *Unit*-Konnektor instanziiert, der direkt mit dem Eingang *Unit* der *SIWAREXU*-Komponente verbunden werden kann. Der *Unit*-Konnektor wird dabei automatisch mit der Adressierung des Moduls versehen.

Justierung

Die Umrechnung eines Gewichtswertes in einen normierten *Digit*-Wert erfolgt mit Hilfe einer durch zwei Punkte definierten Justiergeraden.



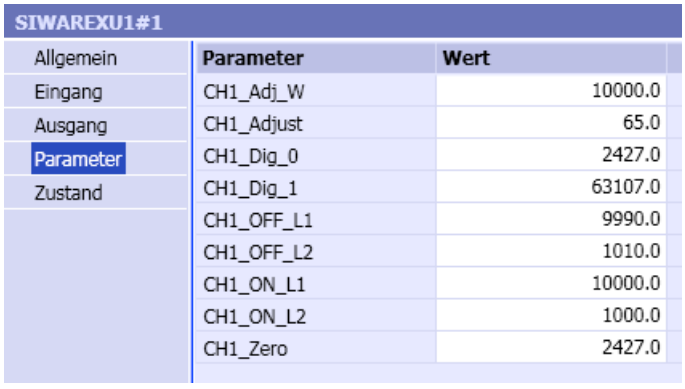
Der erste Justierpunkt ist durch die unbelastete (leere) Waage allein mit ihrem Konstruktionsgewicht bestimmt, der zweite Punkt ist durch das gewählte Justiergewicht bestimmt. Die entsprechenden normierten Zahlenwerte werden als Justierdigits 0 und Justierdigits 1 im Wägemodul gespeichert.

In der Simulation erfolgt die Justage durch das Setzen der Justageparameter. Folgende Tabelle enthält eine Aufstellung der Justierparameter der *SIWAREXU*-Komponententypen und deren Vorbelegung.

Tabelle 8-36 Voreinstellung der Justierparameter

Parameter			Vorbelegung
CH1_Dig_0	CH2_Dig_0	Justierdigits 0	63107
CH1_Dig_1	CH2_Dig_1	Justierdigits 1	2427
CH1_Adj_W	CH2_Adj_W	Justiergewicht	10000

Die Parameter für die Justierdigits sind mit den Werten vorbelegt, die einer Waage mit "theoretischem Abgleich" ohne Nullpunktverschiebung entsprechen. Das Justiergewicht ist mit einem Wert von 10000 Gewichtseinheiten vorbelegt.



Da die mit den *SIWAREXU*-Komponententypen simulierten Waagen über das Setzen von stets gültigen Parametern justiert werden, gelten die simulierten Waagen immer als justiert. Ein von der Steuerung initiierter Justiervorgang einer Waage ist damit in der Simulation nutzlos. Entsprechende Befehle der Steuerung werden von den *SIWAREXU*-Komponenten ignoriert.

Nullpunktverschiebung

Wenn eine *SIWAREXU*-Komponente über das Steuerwort *CMD* den Befehl "Nullstellen" erhält, wird der in diesem Moment anliegende Gewichtswert als neuer Nullpunkt gespeichert. Als Bruttowert wird dementsprechend jetzt der Wert Null ausgegeben. Alle folgenden Messungen sind dann auf diesen Wert bezogen, d. h. es wird dann als Bruttowert die Differenz zwischen anliegendem Gewichtswert und dem zuletzt gespeicherten Nullstellwert ausgegeben.

Der beim Starten der Simulation verwendete Nullstellwert kann als Digit-Wert im Parameter *CH1_Zero* der Komponente eingestellt werden. Voreingestellt ist ein Digit-Wert von 2427, entsprechend einem mit null angenommenen Gewichtswert der unbelasteten Waage.

Dezimalpunktverschiebung

Um die Auflösung des an die Steuerung zu übertragenden Bruttowertes zu erhöhen, kann eine Dezimalpunktverschiebung projektiert werden. Dies führt dazu, dass der gemessene Bruttowert mit einem Faktor 10, 100, 1000, 10.000 oder 100.000 multipliziert wird.

Die Dezimalpunktverschiebung wird durch den Parameter *CH1_Adjust* entsprechend der folgenden Tabelle definiert. Voreingestellt ist hier ein Wert von 65, d. h. keine Dezimalpunktverschiebung.

Tabelle 8-37 Einstelltabelle für die Dezimalpunktverschiebung

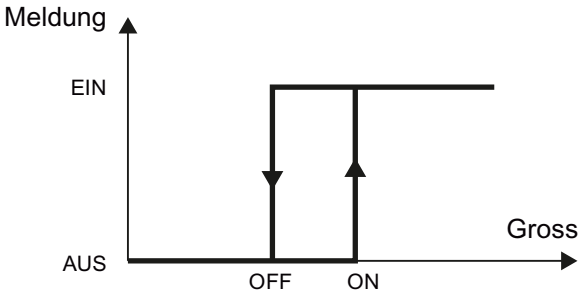
Anzahl Nachkommastellen	Faktor	Parameter CH1_Adjust
0	1	65
1	10	69
2	100	73
3	1000	77
4	10.000	81
5	100.000	85

Hinweis

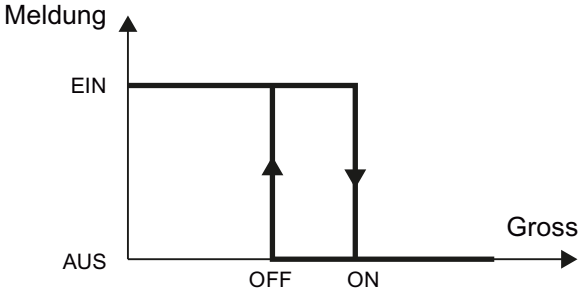
Wenn Sie die Voreinstellung des Parameters *CH1_Adjust* in SIMIT ändern, müssen Sie auch den Parameter C1ADJUST im Instanz-Datenbaustein der Steuerung entsprechend anpassen.

Grenzwerte

Die SIWAREX U verfügt über zwei einstellbare Grenzwerte, deren Ein- und Ausschaltpunkte in Gewichtseinheiten frei vorgegeben werden können. Die Einstellung "Einschaltwert > Ausschaltwert" führt zur Meldung bei Überschreitung eines Gewichtwerts:



Die Einstellung "Ausschaltwert > Einschaltwert" führt zur Meldung bei Unterschreitung eines Gewichtswertes:



Für den Sonderfall, dass die Grenzwerte für *OFF* und *ON* gleichgesetzt werden, meldet Grenzwert 1 eine Überschreitung dieses Wertes, Grenzwert 2 meldet eine Unterschreitung des eingestellten Wertes.

Die Grenzwerte werden mit den Parametern *CH1_ON_L1* und *CH1_OFF_L1* für den ersten Grenzwert und mit den Parametern *CH1_ON_L2* und *CH1_OFF_L2* für den zweiten Grenzwert vorgegeben. Voreingestellt sind die Werte gemäß der folgenden Tabelle.

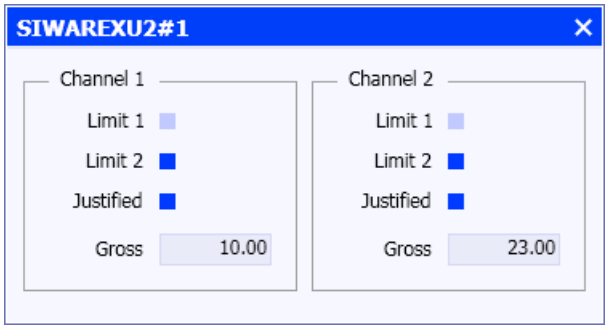
Tabelle 8-38 Voreinstellung der Grenzwerte

Parameter		Vorbelegung
CH1_ON_L1	CH2_ON_L1	10000
CH1_OFF_L1	CH2_OFF_L1	9900
CH1_ON_L2	CH2_ON_L2	1000
CH1_OFF_L2	CH2_OFF_L2	1010

Der Zustand der Grenzwerte wird als Teil der Statusinformation *State* des Komponententyps geführt.

Bedienfenster der SIWAREXU-Komponenten

Im Bedienfenster einer *SIWAREX U*-Komponente wird angezeigt, welche Grenzwerte angesprochen haben und welcher Bruttowert als Gewichtswert an die Steuerung übertragen wird. Der Bruttowert (*Gross*) wird der besseren Lesbarkeit wegen ohne die Dezimalpunktverschiebung als *Anzeigewert* des Funktionsdiagramms angezeigt.

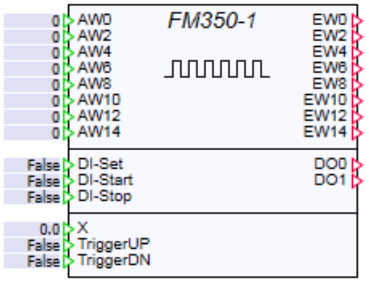


Die Statusanzeige *Justified* ist immer aktiv. Es wird damit darauf hingewiesen, dass die simulierte Waage immer justiert ist.

8.1.5.2 Zählerbaugruppe FM350-1

Funktionsweise der FM350-1

Symbol



Funktion

Die Funktionsbaugruppe FM 350-1 ist eine schnelle Zählerbaugruppe für den Einsatz im Automatisierungssystem S7-300 oder als Modul an einer ET 200M-Station. Nur dezentral am PROFIBUS/PROFINET eingesetzte Baugruppen können mit SIMIT simuliert werden.

Die Baugruppe hat folgende Betriebsarten:

- Zählbetriebsarten
  - Endlos Zählen (Continuous counting)
  - Einmaliges Zählen (Single counting)
  - Periodisches Zählen (Periodic counting)
- Messbetriebsarten
  - Frequenzmessung (Frequency measurement)
  - Drehzahlmessung (RPM measurement)
  - Periodendauermessung (Continuous periodic measurement)

Alle notwendigen Funktionen dieser Betriebsarten werden in SIMIT nachgebildet.

Die Daten der realen FM350-1 werden zyklisch über PROFIBUS DP übertragen. Das Quittierungsprinzip wurde in die SIMIT-Komponente übernommen.

Weitere Informationen finden Sie in der Dokumentation "*S7-300; Zählerbaugruppe FM 350-1*".

Auf einzelne Funktionen wird hier nicht im Detail eingegangen, sondern lediglich auf die Unterschiede hingewiesen und das Grundverhalten der simulierten Komponente beschrieben.

Verhalten der Ein- und Ausgänge in der simulierten Baugruppe

Das Simulationssystem SIMIT basiert auf zyklischer Berechnung mathematischer Gleichungssysteme. Die schnelle Zählfunktion der FM350 wird in SIMIT dadurch simuliert, dass vereinfachend gesagt, die Frequenz der Zählimpulse vorgegeben wird und diese entsprechend in jedem Rechenzyklus zykluszeitabhängig auf den aktuellen Zählstand hinzuaddiert oder subtrahiert wird. Durch das Vorzeichen der Frequenz kann dadurch auch rückwärts gezählt werden. Der Eingang X ist in diesem Fall als "Zählimpulse pro Sekunde" zu verstehen.

Für langsame Zählimpulse kann auch jede einzelne Flanke ausgewertet werden, wenn die Impulse in kürzerer Folge, als die doppelte Zykluszeit eintreffen. Wenn der Eingang X Null ist und an einem der Triggereingänge eine positive Flanke erkannt wird, dann wird der interne Zählwert um eins inkrementiert bzw. dekrementiert.

Die Anschlüsse "CNT", "TriggerUP" und "TriggerDN" dienen zur Veränderung des internen Zählwertes, wenn das interne Tor geöffnet ist.

Der Eingang "CNT" kann in den Zählbetriebsarten direkt als Zählimpulsfrequenz interpretiert werden, und entspricht in den Messbetriebsarten direkt dem gemessenen Wert. Zusätzlich kann über die beiden Triggereingänge der Zählerstand mit jeder positiven Flanke um eins inkrementiert oder dekrementiert werden.

Das Verhalten der restlichen digitalen Eingänge und Ausgänge ist identisch mit der realen Baugruppe.

Hinweis

Prozessalarme werden von der simulierten Baugruppe nicht unterstützt.

Datenlesen über DS0/1 (Diagnosedatensatz) wird in der simulierten Baugruppe nicht unterstützt.

Taktsynchronität wird in der simulierten Baugruppe nicht unterstützt, d. h. die Synchronisation des Funktionsmoduls über den PROFIBUS kann aufgrund der Trennung zwischen physikalischer Nachbildung der Busphysik durch die SIMIT Unit und Nachbildung der Logik in dieser Simulationskomponente nicht realisiert werden.

Betriebsarten der FM350-1

Die Zählbetriebsarten

In den Zählbetriebsarten wird der Messwert durch Aufsummieren der am Eingang X anstehenden Impulse berechnet. Bei sehr langsamen Zählfolgen oder einzelnen Flanken ist es möglich mit einer positiven Flanke am Eingang "TriggerUP" den Zähler um eins zu erhöhen, bzw. am Eingang "TriggerDN" zu verringern. Dadurch ist es möglich, sehr hohe Frequenzen von Zählimpulsen zu simulieren, aber auch das Zählen einzelner Impulse zu ermöglichen.

Zählgrenzen

Für den 31-bit-Zählbereich gilt:

- Die obere Zählgrenze ist festgelegt auf +2 147 483 647 ($2^{31}-1$).
- Die untere Zählgrenze ist festgelegt auf −2 147 483 648 (-2^{31}).

Für den 32-bit-Zählbereich gilt:

- Die obere Zählgrenze ist festgelegt auf +4 294 967 295 ($2^{32}-1$).
- Die untere Zählgrenze ist festgelegt auf 0 (Null).

Endlos Zählen

In dieser Betriebsart zählt die simulierte FM 350-1 ab dem Zählerstand endlos:

- Erreicht der Zähler beim Vorwärtszählen die obere Zählgrenze und kommt ein weiterer Zählimpuls, springt er auf die untere Zählgrenze und zählt von dort ohne Impulsverlust weiter.
- Erreicht der Zähler beim Rückwärtszählen die untere Zählgrenze und kommt ein weiterer Zählimpuls, springt er auf die obere Zählgrenze und zählt von dort ohne Impulsverlust weiter.

Einmaliges Zählen

Durch Erreichen der durch die Hauptzählrichtung festgelegten Zählgrenze wird intern automatisch das Tor geschlossen und muss danach erst durch erneuten Torstart gestartet werden.

Periodisches Zählen

Durch Erreichen der durch die Hauptzählrichtung festgelegten Zählgrenze wird intern auf den Ladewert gesprungen und von dort aus weitergezählt.

Die Messbetriebsarten

Bei den Messbetriebsarten wird am Eingang "CNT" direkt der gemessene Wert angelegt. In diesen Betriebsarten ist der Messwert gleich dem Zählerstand. Auch hier kann der Zählwert mit positiven Flanken an den "TriggerUP"/"TriggerDN" Eingängen um jeweils 1 verändert werden.

Die Aktualisierungszeit spielt bei der simulierten Zählerbaugruppe keine Rolle. Der anliegende Wert wird über die Rückmeldeschnittstelle an die Steuerung übertragen. Grenzwertüberwachung und Torsteuerung sind implementiert.

Frequenzmessung

Die gemessene Frequenz ist gleich der am Eingang "CNT" anliegenden Wert.

Hinweis

Der Wert der realen FM350-1 wird hier in der Einheit 10^{-3} Hz übertragen und muss in der Simulation entsprechend vorgegeben werden.

Drehzahlmessung

Die gemessene Drehzahl ist gleich der am Eingang "CNT" anliegende Wert

Hinweis

Der Wert der realen FM350-1 wird hier in der Einheit 10^{-3} /min übertragen und muss in der Simulation entsprechend vorgegeben werden.

Periodendauermessung

Die gemessene Periodendauer ist gleich der am Eingang "CNT" anliegenden Wert.

Hinweis

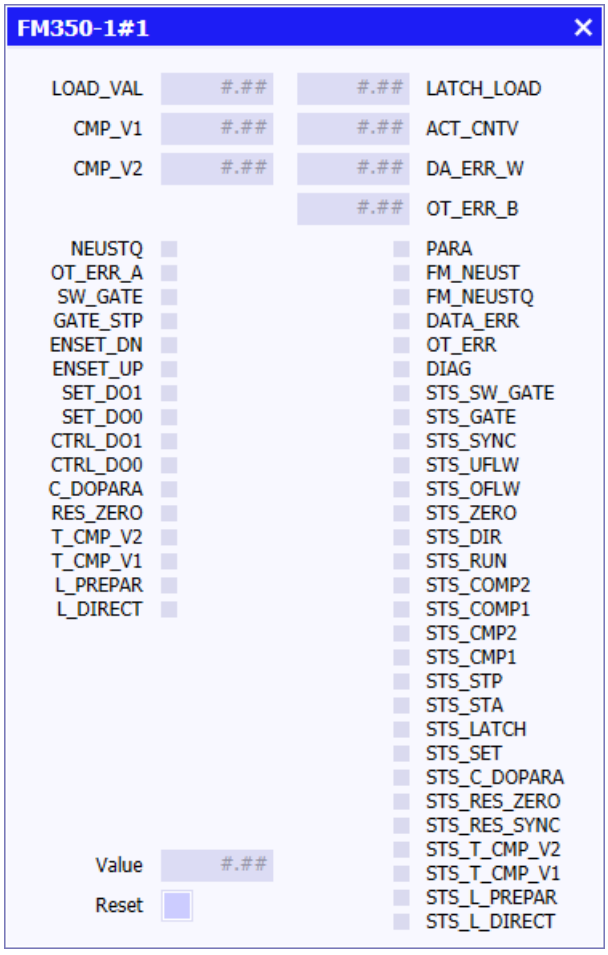
Der Wert der realen FM350-1 wird hier in der Einheit 1µs oder 1/16µs übertragen und muss in der Simulation entsprechend vorgegeben werden.

Weitere Informationen zum Zählbetrieb und zum Messbetrieb finden Sie in der Dokumentation "*S7-300; Zählerbaugruppe FM 350-1*".

Bedienfenster der Komponente FM-350-1

Im Bedienfenster werden links die Signale der Steuerschnittstelle und rechts die Signale der Rückmeldeschnittstelle angezeigt. Es werden immer alle Signale angezeigt, unabhängig von der eingestellten Betriebsart.

Die Anordnung der Signale entspricht der realen Baugruppe.

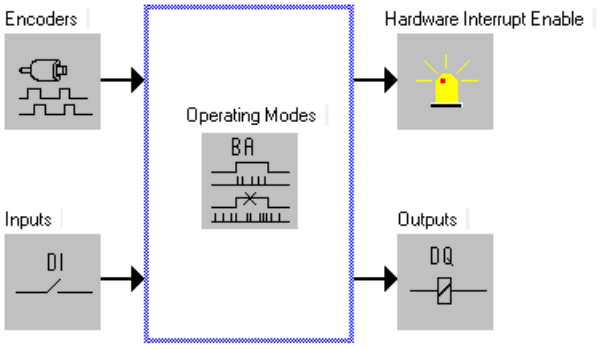


Mit dem Taster "Reset" kann die Zählerbaugruppe manuell rückgesetzt werden. Danach muss eine Neustartkoordinierung durchlaufen werden.

Weitere Information finden Sie in der Dokumentation "*S7-300; Zählerbaugruppe FM 350-1*".

Eigenschaften der Komponente FM-350-1

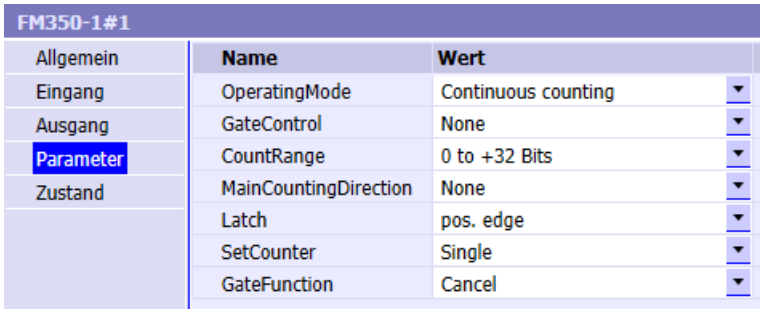
Die Parameter der simulierten Baugruppe entsprechen den Einstellungen in der HW-Konfig in STEP 7. Die Benennung der Parameter und die Auswahlmöglichkeiten entsprechen den englischen Notationen in der HW-Konfig.



Diese Parameter müssen für die jeweilige FM350-1 Instanz angepasst werden, es wird kein automatischer Import aus HW-Konfig durchgeführt.

Eigenschaften der FM 350-1

Die folgende Abbildung zeigt das Eigenschaftsfenster mit ausgewähltem Objekt "Parameter" in der Voreinstellung:



Die einzelnen Parameter und ihre Auswahlmöglichkeiten aus den Klapplisten werden in den folgenden Abschnitten beschrieben.

Parameter "OperatingMode"

Dieser Parameter legt die Betriebsart fest. Er muss auf denselben Wert wie in HW-Konfig in STEP 7 eingestellt sein.

Folgende Funktionen können aus der Klappliste ausgewählt werden:

- Endlos Zählen (Continuous counting)
- Einmaliges Zählen (Single counting)
- Periodisches Zählen (Periodic counting)
- Frequenzmessung (Frequency measurement)
- Drehzahlmessung (RPM measurement)
- Periodendauermessung (Continuous periodic measurement)

Parameter "GateControl"

Dieser Parameter bestimmt die Torsteuerung der Zählerbaugruppe. Mit der Torsteuerung werden die Zählvorgänge gestartet und gestoppt.

Folgende Funktionen können aus der Klappliste ausgewählt werden:

- "None": Tor ist immer offen
- "SW-Gate": Torzustand ist abhängig vom Signal "SW_GATE" der Steuerschnittstelle
- "HW-Gate (Level controlled)": Torzustand ist abhängig vom Eingang "DI_Start" (pegelgesteuert)
- "HW-Gate (Edge-Controlled)": eine positive Flanke am Eingang "DI_Start" öffnet das Tor, eine positive Flanke am Eingang "DI_Stop" schließt es
- "Latch": Voraussetzung für das Speichern von Zählerständen
- "Latch/Retrigger": Voraussetzung für das Speichern von Zählerständen mit anschließendem Setzen auf den Ladewert und Weiterzählen vom Ladewert aus

Parameter "CountRange"

Dieser Parameter stellt den Zählbereich ein. Arbeitet die Zählerbaugruppe im Messbetrieb, hat diese Einstellung keine Auswirkung.

Folgende Zahlenbereiche können aus der Klappliste ausgewählt werden:

- "0 to +32 Bits"
- "−31 to +31 Bits"

Diese Einstellung hat Auswirkungen auf die Zählgrenzen und Vorzeichen der übertragenden Werte.

Parameter "MainCountingDirection"

Dieser Parameter bestimmt die Hauptzählrichtung. Die Hauptzählrichtung ist nur in den Zählbetriebsarten relevant.

Folgende Funktionen können aus der Klappliste ausgewählt werden:

- "None": keine Zählrichtung ist festgelegt
- "Forward": vorwärts zählen
- "Backward": rückwärts zählen

Parameter "Latch"

Dieser Parameter stellt die Voraussetzung für das Speichern der Zählerstände ein.

- "pos. edge": Der Zählerstand wird bei positiver Flanke am Eingang "DI-Start" gespeichert
- "neg. edge": Der Zählerstand wird bei negativer Flanke am Eingang "DI-Start" gespeichert
- "both edge": Der Zählerstand wird bei positiver oder negativer Flanke am Eingang "DI-Start" gespeichert

Parameter "SetCounter"

Dieser Parameter stellt das Verhalten zum Setzen auf den Ladewert auf einmaliges oder mehrfaches Setzen.

- "Single"
- "Multiple"

Parameter "GateFunction"

Dieser Parameter stellt die Torsteuerung auf "Abbrechend" oder "Unterbrechen".

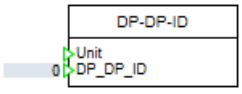
- "Cancel"
- "Interrupt"

8.1.6 Kommunikationskomponenten

8.1.6.1 Komponenten für SIMATIC

DP-DP-ID

Symbol



Funktion

Der Komponententyp *DP-DP-ID* dient der Simulation einer fehlersicheren Master-Master-Kommunikation (Profisafe V1) mit SIMIT und SIMIT Unit. Dabei wird jeweils das Gegenstück eines realen, in der CPU projektierten F_SENDDP/F_RCVDP-Bausteins mit SIMIT und SIMIT Unit simuliert. Die SIMIT UNIT PROFIBUS oder PROFINET simuliert dabei den projektierten DP/DP-Koppler bzw. PN/PN-Koppler mit den jeweiligen Universalmodulen. Die F-Kommunikation wird über die Nutzdaten der Module nachgebildet.

Der Import der Hardware in SIMIT funktioniert wie bei einem Standard-Projekt. Es muss nur in der HW-Konfig des S7-Projektes darauf geachtet werden, dass die Universalmodule im DP/DP-Koppler oder PN/PN Koppler die richtige Länge für die F-Kommunikationsbausteine besitzen.

```
+----------+ ----12 Bytes----> +----------+
| F_SENDDP |                   | F_RECVDP |
+----------+ <----6 Bytes----- +----------+
```

Wird eine falsche Länge projektiert (z. B. ein größerer Bereich), erkennt SIMIT das Modul als Standard-E/A-Modul.

Im SIMIT-Projekt wird automatisch eine Funktion PROFIsafe_V1 eingebaut. Jeder F_SENDDP/F_RCVDP-Baustein in der realen CPU hat in SIMIT das passende Gegenstück.

Bei der Simulation des F_RCVDP müssen lediglich die DP_DP_ID und die Geräteadresse in Form des Unit-Konnektors stimmen.

Bei der Simulation des Senderbausteines F_SENDDP müssen die DP_DP_ID und die Geräteadresse in Form des Unit-Konnektors angegeben werden. Sind die Angaben vor dem CPU-Start korrekt, ist die Verbindung nach dem Starten der CPU erfolgreich aufgebaut. Falls die CPU schon vorher lief, muss der F_RCVDP-Baustein in der CPU über den Eingang ACK_REI wiedereingegliedert werden.

ACHTUNG
Im Fall einer Unterbrechung der Verbindung (z. B. durch Ziehen des PROFIBUS-Kabels), muss die CPU zum Wiedereingliedern neu gestartet werden! Anderenfalls setzt sie ihre internen Telegrammzähler nicht zurück.

Eingänge

Unit

Am Eingang Unit ist über einen Unit-Konnektor eine Baugruppe angeschlossen. In einen Datensatz der am Eingang Unit verbundenen Baugruppe trägt der Komponententyp *DP-DP-ID* die DP-DP-ID ein.

DP_DP_ID

Die DP-DP-ID ist eine systemweit eindeutige Verbindungs-ID. Dieser Parameter muss identisch mit dem gleichnamigen Parameter des Partnerbausteins in der CPU sein.

Beispiel

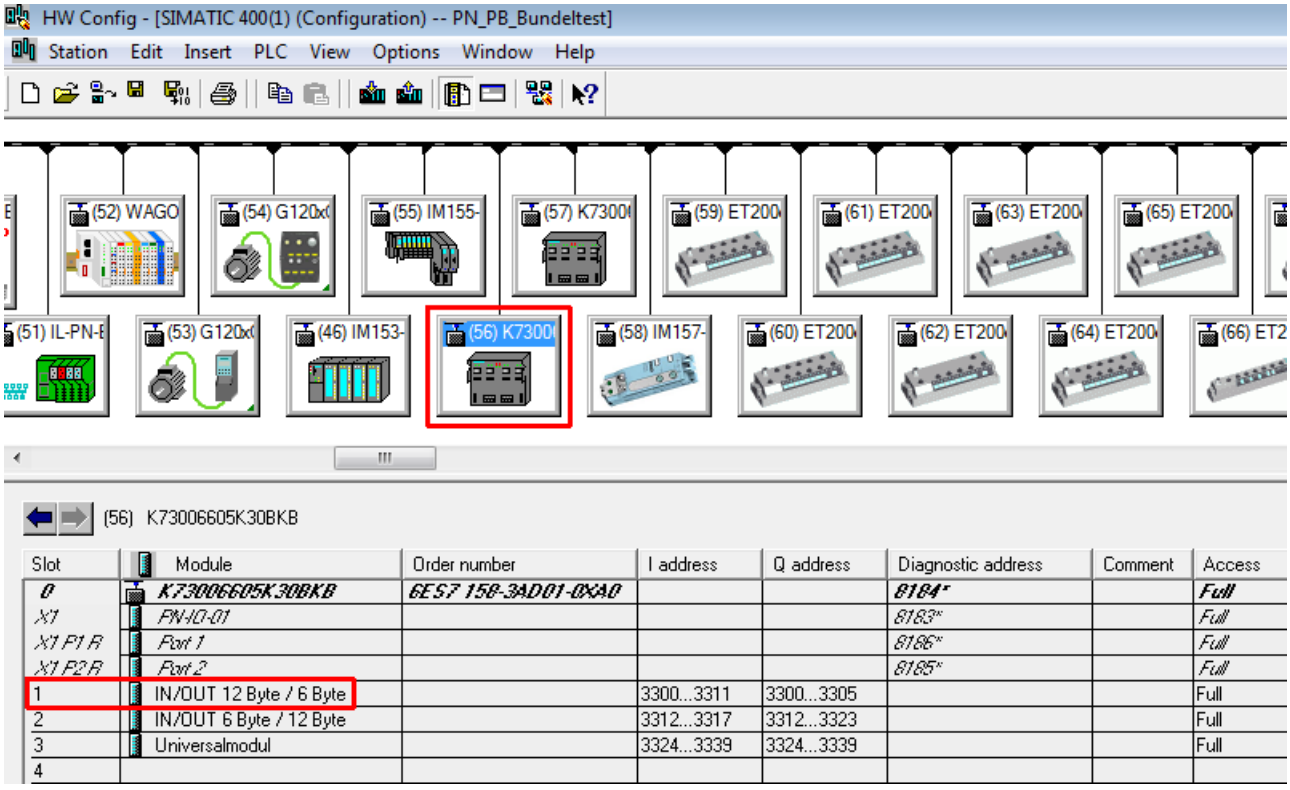


Bild 8-1 Hardware-Konfiguration: Markiert ist Slot 1 des Geräts 56 im PROFINET

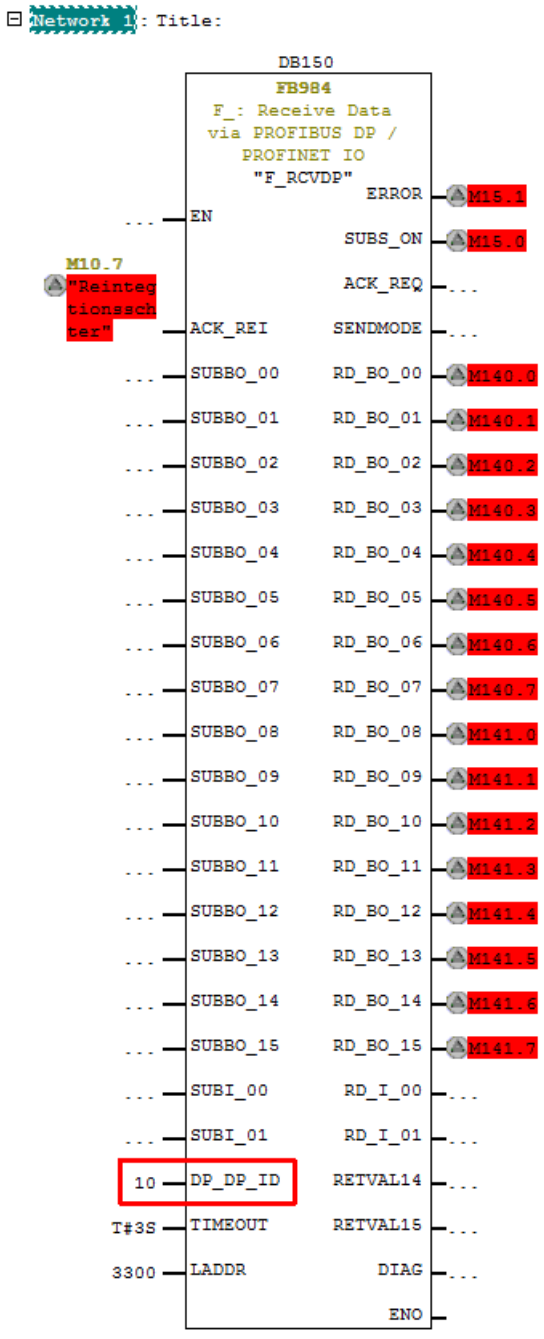


Bild 8-2 F_RCVDP: Die DP-DP-ID hat den Wert 10

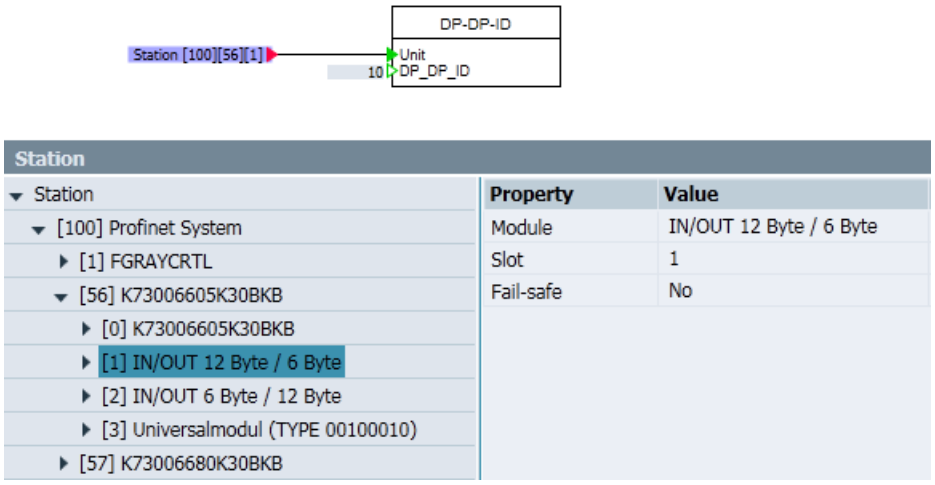


Bild 8-3 DP-DP-ID-Komponente in SIMIT: Über den Unit-Konnektor ist Slot 1 des Geräts 56 im PROFINET mit dem Eingang Unit verbunden. Am Eingang DP_DP_ID ist der gleiche Wert 10 eingetragen wie im F_RCVDP.

Multi HART

Symbol

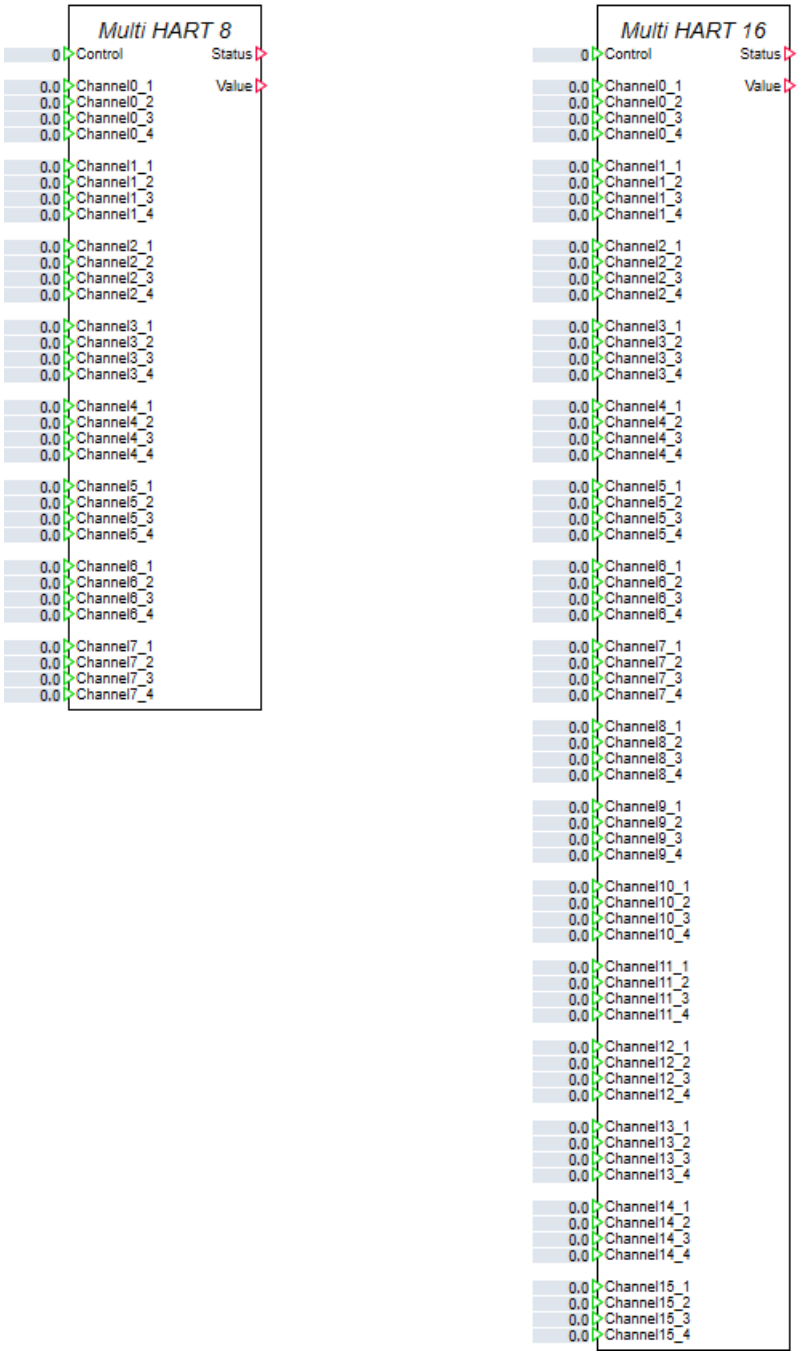


Bild 8-4 Symbol des Komponententyps Multi HART 8 (links) und des Komponententyps Multi HART 16 (rechts)

Funktion

Die Komponententypen *Multi HART 8* und *Multi HART 16* sind Multiplexer für die Hardware-Familie ET200SP HA.

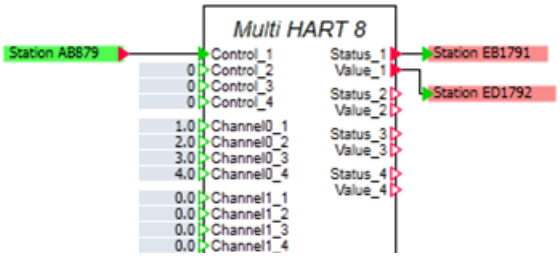
Der Komponententyp *Multi HART 8 / Multi HART 16* schaltet einen der 32 / 64 Eingänge in Abhängigkeit vom Wert des Selektionseingangs Control auf den Ausgang Value. Der Ausgang Status erhält den gleichen Wert wie der Selektionseingang Control.

Der Selektionseingang Control ist mit 0 vorbelegt. Alle anderen Eingänge sind mit 0.0 vorbelegt.

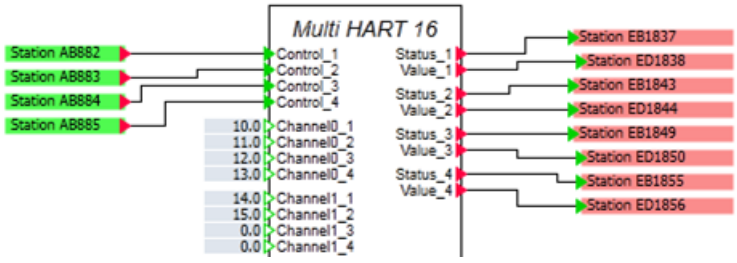
Parametrierung

Die Parametrierung der HART-Nebenvariablen in ET 200SP HA definiert, wie viele Selektionseingänge Sie in SIMIT verschalten müssen:

- Parametrierung als "1 multiHART":



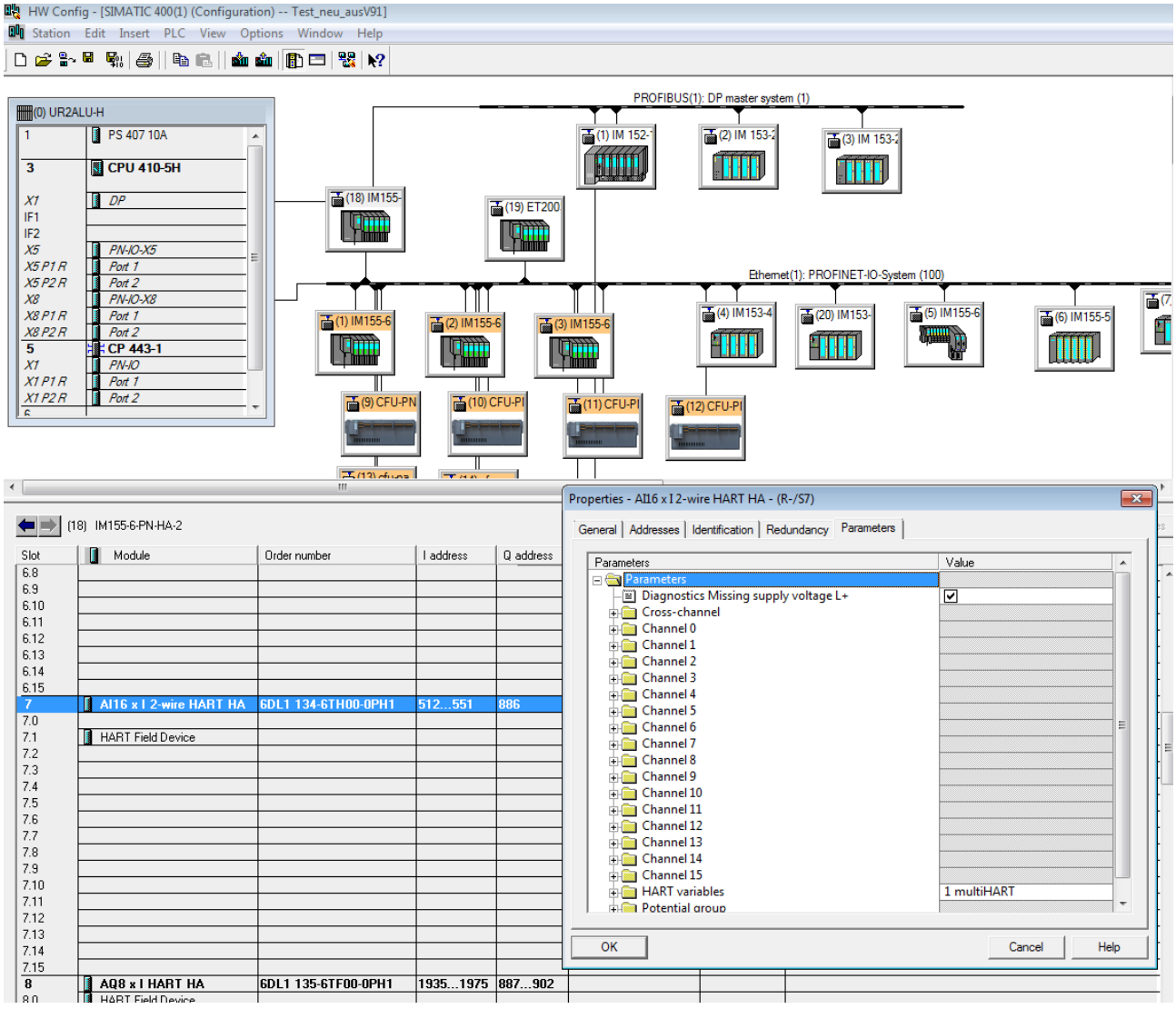
- Parametrierung als "4 multiHART":



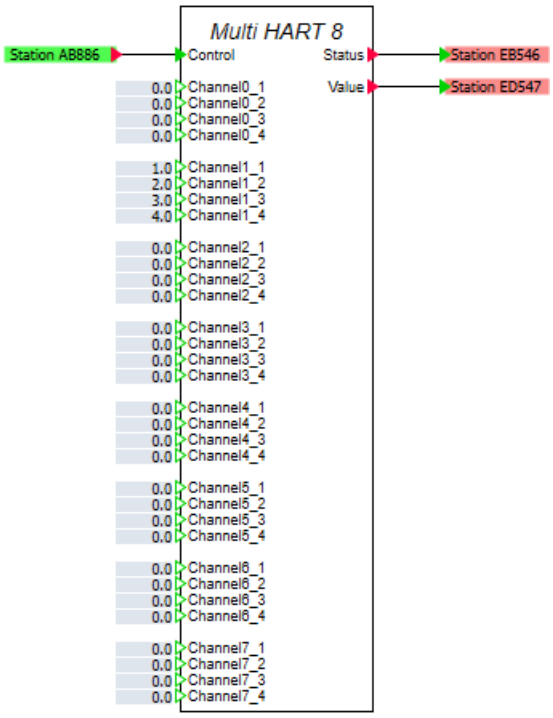
Beispiel

Die Komponente Multi HART dient der Simulation von HART-Nebenvariablen, wenn die HART-Nebenvariablen in ET200SP HA als "1 multiHART" oder "4 multiHART" parametriert sind und der Block Pcs7HaAI im PCS 7-Projekt benutzt wird. Bei der Parametrierung "8 HART" ist die Komponente Multi HART nicht notwendig, denn die Nebenvariablen sind direkt über das Prozessabbild verfügbar.

In folgendem Beispiel werden die vier HART-Nebenvariablen aus Kanal 2 von Modul 7 gelesen.



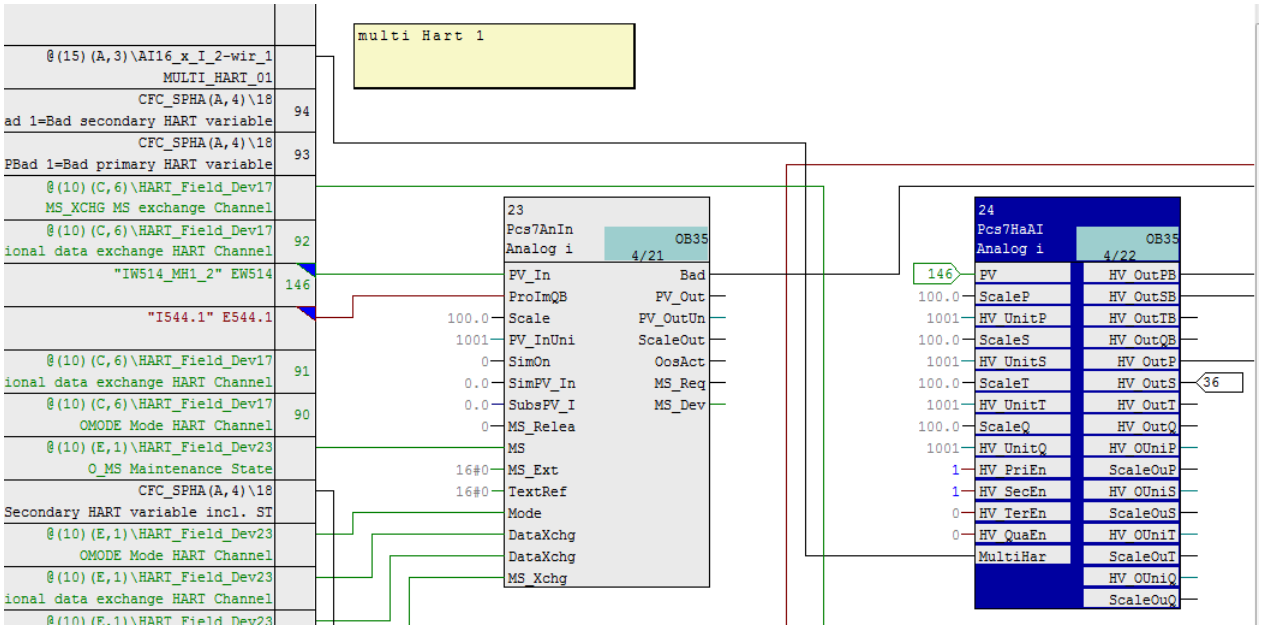
Wie in der Hilfe von ET200SP HA beschrieben ist, steht bei dieser Parametrierung das Control-Byte bei AB886 und das Status-Byte bei EB546. AB886 ist mit dem Eingang Control der Komponente Multi HART verbunden und bestimmt, welche HART-Nebenvariable gerade gelesen wird. Die gerade gelesene HART-Nebenvariable wird an ED547 übergeben.



Im Beispiel wird Control von Pcs7HaAI generiert und folgende Sequenz durchlaufen:

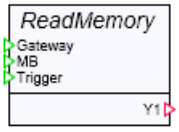
Control = Channel1_1. Dann wird Status = Channel1_1 und Value = 1.0.
Control = Channel1_2. Dann wird Status = Channel1_2 und Value = 2.0.
Control = Channel1_3. Dann wird Status = Channel1_3 und Value = 3.0.
Control = Channel1_4. Dann wird Status = Channel1_4 und Value = 4.0.

In CFC sieht die Projektierung so aus:



ReadMemory – Lesen eines Merkerbereichs

Symbol



Funktion

Der Komponententyp *ReadMemory* erlaubt es, ein oder mehrere aufeinanderfolgende Bytes aus dem Merkerbereich einer Steuerung zu lesen. Am Eingang *MB* ist die Adresse des ersten zu lesenden Byte anzugeben. Der Lesevorgang wird ausgeführt, wenn am Eingang *Trigger* eine steigende Flanke, also ein Wechsel von False zu True stattfindet.

Die Anzahl *N* der Ausgänge ist variabel. Sie kann durch "Aufziehen" der Komponente auf einem Diagramm verändert werden. Sie können maximal 32 Ausgänge einstellen, d. h. Sie können mit einer Komponente dieses Typs maximal 32 Byte lesen. Die gelesenen Bytes stehen an den Ausgängen *Y1* bis *YN* zur Verfügung.

Während der Initialisierung der Simulation wird genau ein Lesevorgang angestoßen. Daher stehen nach der Initialisierung auch ohne Triggersignal Initialwerte aus dem Merkerbereich zur Verfügung.

Die Komponente kann mit folgenden Kopplungen betrieben werden:

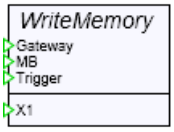
- PLCSIM
- PRODAVE
- Virtual Controller

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

WriteMemory – Schreiben eines Merkerbereichs

Symbol



Funktion

Der Komponententyp *WriteMemory* erlaubt es, ein oder mehrere aufeinanderfolgende Bytes in den Merkerbereich einer Steuerung zu schreiben. Am Eingang *MB* ist die Adresse des ersten zu schreibenden Byte anzugeben. Der Schreibvorgang wird ausgeführt, wenn am Eingang *Trigger* eine steigende Flanke, also ein Wechsel von False auf True stattfindet.

Die Anzahl *N* der Eingänge ist variabel. Sie kann durch "Aufziehen" der Komponente auf einem Diagramm verändert werden. Sie können maximal 32 Eingänge einstellen, d. h. Sie können mit einer Komponente dieses Typs maximal 32 Byte im Merkerbereich schreiben. Die zu schreibenden Bytes sind an den Eingängen *X1* bis *XN* zur Verfügung zu stellen.

Während der Initialisierung der Simulation wird genau ein Schreibvorgang angestoßen. Daher können durch die Initialisierung auch ohne Triggersignal Initialwerte in den Merkerbereich geschrieben werden.

Die Komponente kann mit folgenden Kopplungen betrieben werden:

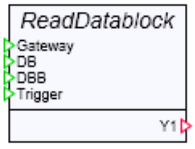
- PLCSIM
- PRODAVE
- Virtual Controller

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ReadDatablock – Lesen eines Datenbausteins

Symbol



Funktion

Der Komponententyp *ReadDatablock* erlaubt es, ein oder mehrere aufeinanderfolgende Bytes aus einem Datenbaustein einer Steuerung zu lesen. Am Eingang *DB* ist die Nummer des Datenbausteins und am Eingang *DBB* die Adresse des ersten zu lesenden Byte anzugeben. Der Lesevorgang wird ausgeführt, wenn am Eingang *Trigger* eine steigende Flanke, also ein Wechsel von False auf True stattfindet.

Die Anzahl *N* der Ausgänge ist variabel. Sie kann durch "Aufziehen" der Komponente auf einem Diagramm verändert werden. Sie können maximal 32 Ausgänge einstellen, d. h. Sie können mit einer Komponente dieses Typs maximal 32 Byte lesen. Die gelesenen Bytes stehen an den Ausgängen *Y1* bis *YN* zur Verfügung.

Während der Initialisierung der Simulation wird genau ein Lesevorgang angestoßen. Daher stehen nach der Initialisierung auch ohne Triggersignal Initialwerte aus dem Datenbaustein zur Verfügung.

Die Komponente kann mit folgenden Kopplungen betrieben werden:

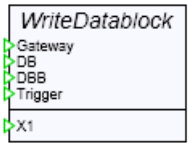
- PLCSIM
- PRODAVE
- Virtual Controller

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

WriteDatablock – Schreiben eines Datenbausteins

Symbol



Funktion

Der Komponententyp *WriteDatablock* erlaubt es, ein oder mehrere aufeinanderfolgende Bytes in den Datenbaustein einer Steuerung zu schreiben. Am Eingang *DB* ist die Nummer des Datenbausteins und am Eingang *DBB* ist die Adresse des ersten zu schreibenden Byte anzugeben. Der Schreibvorgang wird ausgeführt, wenn am Eingang *Trigger* eine steigende Flanke, also ein Wechsel von False auf True stattfindet.

Die Anzahl *N* der Eingänge ist variabel. Sie kann durch "Aufziehen" der Komponente auf einem Diagramm verändert werden. Sie können maximal 32 Eingänge einstellen, d. h. Sie können mit einer Komponente dieses Typs maximal 32 Byte im Datenbaustein schreiben. Die zu schreibenden Bytes sind an den Eingängen *X1* bis *XN* zur Verfügung zu stellen.

Während der Initialisierung der Simulation wird genau ein Schreibvorgang angestoßen. Daher können durch die Initialisierung auch ohne Triggersignal Initialwerte in den Datenbaustein geschrieben werden.

Die Komponente kann mit folgenden Kopplungen betrieben werden:

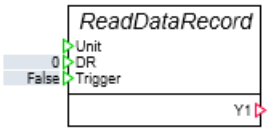
- PLCSIM
- PRODAVE
- Virtual Controller

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ReadDataRecord – Lesen eines Datensatzes

Symbol



Funktion

Der Komponententyp *ReadDataReord* erlaubt es, einen Datensatz zu lesen, der von der Steuerung oder SIMIT VC geschrieben wurde. Am Eingang DR ist die Nummer des zu lesenden Datensatzes anzugeben. Der Lesevorgang wird ausgeführt, wenn am Eingang *Trigger* eine steigende Flanke, also ein Wechsel von False zu True stattfindet.

Die Anzahl *N* der Ausgänge ist variabel. Sie kann durch "Aufziehen" der Komponente auf einem Diagramm verändert werden. Sie können maximal 64 Ausgänge einstellen, d. h. Sie können mit einer Komponente dieses Typs maximal 64 Byte lesen. Die gelesenen Bytes stehen an den Ausgängen *Y1* bis *YN* zur Verfügung.

Während der Initialisierung der Simulation wird genau ein Lesevorgang angestoßen. Daher stehen nach der Initialisierung auch ohne Triggersignal Initialwerte aus dem Merkerbereich zur Verfügung.

Die Komponente kann mit folgenden Kopplungen betrieben werden:

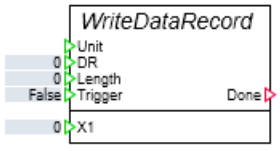
- SIMIT Unit
- Virtual Controller
- PLCSIM Advanced

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

WriteDataRecord – Schreiben eines Datensatzes

Symbol



Funktion

Der Komponententyp *WriteDataRecord* erlaubt es, einen Datensatz zu schreiben, der von der Steuerung oder SIMIT VC gelesen werden kann. Eine Gültigkeitsprüfung für den Datensatz wird nicht ausgeführt. Am Eingang *DR* ist die Nummer des Datensatzes und am Eingang *Length* ist die Länge des Datensatzes anzugeben. Der Schreibvorgang wird ausgeführt, wenn am Eingang *Trigger* eine steigende Flanke, also ein Wechsel von False auf True stattfindet. Wenn Sie nur ein einmaliges Schreiben eines Datensatzes beim Starten der Simulation anstoßen wollen, belegen Sie den Eingang *Trigger* fest mit True vor.

Die Anzahl *N* der Eingänge ist variabel. Sie kann durch "Aufziehen" der Komponente auf einem Diagramm verändert werden. Sie können maximal 64 Eingänge einstellen und mit einer Komponente dieses Typs maximal einen Datensatz mit der Länge 64 Byte schreiben. Die zu schreibenden Bytes sind an den Eingängen *X1* bis *XN* zur Verfügung zu stellen.

Während der Initialisierung der Simulation wird genau ein Schreibvorgang angestoßen. Daher können durch die Initialisierung auch ohne Triggersignal Initialwerte in den Datenbaustein geschrieben werden.

Die Komponente kann mit folgenden Kopplungen betrieben werden:

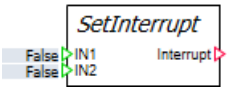
- SIMIT Unit
- Virtual Controller
- PLCSIM Advanced

Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

SetInterrupt

Symbol



Funktion

Der Komponententyp *SetInterrupt* setzt die Hardware-Interrupts in der SIMIT Unit-Kopplung während der Simulation.

Die Anzahl der binären Eingänge ist variabel. Sie kann durch "Aufziehen" der Komponente verändert werden. Sie können maximal 64 Eingänge einstellen, d. h. Sie können mit einer Komponente dieses Typs maximal 64 Hardware-Interrupts an einem Hardware-Objekt der SIMIT Unit-Kopplung auslösen.

Dabei entspricht der Eingang INX dem in der Kopplung angegebenen Alarm [X]. Bei steigender Flanke an INX wird der entsprechende Alarm ausgelöst (Kommend) und bei fallender Flanke

wieder weggenommen (Gehend). Als Interrupt-Ausgang verwenden Sie das entsprechende Signal des Hardware-Objekts aus der SIMIT Unit-Kopplung.

Hinweis

Wenn Sie diese Komponente verwenden, sind die Direkteingaben in der Kopplung nicht mehr möglich. Direkteingaben in der Kopplung werden sofort von den Modellwerten überschrieben.

Siehe auch

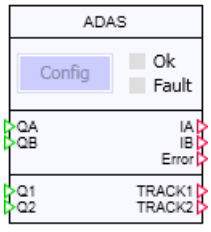
Simulation von Alarmen (Seite 96)

8.1.6.2 Komponenten für SINUMERIK

Einen Komponententyp, der die Übertragung von Achswerten aus SINUMERIK an SIMIT ermöglicht, finden Sie in der Basisbibliothek im Verzeichnis *COMMUNICATION* | *SINUMERIK*.

ADAS – AXIS DATA STREAM PER PROFIBUS

Symbol



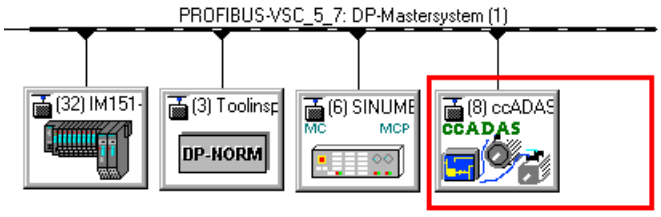
Funktion

Der Komponententyp *ADAS* dient zur Parametrierung und Aufbereitung von Achswerten, die mit Hilfe des Software-Pakets ADAS von einer SINUMERIK-Steuerung über den PROFIBUS DP an SIMIT übertragen werden können.

Hinweis

Das Software-Paket ADAS ist nicht Bestandteil des Produktes SIMIT und muss separat erworben werden.

Zur Übertragung der Achswerte müssen Sie den PROFIBUS-Slave *ccADAS* in die Hardwarekonfiguration der SINUMERIK einfügen. Die GSD-Datei des *ccADAS* finden Sie auf der Installations-CD von SIMIT im Ordner *Tools/ADAS*.



Dieser Slave besitzt je 8 Byte Ein- und Ausgangsdaten für die Kommunikation und weitere 4 Byte pro Achswert.

Profibus

Speichern und Laden

Eingänge Filter rücksetzen

Vorgabe	Symbolname	Adresse	Datentyp	System	Slave	Steckpla	Kommentar
				1	8		
0		ED17	DWORD	1	8	1	
0		ED21	DWORD	1	8	1	

Ausgänge Filter rücksetzen

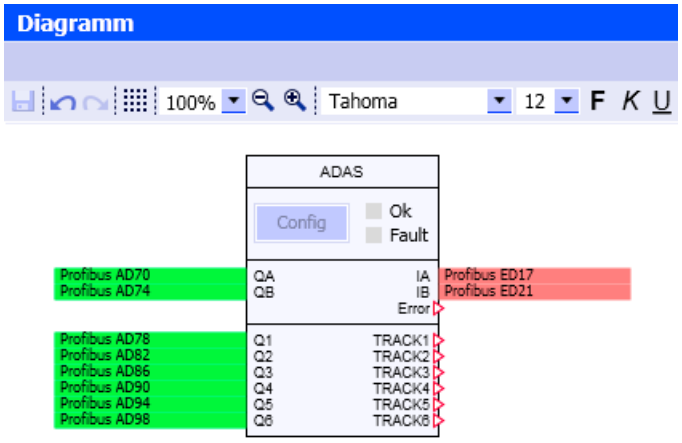
Symbolname	Adresse	Datentyp	System	Slave	Steckpla	Kommentar
			1	8		
	AD70	DWORD	1	8	1	
	AD74	DWORD	1	8	1	
	AD78	DWORD	1	8	1	
	AD82	DWORD	1	8	1	
	AD86	DWORD	1	8	1	
	AD90	DWORD	1	8	1	
	AD94	DWORD	1	8	1	
	AD98	DWORD	1	8	1	

Profibus

- Profibus
  - System 1
    - [3] L-SC IM-SC/def.DP
    - [6] SLAVE_8109
    - [8] ccADAS
      - [1] Header + 6 Tracks (4 Byte)
    - [32] ET 200S HighFeature (Cu)

Eigenschaft	Wert
Modul	Header + 6 Tracks (4 Byte)
Steckplatz	1
Adressbereich Eingänge	EB17 - EB24
Adressbereich Ausgänge	AB70 - AB101
Failsafe	Nein
Modul ziehen	

Alle E/A-Daten werden als Doppelworte in der Kopplung angelegt und müssen an der SIMIT-Komponente angeschlossen werden (*QA*, *QB*, *Q1* .. *Qn* bzw. *IA* und *IB*):



An den analogen Ausgängen *TRACKi*, *i = 2, …, N* einer Komponente vom Typ ADAS stehen die übertragenen Achswerte zur Verfügung. Die Anzahl *N* an verfügbaren Kanälen können Sie durch Aufziehen der Komponente vorgeben. Maximal können Sie 28 Kanäle einstellen. Die Ausgänge *TRACKi* können Sie dann beispielsweise mit dem 3D Viewer-Control von SIMIT verknüpfen, um die 3D-Ansicht der Maschine mit den aktuellen Achspositionen zu animieren.

Hinweis

Die Anzahl der übertragenen Achswerte wird mit der Parametrierung des Slaves *ccADAS* eingestellt. Um alle Kanäle in der Komponente ADAS nutzen zu können, sollten Sie deren Anzahl auf den gleichen Wert wie im Slave einstellen.

Der ganzzahlige Ausgang *Error* liefert Hinweise auf die Ursache im Falle einer fehlgeschlagenen Kommunikation.

Tabelle 8-39 Fehlerwerte der Komponente ADAS

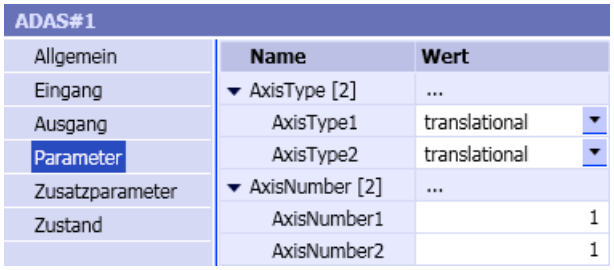
Error	Ursache
0	Kein Fehler
1	Keine Antwort auf das Konfigurieren von Track 1 erhalten
2	Keine Antwort auf das Konfigurieren von Track 2 erhalten
..	
28	Keine Antwort auf das Konfigurieren von Track 28 erhalten
100	Keine Antwort auf den Reset-Befehl erhalten
101	Keine Antwort auf den Befehl Set Communication erhalten

Parameter

Die Komponente *ADAS* besitzt die Parametervektoren *AxisType* und *AxisNumber*, deren Anzahl an Elementen der Anzahl der eingestellten Kanäle entspricht. Damit kann für jeden Kanal festgelegt werden, welche Achse der SINUMERIK-NC übertragen werden sollen und ob die entsprechende Achse linear verfährt oder eine Drehachse ist.

- *AxisType*
  Abhängig von dieser Parametrierung werden die Achswerte in mm (translational), Winkelgraden (rotatory grad) oder Radianten (rotatory rad) ausgegeben.
- *AxisNumber*
  Dieser Parameter legt fest, welche Achse auf dem entsprechenden ADAS-Kanal übertragen werden soll.

Die folgende Abbildung zeigt die Parameter mit ihren Voreinstellungen:



Hinweis

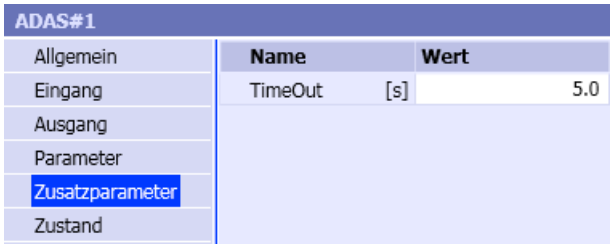
Beachten Sie, dass die Parametrierung nur wirksam wird, wenn Sie bei laufender Simulation und angeschlossener SINUMERIK-Steuerung die Schaltfläche *Config* im Bedienfester oder auf dem Grundsymbol betätigen, um diese Konfiguration an die SINUMERIK zu übertragen.

Alternativ können Sie diese Einstellungen auch in den Maschinendaten der SINUMERIK vornehmen.

Zusatzparameter

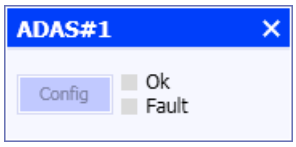
Über den Zusatzparameter *TimeOut* können Sie einstellen, nach welcher Zeit die Komponente die Kommunikation mit der SINUMERIK abbricht, falls nach dem Auftrag zur Übertragung der Konfiguration keine Antwort der SINUMERIK eintrifft.

Die folgende Abbildung zeigt den Zusatzparameter mit seinen Voreinstellungen:



Bedienfenster

Im Bedienfenster der Komponente vom Typ *ADAS* kann die Übertragung der Konfiguration an die SINUMERIK über die Schaltfläche *Config* angestoßen werden. Während der Vorgang läuft, werden die beiden Anzeigen *Ok* und *Fault* ausgeschaltet. Das erfolgreiche Übertragen der Konfiguration wird durch die grüne Anzeige *Ok* angezeigt. Falls die Übertragung scheitert, wird die rote Anzeige *Fault* aktiviert.



Die gleichen Bedien- und Anzeigeelemente befinden sich auch auf dem Grundsymbol und können dort mit gleicher Funktion benutzt werden.

8.1.7 Controls

8.1.7.1 Controls zur Anzeige von Signalwerten

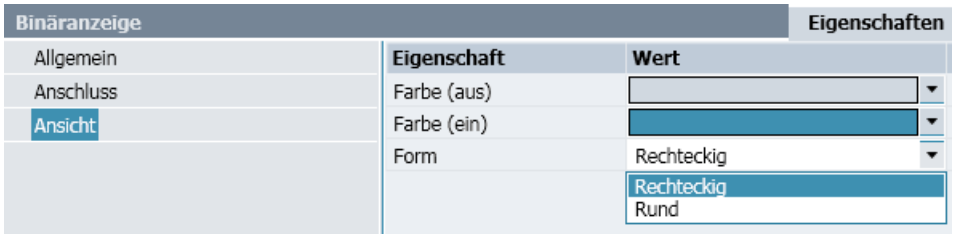
Binäranzeige

Symbol



Funktion

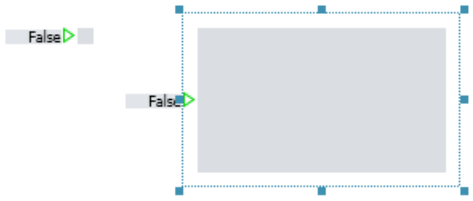
Das Control *Binäranzeige* dient zur Anzeige eines Binärwerts. In den Ansichtseigenschaften können die Farbe und Form des Controls eingestellt werden.



Die Farbe kann für den Auszustand des Signals, d. h. Signalwert null, und den Einzustand, d. h. Signalwert eins, beliebig gewählt werden. Über eine Auswahl kann die Form des Controls zwischen *Rechteckig* und *Rund* umgeschaltet werden.



Die Größe des Controls kann über Anfasser am Selektionsrahmen in Breite und Höhe beliebig verändert werden. Wenn Sie beim Verändern der Größe über die Eckanfasser <Shift> gedrückt halten, wird das Control proportional vergrößert oder verkleinert.



Analoganzeige

Symbol

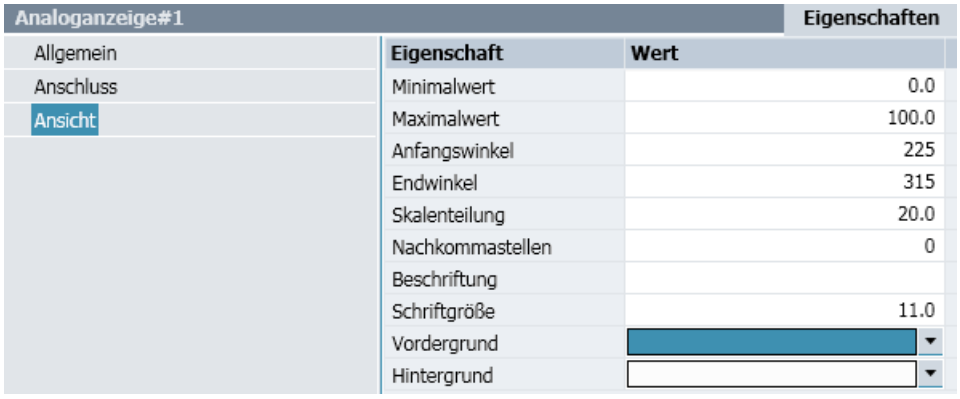


Funktion

Das Control Analoganzeige dient zur Anzeige eines analogen oder ganzzahligen Wertes in Form eines Zeigerinstruments. Der *Datentyp* des anzuzeigenden Signals kann in den allgemeinen Eigenschaften des Controls zwischen *Analog* und *Integer* umgeschaltet werden.



Die Ansicht der Analoganzeige kann durch weitere Einstellungen in den Ansichtseigenschaften individuell angepasst werden. Folgende Abbildung zeigt das Eigenschaftsfenster mit den Voreinstellungen.

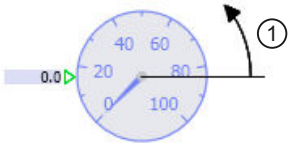


Einstellbar sind der Wertebereich und die Darstellung nach Skala, Beschriftung und die Farbe:

- **Wertebereich**
  Der Wertebereich wird durch den Minimalwert und den Maximalwert bestimmt. Liegt der anzuzeigenden Wert außerhalb des eingestellten Wertbereichs, dann wird der Zeiger der Analoganzeige in roter Farbe dargestellt.



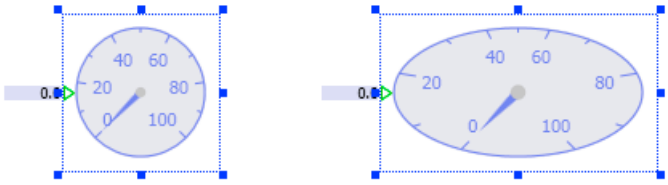
- **Skala**
  Anfangswinkel und *Endwinkel* kennzeichnen Beginn und Ende der Skala. Am Anfangswinkel wird der Minimalwert, am Endwinkel der Maximalwert angezeigt. Der Winkel wird in Grad von der waagerechten Achse entgegen dem Uhrzeigersinn angesetzt.
  Die *Skalenteilung* gibt die Unterteilung für Skalenwerte an. Skalenwerte werden mit der definierten Zahl an *Nachkommastellen* angezeigt.



① Winkel

- **Beschriftung**
  Im Eigenschaftsfeld *Beschriftung* kann ein Text eingegeben werden, der mit der einstellbaren *Schriftgröße* in der Analoganzeige eingeblendet wird.
- **Farbe**
  Berandung, Skala und Texte werden in einer einstellbaren einheitlichen Farbe für den Vordergrund dargestellt. Die Hintergrundfarbe als Füllfarbe des Controls kann ebenfalls eingestellt werden.

Die Größe und damit die runde Form des Controls kann über Anfasser am Selektionsrahmen in Breite und Höhe beliebig verändert werden. Wenn Sie beim Verändern der Größe über die Eckanfasser <Shift> gedrückt halten, wird das Control proportional vergrößert oder verkleinert.



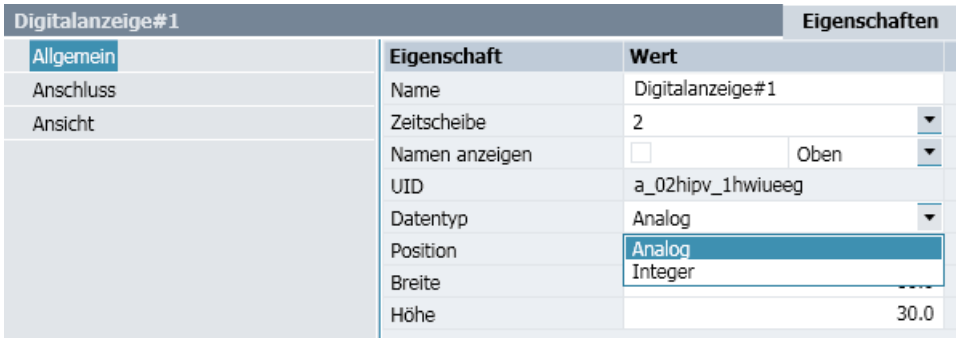
Digitalanzeige

Symbol



Funktion

Das Control *Digitalanzeige* dient zur Anzeige des Wertes eines analogen oder ganzzahligen Signals. Der *Datentyp* des anzuzeigenden Signals kann in den allgemeinen Eigenschaften des Controls zwischen *Analog* und *Integer* umgeschaltet werden.



Im Eigenschaftsfenster *Ansicht* können für analoge Signale die *Schriftgröße* und die Anzahl der anzuzeigenden *Nachkommastellen* eingestellt werden.

Für ganzzahlige Signale können im Eigenschaftsfenster die *Schriftgröße*, das *Anzeigeformat* und die *Datenbreite* eingestellt werden.



Dezimalzahlen können positiv oder negativ sein. In der hexadezimalen Darstellung werden negative Zahlen immer im Zweierkomplement dargestellt. Bei der Umrechnung in hexadezimale Zahlen muss eingestellt werden, wie viele Bytes berücksichtigt werden sollen (1, 2, 4 oder 8 Byte). Da hexadezimale Zahlen mit fester Zeichenanzahl dargestellt werden,

bestimmt diese Einstellung gleichzeitig auch die Anzahl dargestellter Hexadezimalzeichen (2, 4, 8 oder 16 Stellen).

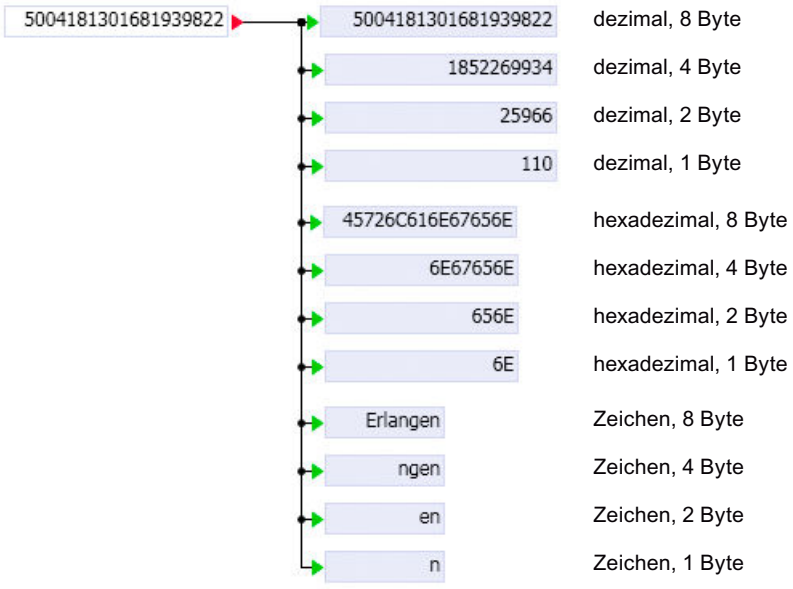
Im Anzeigeformat *Zeichen* wird nur die eingestellte Anzahl Byte berücksichtigt, beginnend mit dem niedrigstwertigen Byte, wodurch sich die Anzeige von 1, 2, 4 oder 8 Zeichen ergibt. Es wird ein Zeichen entsprechend der Codierung des (erweiterten) ASCII-Codes angezeigt, sofern es sich um ein darstellbares Zeichen handelt.

Hinweis

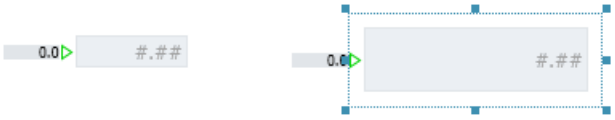
Wenn die Datenbreite auf weniger als 8 eingestellt ist, wird unter Umständen nicht der tatsächliche Wert einer ganzzahligen Eingangsgröße angezeigt.

Die Datenbreite spielt bei der Eingabe keine Rolle; eine Begrenzung des wirksamen Wertes findet nicht statt. Allerdings wird der eingegebene Wert nur entsprechend der eingestellten Datenbreite dargestellt.

Folgende Abbildung zeigt beispielhaft die Auswirkung unterschiedlicher Anzeigeformate und Datenbreiten:



Die Größe des Symbols kann über Anfasser am Selektionsrahmen in Breite und Höhe beliebig verändert und damit der eingestellten Schriftgröße angepasst werden. Wenn Sie beim Verändern der Größe über die Eckanfasser <Shift> gedrückt halten, wird das Control proportional vergrößert oder verkleinert.



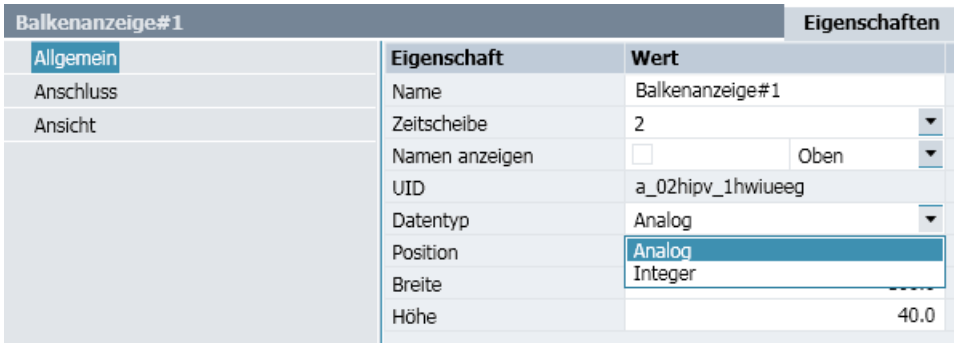
Balkenanzeige

Symbol



Funktion

Die *Balkenanzeige* dient zur Anzeige eines analogen oder ganzzahligen (Integer) Signals als Balkendiagramm. Der *Datentyp* des anzuzeigenden Signals kann in den allgemeinen Eigenschaften des Controls zwischen *Analog* und *Integer* umgeschaltet werden.

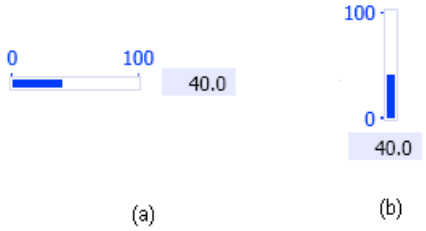


Die Ansicht der Balkenanzeige kann durch weitere Einstellungen im Eigenschaftsfenster individuell angepasst werden.



Einstellbar sind folgende Eigenschaften:

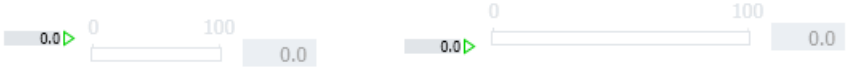
- **Skala**
  Der dargestellte Skalenbereich ist durch den *Anfangswert* und den *Endwert* gegeben. Über das Optionskästchen *Skala anzeigen* kann die Skalenanzeige zu- und abgeschaltet werden.
- **Orientierung**
  Für die Orientierung des Controls kann zwischen *Horizontal* (a) und *Vertikal* (b) gewählt werden.



(a) (b)

- **Anzeige des Balkenwerts**
  Über das Optionskästchen *Wert anzeigen* kann die zusätzliche numerische Anzeige des Balkenwerts zu- oder abgeschaltet werden.

Das Symbol kann über Anfasser am Selektionsrahmen in der Größe verändert werden. Die Balkenanzeige kann somit beispielweise bei horizontaler Ausrichtung auch breiter eingestellt werden.



8.1.7.2 Controls zur Eingabe von Signalwerten

Taster

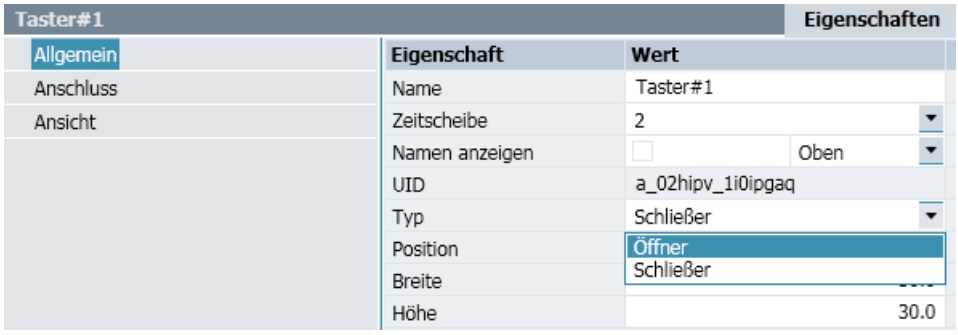
Symbol



Funktion

Der *Taster* dient zur Eingabe eines binären Signals.

In den allgemeinen Eigenschaften kann der Taster als *Schließer* oder *Öffner* eingestellt werden. Die Voreinstellung ist *Schließer*.



Der Taster wird bei laufender Simulation durch einen Mausklick auf die Schaltfläche betätigt. Die folgende Abbildung zeigt mit (a) den nicht betätigten Taster und mit (b) den betätigten Taster:



(a)



(b)

Für den *Taster* kann, wie im Eigenschaftsfenster der folgenden Abbildung zu sehen, ein *Text* vorgegeben werden, der mit der einstellbaren *Schriftgröße* auf der Schaltfläche dargestellt wird:



Über Anfasser am Selektionsrahmen der Schaltfläche kann deren Größe verändert und so beispielsweise der Größe des vorgegebenen Texts angepasst werden. Der Text wird horizontal und vertikal mittig in der Schaltfläche ausgerichtet.



Taster mit Bild

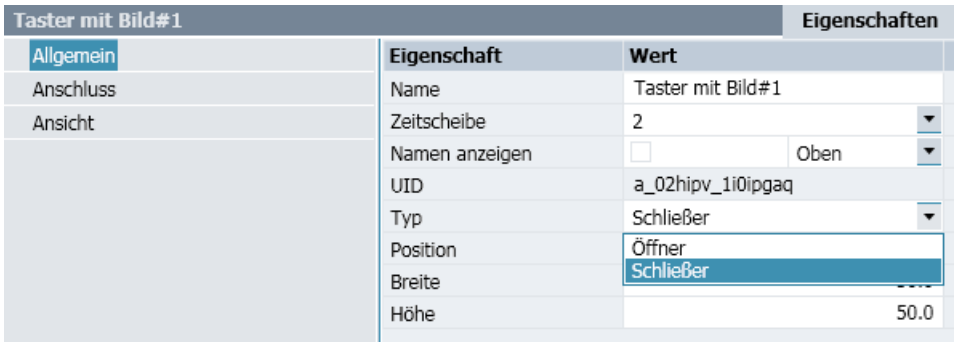
Symbol



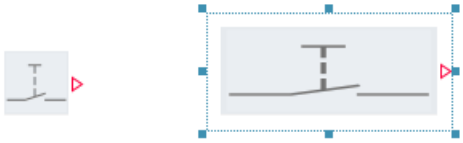
Funktion

Der *Taster mit Bild* dient zur Eingabe eines binären Signals, wobei Bilder zur Darstellung der Taststellung eingesetzt werden.

In den allgemeinen Eigenschaften kann der Taster als *Schließer* oder *Öffner* eingestellt werden. Die Voreinstellung ist *Schließer*.



Der Taster wird bei laufender Simulation durch einen Mausklick auf das Symbol betätigt. Die Größe des Symbols und damit die Größe der sensitiven Fläche kann über Anfasser am Selektionsrahmen verändert werden.



Im Eigenschaftsfenster können Bilder vorgegeben werden, die bei laufender Simulation den *nicht betätigten* und den *betätigten* Taster darstellen. Mit ... wird der Dialog zur Auswahl der entsprechenden Grafikdateien geöffnet. Die Auswahl kann mit X wieder gelöscht werden.



In der Voreinstellung ist das Optionskästchen *an Bildgröße anpassen* nicht aktiviert. Die ausgewählten Bilder werden dann in die Schaltfläche eingepasst, d. h. entsprechend in Breite und Höhe skaliert. Ist das Optionskästchen aktiviert, dann wird die Größe der Schaltfläche an die Größe der Grafik für *Bild (nicht betätigt)* angepasst.

Schalter

Symbol



Funktion

Der *Schalter* dient zum Schalten eines binären Signals.

In den allgemeinen Eigenschaften kann der Schalter als *Schließer* oder *Öffner* in Stellung *Aus* oder *Ein* eingestellt werden. Er besitzt die Voreinstellung *Schließer* in Stellung *Aus*. Die Stellungsauswahl wird als *Voreinstellung* wirksam beim Starten der Simulation.



Betätigt wird der Schalter bei laufender Simulation durch Mausklick auf die Schaltfläche. Der geschlossene Schalter wird mit einem dunkelblauen Rahmen dargestellt. Die Tabelle zeigt den Schalter als Öffner und Schließer mit den Voreinstellungen *Ein* und *Aus*.

Voreinstellung	Öffner	Schließer
Ein		
Aus		

Über Anfasser am Selektionsrahmen der Schaltfläche kann deren Größe verändert werden.



Schalter mit Bild

Symbol



Funktion

Der *Schalter mit Bild* dient zum Schalten eines binären Signals, wobei Bilder zur Darstellung der Schaltstellung eingesetzt werden.

In den allgemeinen Eigenschaften kann der Schalter als *Schließer* oder *Öffner* in Stellung *Aus* oder *Ein* eingestellt werden. Er besitzt die Voreinstellung *Schließer* in Stellung *Aus*. Die Stellungsauswahl wird als *Voreinstellung* wirksam beim Starten der Simulation.



Betätigt wird der Schalter bei laufender Simulation durch Mausklick auf das Symbol. Die Größe des Symbols und damit die Größe der sensitiven Fläche kann über Anfasser am Selektionsrahmen verändert werden.



In den Ansichtseigenschaften können Bilder vorgegeben werden, die bei laufender Simulation den Schalter in den beiden möglichen Schaltzuständen *Aus* und *Ein* darstellen. Mit ... wird der Dialog zur Auswahl der entsprechenden Grafikdateien geöffnet. Die Auswahl kann mit X wieder gelöscht werden.



In der Voreinstellung ist das Optionskästchen *an Bildgröße anpassen* nicht aktiviert. Die ausgewählten Bilder werden dann in die Schaltfläche eingepasst, d. h. entsprechend in Breite und Höhe skaliert. Ist das Optionskästchen aktiviert, dann wird die Größe der Schaltfläche an die Größe der Grafik für *Bild (Aus)* angepasst.

Stufenschalter

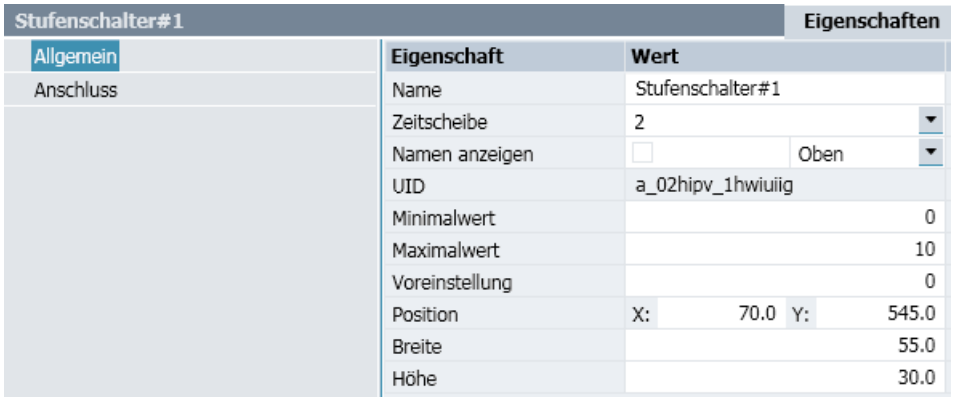
Symbol



Funktion

Der *Stufenschalter* dient zum stufenweisen Schalten eines ganzzahligen (Integer) Wertes: Der Zahlenwert wird mit jedem Schalten um eins erhöht oder erniedrigt.

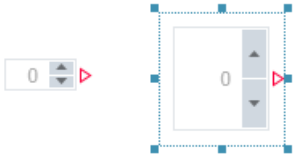
In den allgemeinen Eigenschaften können der Minimalwert und der Maximalwert für das geschaltete Signal definiert werden. Voreingestellt ist ein zehnstufiger Schalter mit einem Wertebereich von 0 bis10 beginnend beim Schaltwert 0.



Betätigt wird der Stufenschalter bei laufender Simulation durch Mausklick auf die obere ( ) oder untere ( ) Schaltfläche. Der aktuell eingestellte Schaltwert wird im Symbol angezeigt.



Die Größe des Symbols und damit auch die Größe der beiden Schaltflächen kann über Anfasser am Selektionsrahmen verändert werden.



Stufenschalter mit Bild

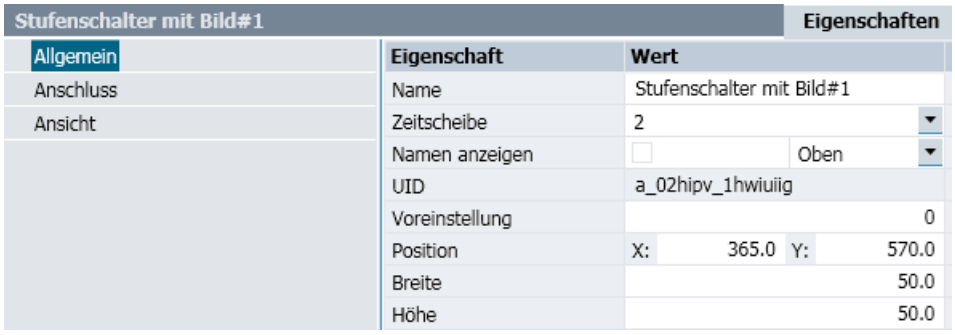
Symbol



Funktion

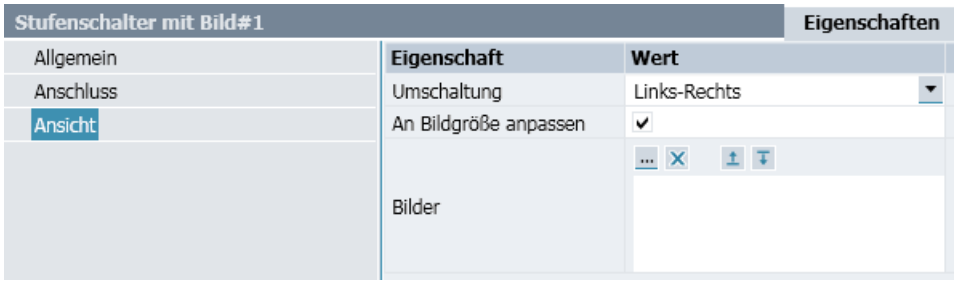
Der *Stufenschalter mit Bild* dient zum stufenweisen Schalten eines ganzzahligen (Integer) Wertes, wobei Bilder zur Darstellung der Schaltstellung eingesetzt werden.

In den allgemeinen Eigenschaften kann der Schaltwert vorgegeben werden, mit dem der Stufenschalter beginnt. Die *Voreinstellung* ist null. Der Zahlenwert des geschalteten Signals wird mit jedem Schalten um eins erhöht oder erniedrigt.



Die Anzahl der Schaltstufen ist durch die Anzahl der Bilder definiert. Die Bilder werden in den Ansichtseigenschaften vorgegeben. Sie werden bei laufender Simulation beim Weiterschalten in der durch die *Bilder*-Liste vorgegebenen Reihenfolge verwendet.

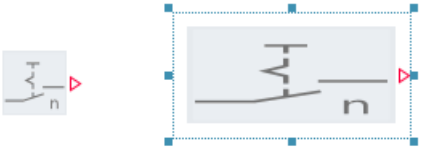
Mit ... wird der Dialog zur Auswahl der entsprechenden Grafikdateien geöffnet. Das ausgewählte Bild wird der Liste hinzugefügt. Die Reihenfolge der Bilder kann durch paarweises Tauschen eines Bildes mit seinem Vorgänger ( ) oder Nachfolger ( ) geändert werden. Mit kann ein Bild gelöscht werden.



Betätigt wird der Stufenschalter bei laufender Simulation durch Mausklick auf das Symbol. Die *Umschaltung* wird über die Klappliste ausgewählt werden.

Der Wert des Stufenschalters wird in der Parametrierung "Oben-Unten" inkrementiert, wenn man in die obere Hälfte des Controls klickt, und dekrementiert, wenn man in die untere Hälfte klickt.

Die Größe des Symbols kann über Anfasser am Selektionsrahmen verändert werden.



In der Voreinstellung ist das Optionskästchen *an Bildgröße anpassen* nicht aktiviert. Die Bilder werden dann in die Schaltfläche eingepasst, d. h. entsprechend in Breite und Höhe skaliert. Ist das Optionskästchen aktiviert, dann wird die Größe der Schaltfläche an die Größe der Grafik für das erste Bild angepasst.

Digitaleingabe

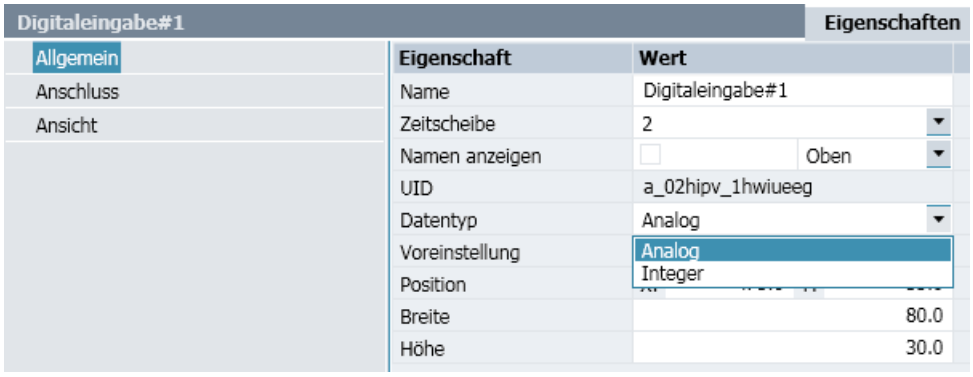
Symbol



Funktion

Die *Digitaleingabe* dient zur numerischen Eingabe eines analogen oder ganzzahligen (Integer) Signalwerts.

In den allgemeinen Eigenschaften kann als *Datentyp* des Signals *analog* oder *integer* gewählt werden. Die *Voreinstellung* des Signalwerts kann gesetzt werden. Sie ist mit null vorgegeben.



Für analoge Signale können die *Schriftgröße* und die Anzahl der angezeigten *Nachkommastellen* in den Ansichtseigenschaften gesetzt werden.



Für ganzzahlige Signale können im Eigenschaftsfenster die *Schriftgröße*, das *Anzeigeformat* und die *Datenbreite* eingestellt werden.



Die Eingaben müssen im eingestellten Anzeigeformat erfolgen. Falls die Eingabe bezogen auf das eingestellte Anzeigeformat syntaktisch falsch ist, wird die Eingabe als null interpretiert.

Dezimalzahlen können positiv oder negativ sein. In der hexadezimalen Darstellung werden negative Zahlen immer im Zweierkomplement dargestellt. Bei der Umrechnung in hexadezimale Zahlen muss eingestellt werden, wie viele Bytes berücksichtigt werden sollen (1, 2, 4 oder 8 Byte). Da hexadezimale Zahlen mit fester Anzahl Zeichen dargestellt werden, bestimmt diese Einstellung gleichzeitig auch die Anzahl dargestellter Hexadezimalzeichen (2, 4, 8 oder 16 Stellen).

Im Anzeigeformat *Zeichen* wird nur die eingestellte Anzahl Byte berücksichtigt, beginnend mit dem niedrigstwertigen Byte, wodurch sich die Anzeige von 1, 2, 4 oder 8 Zeichen ergibt. Es wird ein Zeichen entsprechend der Codierung des (erweiterten) ASCII-Codes angezeigt, sofern es sich um ein darstellbares Zeichen handelt.

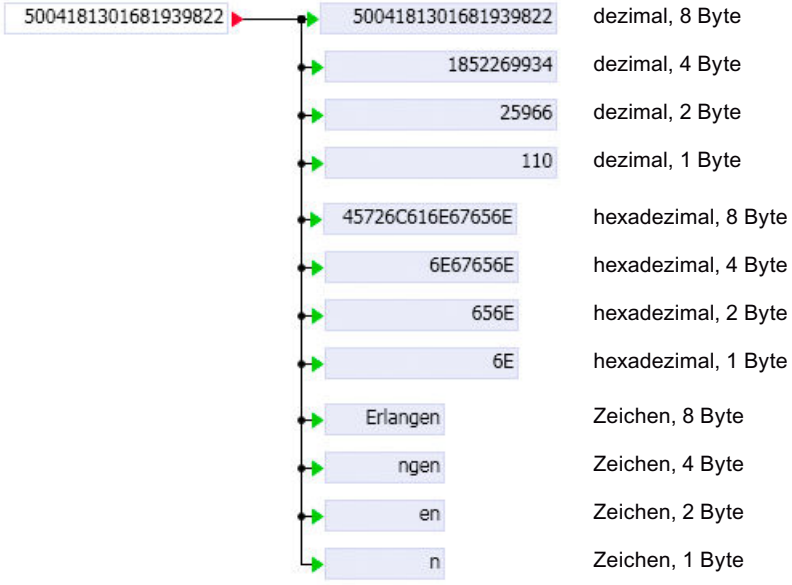
Die Datenbreite spielt bei der Eingabe keine Rolle, eine Begrenzung des wirksamen Wertes findet nicht statt. Der eingegebene Wert wird nur entsprechend der eingestellten Datenbreite dargestellt.

Hinweis

Wenn die Datenbreite auf weniger als 8 eingestellt ist, wird unter Umständen nicht der tatsächliche Wert einer ganzzahligen Eingangsgröße angezeigt.

Bei der Eingabe von Werten über das Control Digitaleingabe müssen die Eingaben im eingestellten Anzeigeformat erfolgen. Ist die Eingabe syntaktisch falsch, wird sie als "0" interpretiert.

Folgende Abbildung zeigt beispielhaft die Auswirkung unterschiedlicher Anzeigeformate und Datenbreiten:



Die Größe des Symbols kann über Anfasser am Selektionsrahmen in Breite und Höhe beliebig verändert und damit der eingestellten Schriftgröße angepasst werden. Wenn Sie beim Verändern der Größe über die Eckanfasser <Shift> gedrückt halten, wird das Control proportional vergrößert oder verkleinert.



Die Signalwerte werden bei laufender Simulation rechtsbündig im Symbol eingeblendet.

Schieber

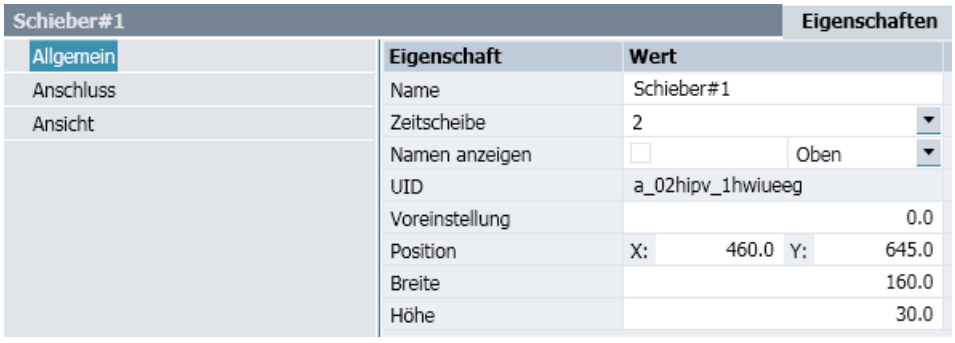
Symbol



Funktion

Der *Schieber* dient zur Vorgabe eines analogen oder ganzzahligen (Integer) Signals.

In den allgemeinen Eigenschaften kann die Voreinstellung für den Signalwert gesetzt werden. Die Vorbelegung ist null.



Bei laufender Simulation wird der Schieber mit der Maus verfahren: Stellen Sie dazu den Mauszeiger über den Knopf des Schiebers und bewegen Sie den Knopf mit gedrückter linker Maustaste auf die gewünschte Einstellposition. Durch Klick rechts oder links vom Knopf können Sie den Einstellwert jeweils um einen Schritt erhöhen oder erniedrigen.



Die Ansicht des Schiebers kann durch weitere Einstellungen im Eigenschaftsfenster individuell angepasst werden.



- **Skala**
  Die Skala ist durch den *Anfangswert*, den *Endwert* und die *Schrittweite* gegeben.
- **Orientierung**
  Für die Orientierung des Controls kann zwischen *Horizontal* und *Vertikal* gewählt werden. Einstellbar sind folgende Eigenschaften:



- **Anzeige des Balkenwerts**
  Über das Optionskästchen *Wert anzeigen* kann die zusätzliche numerische Anzeige des Signalwerts zu- oder abgeschaltet werden.

Das Symbol kann über Anfasser am Selektionsrahmen in der Größe verändert werden.



8.1.7.3 Sonstige Controls

Signaltrenner

Symbol

Funktion

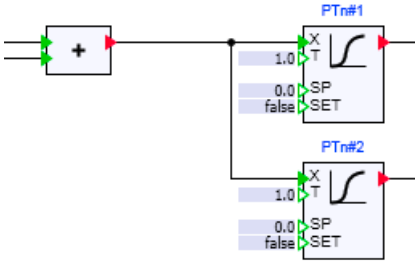
Der *Signaltrenner* ermöglicht das Forcen, d. h. das Setzen von Werten an Ein- und Ausgängen von Komponenten oder von Kopplungssignalen bei laufender Simulation mit Hilfe von beliebigen Eingabe-Controls.

Das Forcen steht bereits im Eigenschaftsfenster für alle Ein- und Ausgänge von Komponenten zur Verfügung.

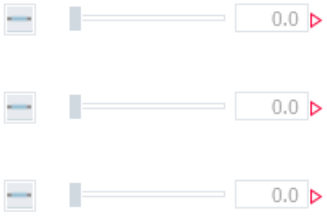
Mit dem Control *Signaltrenner* kann das Forcen von Ein- und Ausgängen darüber hinaus mit jedem Eingabe-Control realisiert werden, wie im Folgenden beispielhaft beschrieben.

Tipp: Sie können den Signaltrenner auch ohne zugeordnetes Eingabe-Control verwenden und beispielsweise mit dem zu forcenden Ein- oder Ausgang einer Komponente verbinden. Dieser Signaltrenner hat dann die gleiche Funktion wie der Signaltrenner in den Eigenschaften des Ein- bzw. Ausgangs.

In der folgenden Abbildung ist in einem Diagrammausschnitt ein Addierer zu sehen, dessen Ausgang mit den Eingängen von zwei PTn-Gliedern verbunden ist.



Sollen das Forcen des Addiererausgangs und der beiden Eingänge der PTn-Glieder über Schieber realisiert werden, dann fügen Sie auf einem Diagramm für jeden zu forcenden Ein- und Ausgang einen Schieber und einen Signaltrenner ein.

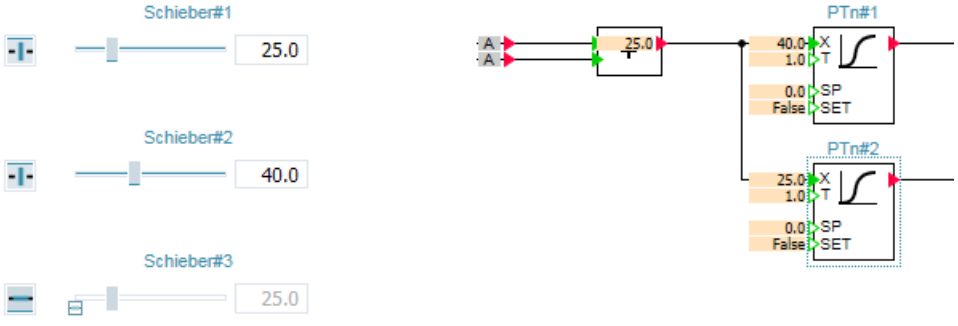


Öffnen Sie dann das Eigenschaftsfenster des ersten Schiebers, setzen Sie dessen Anschluss auf unsichtbar und tragen Sie als verbundenes Signal den Ausgang des Addierers ein. Öffnen Sie dann die Anschlusseigenschaften des zugehörigen Signaltrenners. Wie in der folgenden Abbildung zu sehen ist, ist dessen Anschluss immer unsichtbar.

Tragen Sie auch hier als Signal den Ausgang des Addierers ein.



Die beiden anderen Schieber und Signaltrenner verbinden Sie auf die gleiche Weise mit den Eingängen der beiden PTn-Glieder. Bei laufender Simulation können Sie nun den Ausgang und die Eingänge beispielsweise wie in der folgenden Abbildung dargestellt forcen.



Die Signaltrenner zu den beiden Controls *Schieber#1* und *Schieber#2* sind eingeschaltet ( ). Das Forcen des Addiererausgangs und des Eingangs von PTn#1 ist über die beiden Schieber damit aktiviert. Der Signaltrenner zu *Schieber#3* ist nicht eingeschaltet ( ), der Eingang PTn#3

kann damit nicht geforced werden. Der *Schieber#3* ist inaktiv dargestellt und zusätzlich mit dem Overlay ⊟ gekennzeichnet.

Mit den ersten beiden Schiebern können nun Werte für der Aus- und Eingang vorgegeben werden. Im Beispiel ist der Addiererausgang auf den Wert 25 und der Eingang des PTn#1 ist auf den Wert 40 gesetzt. Der Eingang von PTn#2 ist mit dem Ausgang des Addierers verbunden und nimmt damit dessen Wert (25) an. Der (inaktive) Schieber#3 zeigt den Wert des mit ihm verbundenen Eingangs.

Beim Einsatz von Signaltrennen sind somit folgende Punkte zu beachten:

- Der Signaltrenner und das zugehörige Eingabe-Control sind mit dem zu forcenden Ein- oder Ausgang zu verbinden.
- Für nicht eingeschaltete Signaltrenner ist das Forcen nicht aktiviert. Das zugehörige Eingabe-Control ist inaktiv dargestellt und durch das Overlay ⊟ gekennzeichnet. Es zeigt den Wert des verbundenen Ein- oder Ausgangs an.
- Das Einschalten des Signaltrenners aktiviert das Forcen.

Siehe auch

Eigenschaften der Eingänge (Seite 401)

Eigenschaften der Ausgänge (Seite 403)

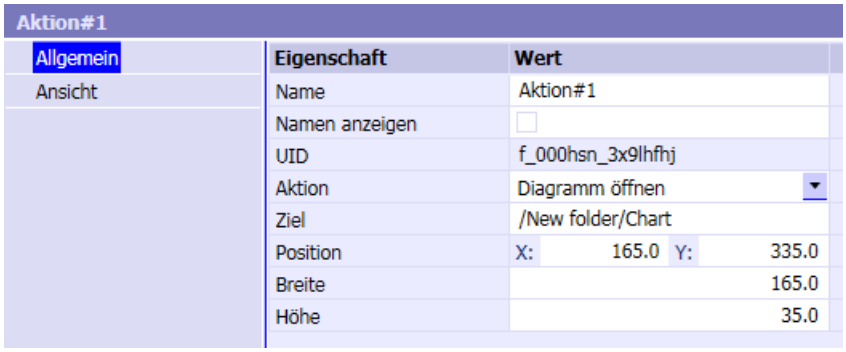
Aktion

Symbol

Funktion

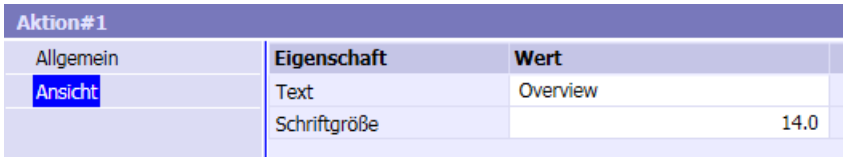
Die Aktion dient zum Öffnen eines Diagramms oder eines Kurvenbildes.

In den allgemeinen Eigenschaften kann unter "Aktion" eingestellt werden, ob ein Diagramm oder ein Kurvenbild geöffnet werden soll. Als "Ziel" tragen Sie das zu öffnende Diagramm oder Kurvenbild ein, beginnend mit einem Schrägstrich und mit Angabe der kompletten, durch Schrägstriche getrennten Ordnerhierarchie. Alternativ kann auch das entsprechende Diagramm oder Kurvenbild aus dem Projektbaum per Drag & Drop in das Eigenschaftsfeld "Ziel" gezogen werden.



Klicken Sie bei laufender Simulation auf die Schaltfläche, um die Aktion aufzurufen.

Für die Aktion kann ein Text vorgegeben werden, der mit der einstellbaren Schriftgröße auf der Schaltfläche dargestellt wird:



Die Größe der Schaltfläche kann verändert und so z. B. der Größe des vorgegebenen Texts angepasst werden. Der Text wird horizontal und vertikal mittig in der Schaltfläche ausgerichtet.

8.1.7.4 3D Viewer-Control

Der grafische Editor von SIMIT ermöglicht es Ihnen, das Verhalten einer Maschine mit einfachen grafischen Grundelementen anschaulich zu visualisieren. Mit zweidimensionaler Grafik können Sie eine Maschine skizzieren und durch Animation von entsprechenden Teilen dieser Grafik Bewegungen der Maschine veranschaulichen. Das 3D Viewer-Control bietet Ihnen darüber hinaus die Möglichkeit, dreidimensionale animierte Ansichten einer Maschine in Ihre Simulation einzubinden. Die Repräsentation der Maschine ist so anschaulicher und die Bewegungen der Maschine werden realitätsnäher visualisiert.

Voraussetzung für den Einsatz des 3D Viewer-Controls ist, dass ein dreidimensionales Geometriemodell der Maschine vorliegt und dass dieses 3D-Modell kinematisiert, d. h. so aufbereitet ist, dass es mit Signalen aus der funktionalen Simulation verknüpft werden kann und dann in der Simulation die durch diese Signale gesteuerten Animationen ausführt. Für die Kinematisierung des 3D-Modells sind Elemente definiert, die vom 3D Viewer-Control ausgewertet und in Animationen des 3D-Modells umgesetzt werden.

Das 3D Viewer-Control wird wie die anderen Controls der Basisbibliothek in ein Diagramm eingefügt und entsprechend parametriert. Es hat ebenso Anschlüsse, über die es mit Signalen des funktionalen Modells verbunden werden kann. Als Besonderheit hat das 3D Viewer-Control ein Menü, das es Ihnen ermöglicht, die Ansicht des 3D-Modells per Kommando anzupassen.

Anforderungen an das Datenformat

Voraussetzung für die Nutzung des 3D Viewer-Controls ist das Vorliegen eines dreidimensionalen Geometriemodells im Format VRML V2.0. Dieses VRML-Format können Sie aus den meisten CAD-Systemen exportieren. Gegebenenfalls müssen Sie jedoch im Anschluss an den Export dieses VRML-Modell noch umstrukturieren, um die zu animierenden Körper oder Körpergruppen als Kinematisierungspunkte identifizieren und fassen zu können. Im Hinblick auf den Umfang eines aus CAD-Systemen exportierten Modells empfiehlt sich auch eine Reduktion durch Eliminieren von für die Visualisierung unwesentlichen Details des Geometriemodells.

Für das Umstrukturieren und Kinematisieren des 3D-Modells können Sie einen geeigneten VRML-Editor einsetzen, der Ihnen die geometrische Struktur des 3D-Modells verdeutlicht und es Ihnen ermöglicht, den VRML-Code zu modifizieren. Informationen zu Editierumgebungen für VRML finden Sie beispielsweise auf den Webseiten des Web3D Consortium: www.web3d.org.

Animieren des 3D-Modells

Mit dem 3D Viewer-Control können Sie einzelne Körper oder Körpergruppen des 3D-Modells auf verschiedene Arten animieren. Sie können

- Körper oder Körpergruppen im Raum verschieben und drehen, d. h. translatorisch und rotatorisch bewegen,
- Körper oder Körpergruppen verformen, d. h. in ihrer Größe skalieren,
- Die Farbe und Transparenz von einzelnen Körpern verändern.

Im 3D-Modell müssen Sie dazu die einzelnen Körper oder Körpergruppen mit entsprechenden Elementen oder Kennungen versehen:

- Für Bewegungsanimationen werden spezifische Bewegungssensoren in das 3D-Modell eingefügt,
- Für Verformungen und Änderungen der Oberfläche eines Körpers werden der Definition des Körpers entsprechende Kennungen hinzugefügt.

Im Folgenden wird die Funktionsweise der verschiedenen Bewegungssensoren erläutert und es werden die weiteren Optionen zur Modellmodifikation beschrieben. Anhand von Beispielen wird gezeigt, wie die verschiedenen Sensoren und Kennungen in eine VRML-Datei eingefügt werden können.

Sensoren zur Animation

Mit Hilfe von Sensoren reagiert ein 3D-Modell üblicherweise auf Benutzeraktionen. Im Unterschied dazu werden Sensoren im 3D Viewer-Control auf Anschlüsse geführt. Folglich können Sie die Anschlüsse des 3D Viewer-Controls mit Signalen Ihres funktionalen Modells verbinden. Die Bewegungen werden so während der Simulation vom funktionalen Modell berechnet und vom 3D Viewer-Control visualisiert.

Vom 3D Viewer-Control werden die folgenden Sensoren zur Animation des 3D-Modells unterstützt:

- Ebenensensoren zur translatorischen Bewegung von Objekten,
- Zylindersensoren zur Rotation von Objekten um die lokalen Koordinatenachsen,
- Kugelsensoren zur Rotation von Objekten um einen Vektor.

Für jeden Bewegungssensor des 3D-Modells gilt folgendes:

- Zu einem Sensor muss mindestens ein übergeordneter Transform-Knoten existieren. Der Sensor kann an einer beliebigen Stelle im Transform-Knoten eingefügt werden.
- Die Position des Sensors allein entscheidet darüber, welche Körper bzw. Körpergruppen bewegt werden: Der dem Sensor übergeordnete Transform-Knoten und alle seine Unterknoten werden durch den Sensor bewegt.
- Routen zu Sensoren werden nicht benötigt und nicht ausgewertet.

Für jeden Sensor wird eine spezifische Anzahl von Anschlüssen in den Anschlusseigenschaften des Controls zur Verfügung gestellt. Verbinden Sie davon die Anschlüsse, die die gewünschte Bewegung im 3D-Modell ausführen mit den entsprechenden Signalen in Ihrem funktionalen Modell.

Fehlerhafte Sensoren, d. h. Sensoren, die keinem Transform-Knoten zugeordnet sind, werden als nicht vorhanden interpretiert. Folglich sind für fehlerhafte Sensoren auch keine Anschlüsse zur Animation in den Eigenschaften des 3D Viewer-Controls verfügbar.

PlaneSensor – der Ebenensensor

Ein Ebenensensor ermöglicht die Translation eines Objekts, d. h. eines Körpers oder einer Körpergruppe. Im VRML-Standard können mittels Ebenensensoren Objekte in zwei Raumrichtungen verschoben werden, in X- und Y-Richtung des lokalen Koordinatensystems. Das 3D Viewer-Control ermöglicht an jedem Ebenensensor eine Translation in alle drei Raumrichtungen. Ein Ebenensensor wird mit dem Schlüsselwort **PlaneSensor** in das VRML-Modell eingefügt.

Nach dem Einlesen der VRML-Datei in das 3D Viewer-Control sind für einen Ebenensensor drei analoge Anschlüsse in den Eigenschaften des Controls verfügbar:

Sensorname#TX	für die Translation des Objekts in X-Richtung,
Sensorname#TY	für die Translation des Objekts in Y-Richtung,
Sensorname#TZ	für die Translation des Objekts in Z-Richtung.

Ist kein *Sensorname* für einen Ebenensensor definiert, dann wird als Sensorname *TranslationN* gesetzt, wobei *N* eine fortlaufende Nummer für den Ebenensensor ist, also *N* = 1, 2, …

Im folgenden Beispiel ist ein Kegel definiert, dem ein Ebenensensor namens *ConeSensor* zugeordnet ist:

```
#VRML V2.0 utf8
DEF ConeTransform Transform
{
    children
    [
        DEF Cone Shape
        {
            appearance Appearance
            {
                material Material {}
            }
            geometry Cone {}
        }
        DEF ConeSensor PlaneSensor {}
    ]
}
```

Nach dem Einlesen dieser VRML-Datei in das 3D Viewer-Control sind die folgenden drei analogen Anschlüsse für die Translation des Kegels in den Eigenschaften des 3D Viewer-Controls verfügbar:

ConeSensor#TX	(Translation in X-Richtung),
ConeSensor#TY	(Translation in Y-Richtung),
ConeSensor#TZ	(Translation in Z-Richtung).

Sie können jetzt jeden dieser Anschlüsse mit einem analogen Signal Ihres funktionalen Modells verbinden, um die gewünschte Verschiebung des Kegels zu animieren. Für Anschlüsse, die Sie nicht mit Signalen verbunden haben, erfolgt keine Verschiebung des Kegels in die entsprechende Richtung.

CylinderSensor – der Zylindersensor

Zylindersensoren ermöglichen die Rotation eines Objekts, d. h. eines Körpers oder einer Körpergruppe um eine der drei lokalen Koordinatenachsen. Zylindersensoren sind folglich zu verwenden, wenn Objekte um die lokalen Koordinatenachsen X, Y oder Z gedreht werden sollen. Ein Zylindersensor wird mit dem Schlüsselwort **CylinderSensor** in das VRML-Modell eingefügt. Der Rotationswinkel für eine Achse wird in Grad angegeben.

Nach dem Einlesen der VRML-Datei in das 3D Viewer-Control sind für einen Zylindersensor drei analoge Anschlüsse in den Eigenschaften des Controls verfügbar:

Sensorname#RX	für die Rotation des Objekts um die lokale X-Achse,
Sensorname#RY	für die Rotation des Objekts um die lokale Y-Achse,
Sensorname#RZ	für die Rotation des Objekts um die lokale Z-Achse.

Ist kein *Sensorname* für einen Zylindersensor definiert, dann wird als Sensorname *RotationN* gesetzt, wobei *N* eine fortlaufende Nummer für den Zylindersensor ist, also *N* = 1, 2, …

Im folgenden Beispiel ist ein Kegel definiert, dem ein Zylindersensor namens *ConeSensor* zugeordnet ist:

```
#VRML V2.0 utf8
DEF ConeTransform Transform
{
    children
    [
        DEF Cone Shape
        {
            appearance Appearance
            {
                material Material {}
            }
            geometry Cone {}
        }
        DEF ConeSensor CylinderSensor {}
    ]
}
```

Nach dem Einlesen dieser VRML-Datei in das 3D Viewer-Control sind die folgenden drei analogen Anschlüsse für die Rotation des Kegels in den Eigenschaften des 3D Viewer-Controls verfügbar:

ConeSensor#RX	(Rotation um die X-Achse),
ConeSensor#RY	(Rotation um die Y-Achse),
ConeSensor#RZ	(Rotation um die Z-Achse).

Sie können jetzt jeden dieser Anschlüsse mit einem analogen Signal Ihres funktionalen Modells verbinden, um die gewünschte Rotation des Kegels zu animieren. Für Anschlüsse, die

Sie nicht mit Signalen verbunden haben, erfolgt keine Rotation des Kegels um die entsprechende Achse.

SphereSensor – der Kugelsensor

Die Einschränkung von Zylindersensoren, nur um Koordinatenachsen rotieren zu können, wird durch Kugelsensoren aufgehoben. Für Kugelsensoren wird ein Richtungsvektor angegeben, um den die Rotation erfolgen soll. Ein Kugelsensor wird mit dem Schlüsselwort **SphereSensor** in das VRML-Modell eingefügt. Der Rotationswinkel wird in Grad angegeben.

Nach dem Einlesen der VRML-Datei in das 3D Viewer-Control sind für einen Kugelsensor vier analoge Anschlüsse in den Eigenschaften des Controls verfügbar:

Sensorname#R	Winkel der Rotation,
Sensorname#X	X-Koordinate des Richtungsvektors,
Sensorname#Y	Y-Koordinate des Richtungsvektors,
Sensorname#Z	Z-Koordinate des Richtungsvektors.

Ist kein *Sensorname* für einen Kugelsensor definiert, dann wird als Sensorname *SphereSensorN* gesetzt, wobei *N* eine fortlaufende Nummer für den Kugelsensor ist, also *N* = 1, 2, …

Im folgenden Beispiel ist ein Kegel definiert, dem ein Kugelsensor namens *ConeSensor* zugeordnet ist:

```
#VRML V2.0 utf8
DEF ConeTransform Transform
{
    children
    [
        DEF Cone Shape
        {
            appearance Appearance
            {
                material Material {}
            }
            geometry Cone {}
        }
        DEF ConeSensor SphereSensor {}
    ]
}
```

Nach dem Einlesen dieser VRML-Datei in das 3D Viewer-Control sind die folgenden vier analogen Anschlüsse für die Rotation des Kegels in den Eigenschaften des 3D Viewer-Controls verfügbar:

ConeSensor#R	Winkel der Rotation,
ConeSensor#X	X-Koordinate des Richtungsvektors,
ConeSensor#Y	Y-Koordinate des Richtungsvektors,
ConeSensor#Z	Z-Koordinate des Richtungsvektors.

Sie können jetzt jeden dieser Anschlüsse mit einem analogen Signal Ihres funktionalen Modells verbinden, um den Richtungsvektor einzustellen und die gewünschte Rotation des Kegels zu animieren.

Skalieren von Objekten

Soll ein Objekt, d. h. ein Körper oder eine Körpergruppe in der Größe skaliert werden, so muss der Name des dem Objekt übergeordneten Transform-Knotens bzw. des Transform-Knotens, der die zu skalierenden Objekte enthält, modifiziert werden: Dem Namen des Transform-Knotens ist die Kennung **SCALE** voranzustellen. Für negative Skalierungswerte wird das Objekt in der Skalierungsachse gespiegelt.

Hinweis

Ein Skalierungswert null für zwei Freiheitsgrade, d. h. in zwei Richtungen lässt das Objekt in der Animation verschwinden.

Nach dem Einlesen der VRML-Datei in das 3D Viewer-Control sind für einen skalierten Transform-Knoten drei analoge Anschlüsse in den Eigenschaften des Controls verfügbar:

SCALETransformname#SX	für die Skalierung in X-Richtung,
SCALETransformname#SY	für die Skalierung in Y-Richtung,
SCALETransformname#SZ	für die Skalierung in Z-Richtung.

Dabei ist *Transformname* der Name des Transform-Knotens.

Das folgende Beispiel zeigt ein VRML-Modell mit einem Quader, für den eine Skalierung definiert ist:

```
#VRML V2.0 utf8
DEF SCALEBoxTransform Transform
{
    children
    [
        Shape
        {
            appearance Appearance
            {
                material Material {}
            }
            geometry Box {}
        }
    ]
}
```

Nach dem Einlesen dieser VRML-Datei in das 3D Viewer-Control sind die folgenden drei analogen Anschlüsse für die Skalierung des Quaders in den Eigenschaften des 3D Viewer-Controls verfügbar:

SCALEBoxTransform #SX	(Skalierung in X-Richtung),
SCALEBoxTransform #SY	(Skalierung in Y-Richtung),
SCALEBoxTransform #SZ	(Skalierung in Z-Richtung).

Sie können jetzt jeden dieser Anschlüsse mit einem analogen Signal Ihres funktionalen Modells verbinden, um die gewünschte Verformung des Quaders zu animieren. Für Anschlüsse, die Sie nicht mit Signalen verbunden haben, erfolgt keine Skalierung des Quaders in der entsprechenden Achse.

Ändern von Farbe und Transparenz eines Körpers

In VRML wird ein Körper durch einen Shape-Knoten definiert. Um die Farbe oder die Transparenzeigenschaften eines Körpers zu ändern, muss der Name des Körpers modifiziert werden: Dem Namen des Shape-Knotens wird die Kennung **RGBT** vorangestellt.

Nach dem Einlesen der VRML-Datei in das 3D Viewer-Control sind für einen mit der Kennung RGBT versehenen Shape-Knoten vier analoge Anschlüsse in den Eigenschaften des Controls verfügbar:

RGBTShapename#CT	für den Transparenzwert des Körpers,
RGBTShapename#CR	für den Rot-Anteil der Farbe des Körpers,
RGBTShapename#CG	für den Grün-Anteil der Farbe des Körpers,
RGBTShapename#CB	für den Blau-Anteil der Farbe des Körpers.

Dabei ist *Shapename* der Name des Shape-Knotens.

Die Farbe des Körpers wird durch entsprechende Werte für den Rot-, Grün- und Blau-Anteil bestimmt. Gültige Werte für einen Farbanteil liegen im Bereich 0, … ,1. Auch die Werte für die Transparenz des Körpers liegen im Bereich 0, … ,1. Der Transparenzwert 1 bedeutet hierbei, dass der Körper transparent und damit unsichtbar ist, der Transparenzwert 0 bedeutet, dass der Körper nicht transparent ist.

Im folgenden Beispiel ist ein Zylinder konstruiert, für den eine Animation der Farbe und Transparenz definiert ist:

```
#VRML V2.0 utf8
Transform
{
    children
    [
        DEF RGBTCylinder Shape
        {
            appearance Appearance
            {
                material Material {}
            }
            geometry Cylinder {}
        }
    ]
}
```

Nach dem Einlesen dieser VRML-Datei in das 3D Viewer-Control sind die folgenden vier analogen Anschlüsse für die Transparenz und Farbe des Zylinders in den Eigenschaften des 3D Viewer-Controls verfügbar:

RGBTCylinder #CT	(Transparenz des Körpers),
RGBTCylinder #CR	(Rot-Anteil des Körpers),

RGBTCylinder #CG (Grün-Anteil des Körpers),
RGBTCylinder #CB (Blau-Anteil des Körpers).

Sie können jetzt jeden dieser Anschlüsse mit einem analogen Signal Ihres funktionalen Modells verbinden, um die gewünschte Farbe und Transparenz des Zylinders zu animieren.

Wollen Sie nur die Sichtbarkeit eines Körpers animieren, dann müssen Sie lediglich den Anschluss *#CT* mit einem Signal verbinden. Der Körper wird dann in seiner Ursprungsfarbe (wie in der VRML-Datei definiert) dargestellt und Sie können ihn über die Signalwerte null und eins sichtbar und unsichtbar schalten.

Hinweis

Auch bei Änderungen der Farbwerte eines unsichtbaren Körpers bleibt der Körper weiter unsichtbar. Erst die Änderung des Transparenzwertes macht ihn wieder sichtbar.

Wenn Sie eine ganze Gruppe von Körpern unsichtbar schalten wollen, dann verschieben Sie die Gruppe in einen neuen Transformationsknoten und skalieren Sie den Knoten in zwei Achsrichtungen auf null bzw. eins. Weitere Informationen hierzu finden Sie im Abschnitt: Skalieren von Objekten (Seite 601).

Wollen Sie mehreren Körpern die gleiche Farbe zuordnen und diese animieren, dann können Sie die Vererbung von Eigenschaften nutzen: Die Materialeigenschaft von einem primären Körper kann auf beliebig viele andere Körper vererbt werden. Der primäre Körper erhält in seinem Namen die Kennung *RGBT*. Mit den entstehenden vier Anschlüssen des primären Körpers kann die Farbe (und auch die Sichtbarkeit/Transparenz) aller anderen Körper umgeschaltet werden, die diese Materialeigenschaft erben.

Das folgende Beispiel verdeutlicht dies: Zwei Zylinder werden gleichzeitig in der Farbe geändert, jedoch wird nur einer mit RGBT als Kennung gekennzeichnet.

```
#VRML V2.0 utf8
Transform
{
    children
    [
        DEF RGBTCylinder Shape
        {
            appearance Appearance
            {
                # Definition der primären Materialeigenschaft
                material DEF CylinderFarbe Material
                {
                    diffuseColor 0.2 0 0.8
                }
            }
            geometry Cylinder {}
        }
    ]
    Translation 2 0 0
}
Transform
{
    children
    [
        Shape
        {
            appearance Appearance
            {
                # Vererbung der Materialeigenschaft
                material USE CylinderFarbe
            }
            geometry Cylinder {}
        }
    ]
    Translation -2 0 0
}
```

Umschalten von Beobachtungspunkten

Sind Beobachtungspunkte (Viewpoints) in einer VRML-Datei enthalten, so können Sie diese sowohl dynamisch in der Simulation, als auch durch manuelles Bedienen des 3D Viewer-Controls umschalten. Wird eine VRML-Datei in das 3D Viewer-Control eingelesen, dann erscheint der ganzzahlige (Integer) Anschluss *VIEWPOINT* in den Anschlusseigenschaften des 3D Viewer-Controls, falls Beobachtungspunkte in der Datei definiert wurden. Der Anschluss *VIEWPOINT* kann im Wertebereich von 1, …, N angesteuert werden, wobei *N* die Anzahl der definierten Beobachtungspunkte ist.

Das folgende Beispiel zeigt die Syntax zweier Beobachtungspunkte mit den Namen Eingangsansicht und Frontansicht:

```
DEF MainView Viewpoint
{
    position    -2.076697e3 574.432739 4.039152e3
    orientation    0.192129 -0.980301 4.578408e-2 0.482376
    description    "Eingangsansicht"
}
DEF FrontView Viewpoint
{
    Position    -78.220909 1.075813e3 4.449046e3
    orientation    -0.631093 -0.427644 0.647179 2.717778e-2
    description    "Frontansicht"
}
```

Zur Umschaltung der Beobachtungspunkte verbinden Sie den Anschluss *VIEWPOINT* mit einem ganzzahligen (Integer) Signal, über dessen Werte Sie auf den entsprechenden Beobachtungspunkt umschalten können.

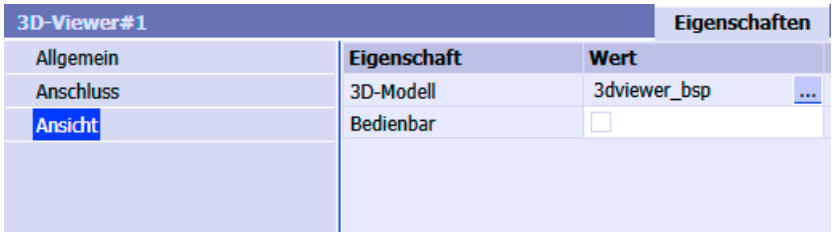
Konfigurieren des 3D Viewer-Controls

Um ein 3D-Modell im 3D Viewer-Control darzustellen und dynamisch zu modifizieren, müssen Sie ein 3D Viewer-Control in das Simulationsprojekt einfügen. In den Ansichtseigenschaften des Controls wählen Sie die VRML-Datei aus, die das 3D-Modell beschreibt. Nach der Auswahl der VRML-Datei stehen Ihnen in den Eigenschaften des Controls alle Freiheitsgrade des 3D-Modells als Anschlüsse zur Animation des Modells zur Verfügung. Die Anschlüsse können Sie dann beispielsweise per Drag & Drop mit Simulationssignalen aus der Task-Card "Signale" verknüpfen.

Importieren des 3D-Modells

Das 3D Viewer-Control wird in der Task-Card *Controls* in der Palette *Sonstige* zur Verfügung gestellt. Um ein 3D-Modell in die Simulation einzubinden, müssen Sie eine Instanz des 3D Viewer-Controls auf einem Diagramm platzieren. Dies geschieht wie auch bei allen anderen Controls per Drag & Drop aus der Task-Card "Controls". Die Größe des Controls auf dem Diagramm können Sie über Anfasser am Selektionsrahmen in der Breite und Höhe beliebig verändern und so der Größe des 3D-Modells anpassen.

In den Ansichtseigenschaften des Controls können Sie dann die VRML-Datei, die die Beschreibung des 3D-Modells enthält, einlesen.

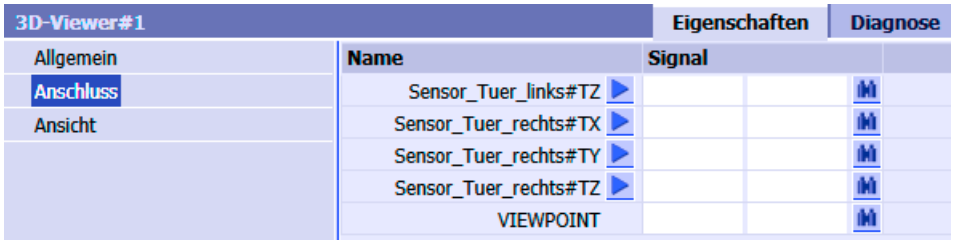


Über die Eigenschaft *Bedienbar* können Sie das 3D Viewer-Control auch bei gestoppter Simulation bedienbar schalten und die Ansicht des 3D-Modells einstellen. Weitere Informationen hierzu finden Sie im Kapitel: Simulieren mit dem 3D Viewer-Control (Seite 606)

Die eingestellte Szene (Kameraeinstellungen) wird mit dem Diagramm gespeichert und beim Öffnen des Diagramms entsprechend wiederhergestellt.

Verknüpfen der Anschlüsse mit Signalen

Mit dem Einlesen der VRML-Datei werden alle in ihr definierten Sensoren und Modifikationen erfasst und als Anschlüsse in den Anschlusseigenschaften zur Verfügung gestellt.



Die Anschlüsse, die eine Translation, Rotation, Skalierung, ein Ausblenden oder eine Farbänderung von Körpern ermöglichen, sind mit der Schaltfläche ▶ versehen. Durch Betätigung dieser Animationsschaltfläche wird eine Hilfsanimation für diesen Anschluss gestartet, die eine leichte Identifizierung des animierten Objekts bzw. seiner Bewegungsachse in der 3D-Szene ermöglicht. Verknüpfen Sie dann die Anschlüsse, die die gewünschten Animationen ausführen, per Drag & Drop mit den entsprechenden Signalen der Task-Card "Signale".

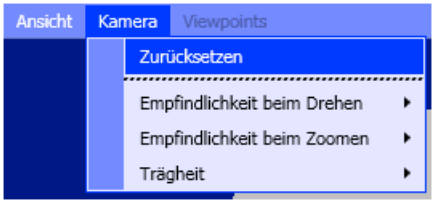
Hinweis

Von den Anschlüssen eines Kugelsensors ist nur der Anschluss #R, über den der Winkel der Rotation animiert wird, mit einer Animationsschaltfläche ▶ versehen. Beim Aktivieren dieser Animationsschaltfläche wird eine Rotation des Objekts um die X-Achse animiert.

Simulieren mit dem 3D Viewer-Control

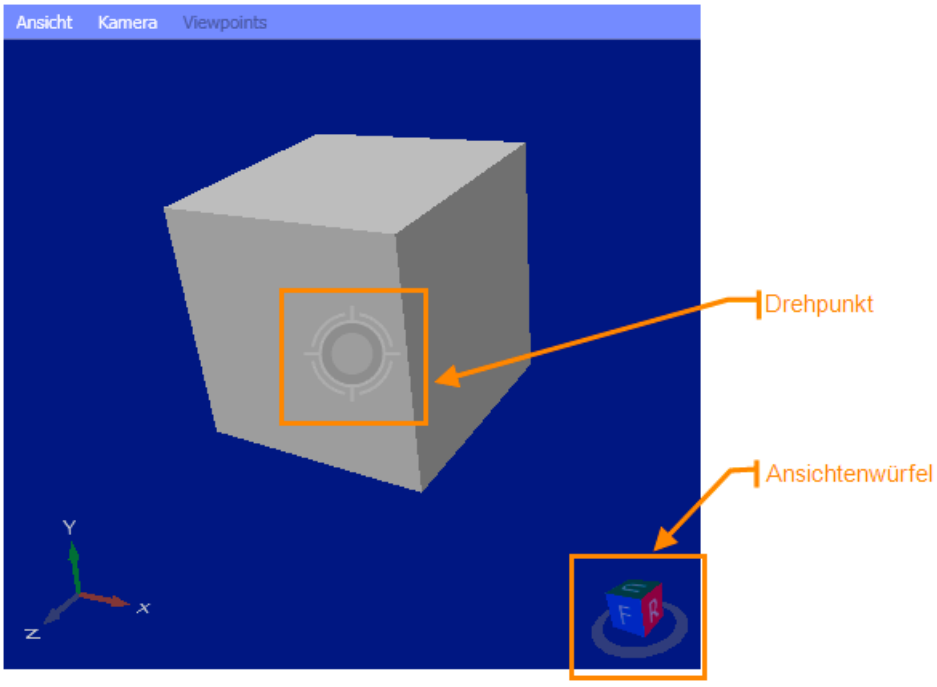
Nach dem Start der Simulation können Sie die Ansicht des 3D-Modells im 3D Viewer-Control mit Hilfe von Befehlen anpassen. Sie können die Szene (Kameraeinstellungen) drehen, verschieben oder zoomen, sowie auf definierte Beobachtungspunkte umschalten. Die gleichen Anpassungen der Szene können Sie auch vor dem Start der Simulation vornehmen, indem Sie für das 3D Viewer-Control die Ansichtseigenschaft "Bedienbar" setzen.

Sie können jederzeit die Standardansicht wieder herstellen, indem Sie den Menübefehl "Kamera > Zurücksetzen" ausführen oder die Pos1-Taste betätigen.



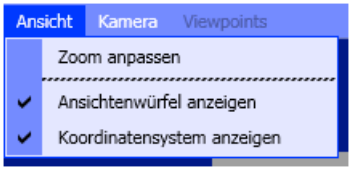
Rotieren der Szene

Die Szene können Sie mit Hilfe der Maus oder Tastatur manuell drehen. Bewegen Sie dazu den Mauszeiger auf einen Punkt der Szene und drücken Sie die linke Maustaste. Im Zentrum der Szene wird jetzt der Drehpunkt angezeigt, der gleichzeitig auch der Blickpunkt aus Sicht des Betrachters (Kamera) ist. Halten Sie dann die Maustaste gedrückt und drehen Sie die Szene in die gewünschte Richtung. Alternativ können Sie die Szene auch über die Pfeil-Tasten vertikal beziehungsweise horizontal drehen.



Den Drehpunkt/Blickpunkt können Sie durch einen Doppelklick auf ein Element der Szene neu setzen.

Der Ansichtenwürfel im rechten unteren Eck des 3D Viewer-Controls zeigt Ihnen die aktuelle Blickrichtung auf die 3D-Szene an. Durch Klicken auf eine Würfelseite können Sie die Szene auf die entsprechende Betrachtungsebene zurücksetzen. Mit einem Doppelklick stellen Sie die jeweils gegenüberliegende Betrachtungsebene ein. Den Ansichtenwürfel können Sie über den Menüpunkt "Ansicht > Ansichtenwürfel anzeigen" zu- bzw. abschalten.



Darüber hinaus können Sie die in der folgenden Tabelle gelisteten Tastenkürzel zum Umschalten der Betrachtungsebene verwenden.

Tabelle 8-40 Tastenkürzel für das Umschalten der Betrachtungsebene

Betrachtungsebene	Abkürzung	Tastenkürzel
Vorderansicht	F (Front)	Strg+F
Rückansicht	B (Back)	Strg+B
Linke Ansicht	L (Left)	Strg+L
Rechte Ansicht	R (Right)	Strg+R
Obere Ansicht	T (Top)	Strg+T
Untere Ansicht	B (Bottom)	Strg+B

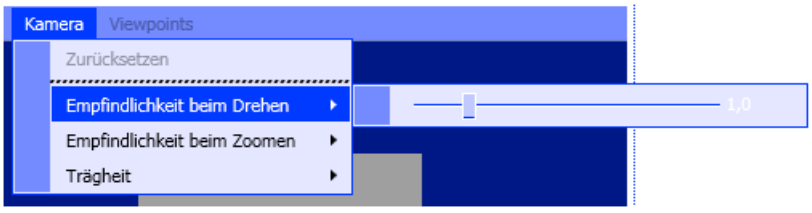
Im linken unteren Eck des 3D Viewer-Controls wird Ihnen auch das Koordinatensystem mit den drei Achsen *X*, *Y* und *Z* angezeigt. Dieses Koordinatensystem können Sie über den Menüpunkt "Ansicht > Koordinatensystem anzeigen" (s. Abbildung oben) zu- bzw. abschalten.

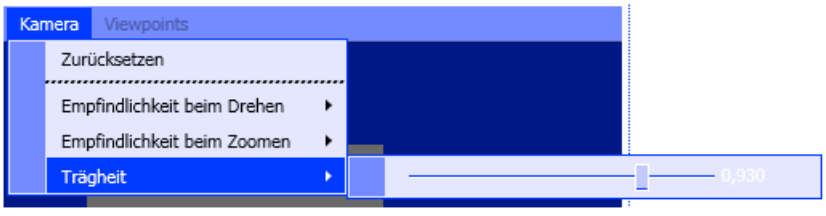
Hinweis

Die Betrachtungsebenen ergeben sich aus der VRML-Spezifikation: Verwendet wird ein kartesisches, rechtsdrehendes, dreidimensionales Koordinatensystem, bei dem die positive Y-Achse nach oben zeigt und der Betrachter von der positiven Z-Achse aus in Richtung der negativen Z-Achse blickt.

Achten Sie beim Erstellen des 3D-Modells bzw. beim Exportieren des 3D-Modells in eine VRML-Datei darauf, dass die Y-Achse entsprechend der VRML-Spezifikation nach oben zeigt.

Die Reaktion auf die Drehung mit der Maus oder mit den Pfeil-Tasten können Sie über die Menüpunkte "Kamera > Empfindlichkeit beim Drehen" und "Kamera > Trägheit" beeinflussen. Über einen Schieber können Sie die Empfindlichkeit wie auch die Trägheit verringern oder vergrößern. Voreingestellt sind beide Schieber auf den Wert eins.





Schwenken der Kamera

Durch Schwenken der Kamera können Sie die Szene in der Darstellungsfläche des 3D Viewer-Controls verschieben, d. h. die Szene aus dem oder in den Bildmittelpunkt schieben. Im Gegensatz zur Rotation bleibt beim Schwenken aber die Position des Betrachters (Kamera) konstant, es ändert sich der Blickpunkt (Betrachtungsziel).

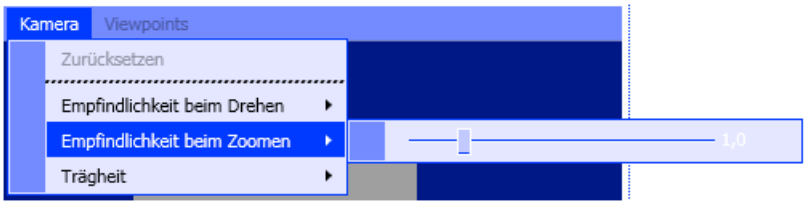
Die Bedienhandlungen für das Schwenken sind die gleichen wie für das Rotieren der Szene, nur dass Sie zusätzlich <Shift> gedrückt halten müssen. Informationen zum Rotieren der Szene finden Sie im Abschnitt: Rotieren der Szene (Seite 607).

Zoomen der Szene

Um Teilaspekte einer Szene genauer zu betrachten, können Sie diese beliebig heran- oder wegzoomen indem Sie die Bild-Auf- bzw. Bild-Ab-Taste betätigen. Das Gleiche erreichen Sie, wenn Sie das Mausrad drehen oder bei gedrückter Strg-Taste mit der linke Maustaste ziehen. Wenn Sie <Strg> und <Shift> gleichzeitig betätigen, können Sie mit gedrückter linker Maustaste einen Bereich aufziehen, der dann herangezoomt wird.

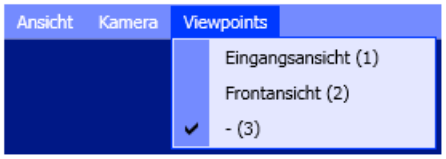
Über den Menüpunkt "Ansicht > Zoom anpassen" können Sie den Zoom jederzeit so einstellen, dass die komplette Szene flächenfüllend im 3D Viewer-Control dargestellt wird. Alle anderen Kameraeinstellungen bleiben dabei unverändert.

Die Reaktion der Szene auf das Zoomen können Sie über die Menüpunkte "Kamera > Empfindlichkeit beim Zoomen" und "Kamera > Trägheit" beeinflussen. Über einen Schieber können Sie die Empfindlichkeit wie auch die Trägheit verringern oder vergrößern. Voreingestellt sind beide Schieber auf den Wert eins.



Umschalten des Beobachtungspunkts

Ein Beobachtungspunkt (Viewpoint) beschreibt eine bestimmte Betrachterposition. Alle Beobachtungspunkte, die in der VRML-Datei definiert sind, werden im 3D Viewer-Control im Menü *Viewpoints* mit ihrem Namen und einer fortlaufenden Nummer aufgelistet. Besitzt ein Beobachtungspunkt keinen Namen, so wird er mit der Kennung "–" aufgelistet. Sie können in diesem Menü den jeweiligen Beobachtungspunkt selektieren, auf den Sie umschalten wollen.



8.2 Bibliotheken CHEM-BASIC und FLOWNET

8.2.1 Einleitung

Die Bibliotheken FLOWNET und CHEM-BASIC sind Erweiterungen von SIMIT, die Komponententypen zur Erstellung von Simulationen von Rohrleitungsnetzen zur Verfügung stellen. Mit den Komponententypen von CHEM-BASIC erstellen Sie insbesondere Simulationen in der chemischen Industrie und der Pharmaindustrie.

Durch Verschalten von Komponenten dieser Bibliotheken wird in SIMIT ein Modell eines Rohrleitungsnetzes, ein Flussnetz, erzeugt, mit dem die thermodynamischen Vorgänge in Rohrleitungsnetzen simuliert werden können. Dazu wird mit den Bibliotheken FLOWNET und CHEM-BASIC ein spezielles Lösungsverfahren in SIMIT nutzbar, das die Durchflüsse, Drücke und spezifischen Enthalpien in der Simulation von Rohrleitungsnetzen berechnet.

Mit CHEM-BASIC werden dazu Stoffgemische als Pseudo-Einstoffe mit den entsprechenden Gemischeigenschaften (insbesondere Wärmekapazität und Dichte) modelliert. Die einzelnen Komponenten der Flussnetze können zusätzlich an Modelle der Aktor-Sensor-Ebene (Basis-Bibliothek) angeschlossen werden.

Obwohl den Flussnetzen in SIMIT ein auf den physikalischen Bilanzgleichungen basierender Modellansatz zugrunde liegt, ist es nicht das Ziel, damit dynamische Prozess-Simulationen zur Auslegung von Anlagenkomponenten oder Anlagen zu ermöglichen, sondern eine physikalisch plausible Simulation der thermodynamischen Größen in Rohrleitungsnetzen für die virtuelle Inbetriebnahme zur Verfügung zu stellen. Diese Simulation soll sich einfach aus Komponenten auf einer grafischen Oberfläche erstellen lassen und auch in Grenzsituationen stabil verhalten. Bei der Implementierung der Komponententypen der Bibliotheken FLOWNET und CHEM-BASIC stand somit nicht eine detaillierte Nachbildung der Physik im Vordergrund, sondern eine einfache Parametrierung der Komponenten und ein stabiles Verhalten im Flussnetz.

Mit den Komponententypen der FLOWNET- und CHEM-BASIC-Bibliotheken können Flussnetze für unterschiedliche Medien realisiert werden:

- Wasser/Dampf,
- Flüssigkeiten oder
- ideale Gase.

Mit dem Modul Component Type Editor (CTE) von SIMIT können Sie eigene Flussnetzkomponenten erstellen und so Ihre Flussnetzbibliothek spezifisch erweitern. Das Flussnetzlösungsverfahren kann dabei über FLOWNET-spezifische Verbindungstypen von den Komponenten genutzt werden.

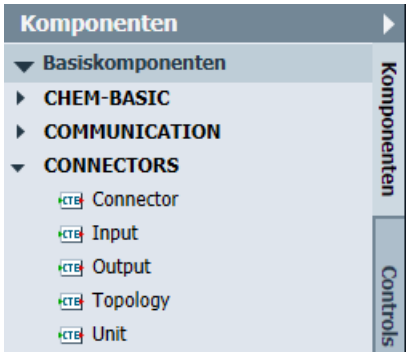
Vorhandene Komponententypen der Bibliothek CHEM-BASIC können jedoch nicht vom Anwender modifiziert werden.

Hinweis

Ob Ihre SIMIT-Installation eine Lizenz der FLOWNET- oder CHEM-BASIC-Bibliothek enthält, wird beim Starten der Simulation geprüft. Wenn in Ihrem Simulationsprojekt Flussnetzkomponenten vorhanden sind, also Komponenten, die das Lösungsverfahren für Flussnetze nutzen, dann kann die Simulation nur gestartet werden, wenn Sie die Lizenz für FLOWNET oder CHEM-BASIC besitzen.

8.2.2 Topologischer Konnektor

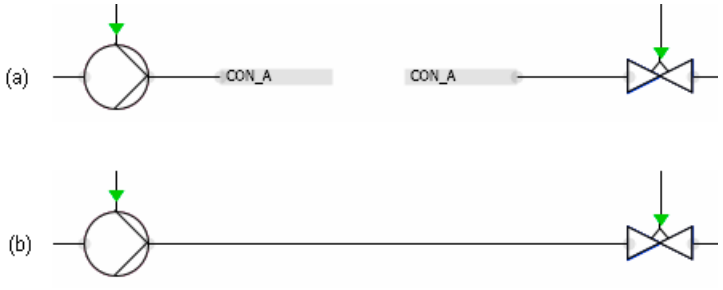
Im Verzeichnis *CONNECTORS* der Basisbibliothek von SIMIT steht der topologische Konnektor *Topology* zur Verfügung, mit dem topologische Verbindungen von Flussnetzkomponenten über Diagrammgrenzen hinweg realisiert werden können.



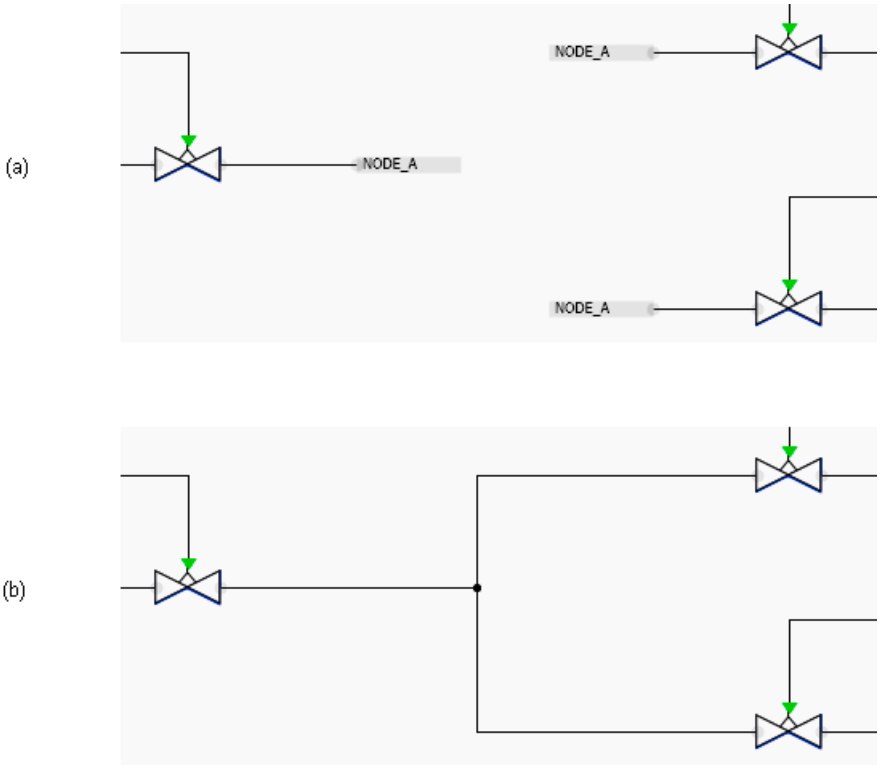
Das Symbol von *Topology* ist in der folgenden Abbildung dargestellt:



Mit dem Konnektor *Topology* kann eine topologische Flussnetzverbindung zwischen zwei oder mehreren Zweigkomponenten hergestellt werden. In der folgenden Abbildung sind unter (a) zwei Komponenten über den Konnektor *CON_A* verbunden. Diese Verbindung ist funktional identisch zur direkten Verbindung der beiden Komponenten über eine Verbindungslinie wie in der folgenden Abbildung unter (b) dargestellt:



In der folgenden Abbildung sind unter (a) drei Komponenten über den Konnektor *NODE_A* verbunden. Diese Konfiguration ist funktional identisch zu der in der folgenden Abbildung unter (b) dargestellten Verbindung mit Hilfe eines Verbindungsknotens.



8.2.3 Geodätische Höhe

Funktionsprinzip

Einige Komponenten der CHEM-BASIC Bibliothek besitzen einen (Zusatz-)Parameter für die geodätische Höhe, *geoHeight*. Die geodätische Höhe gibt dabei die Höhenlage *z* des Apparats an. Negative Werte symbolisieren dabei tiefer liegende Komponenten. Die Druckänderung durch den geodätischen Druck wird mit folgender Formel berechnet:

$\Delta P_{geo} = p \cdot g \cdot geoHeight$

Bei dem Einfluss der geodätischen Höhe auf die Simulation wird unterschieden zwischen:

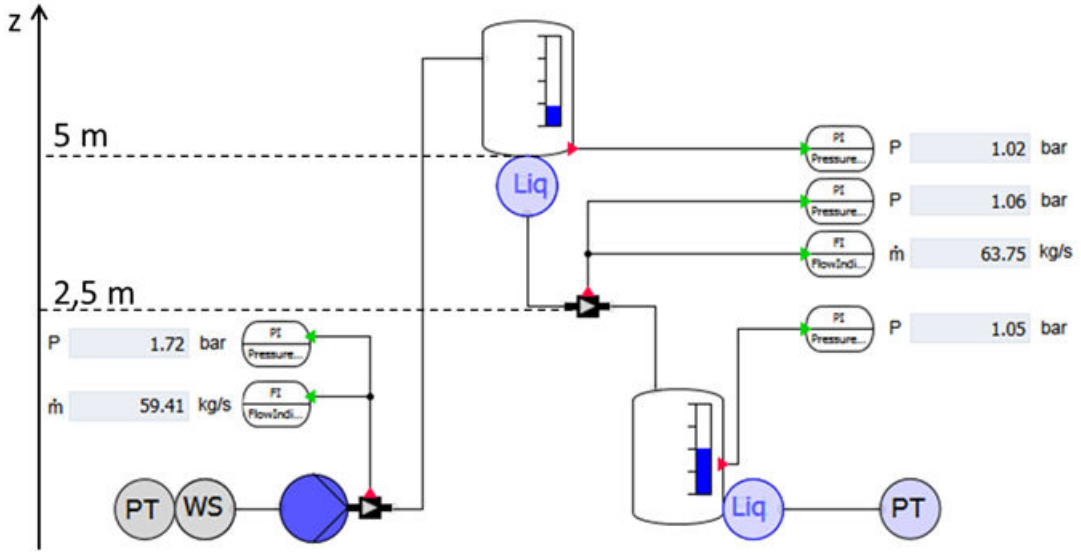
- Komponenten, die nur topologisch verknüpfte Ströme miteinander verbinden.
  Beispiel: *HeatExchangerGeneral*. Anschluss von zwei topologisch getrennten Flussnetzen. Diese tauschen Wärmemengen aus, Drücke und Massenströme beider Flussnetze sind aber getrennt voneinander.
- Komponenten, die Ströme aus topologisch nicht verknüpften Flussnetzen zusammenführen.
  Beispiel: *StorageTankLiquid*. Ströme aus verschiedenen Flussnetzen werden gemischt. Der Innendruck des Tanks wird auf die verschiedenen Flussnetze abgebildet und beeinflusst somit die Massenströme.)

Komponenten, die nur topologisch verknüpfte Ströme miteinander verbinden arbeiten nur mit Differenzdrücken. Hierfür spielt die geodätische Höhe keine Rolle. Bei diesen Komponenten ist die geodätische Höhe relevant für Stoffeigenschaften und interne Funktionen, z. B. *HeatExchangerGeneral*: Einfluss auf Siedetemperatur von Wasser.

Komponenten, die Ströme aus topologisch getrennten Flussnetzen zusammenführen, arbeiten mit Absolutdrücken. Daher hat die geodätische Höhe sowohl auf eingehende als auch austretende Ströme Einfluss. Der Druck ist in diesen Knoten um den geodätischen Druck erhöht.

Beispiel

Eine Pumpe muss eine zusätzliche Druckdifferenz aufbauen um eine Flüssigkeit in einen 5 m höher liegenden Tank zu befördern. Von dem höher liegenden Tank fließt die Flüssigkeit dann in einen niedriger liegenden Tank. Auf eine Höhe von 2,5 m ist eine Messstelle eingebaut. Alle Messstellen geben den Druck wieder, der auch real an der jeweiligen Stelle gemessen würde.



Siehe auch

Centrifuge – Zentrifuge (Seite 672)

Compressor – Verdichter (Seite 667)

Condenser – Kondensator (Seite 639)

DrumWS – Speichertank für Wasser/Dampf (Seite 710)

ElectricHeatExchanger – Elektrisch beheizter Wärmeübertrager (Seite 644)

HeatExchangerGeneral – Allgemeiner Wärmetauscher (Seite 646)

HeatExchangerShellTube – Rohrbündelwärmeübertrager (Seite 648)

Mixer – Mischer (Seite 664)

PipeMeasure – Rohrmessstelle (Seite 659)

Pump – Pumpe (Seite 670)

PressureStrainer – Drucknutsche (Seite 691)

ScreeningDevice – Siebeinrichtung (Seite 699)

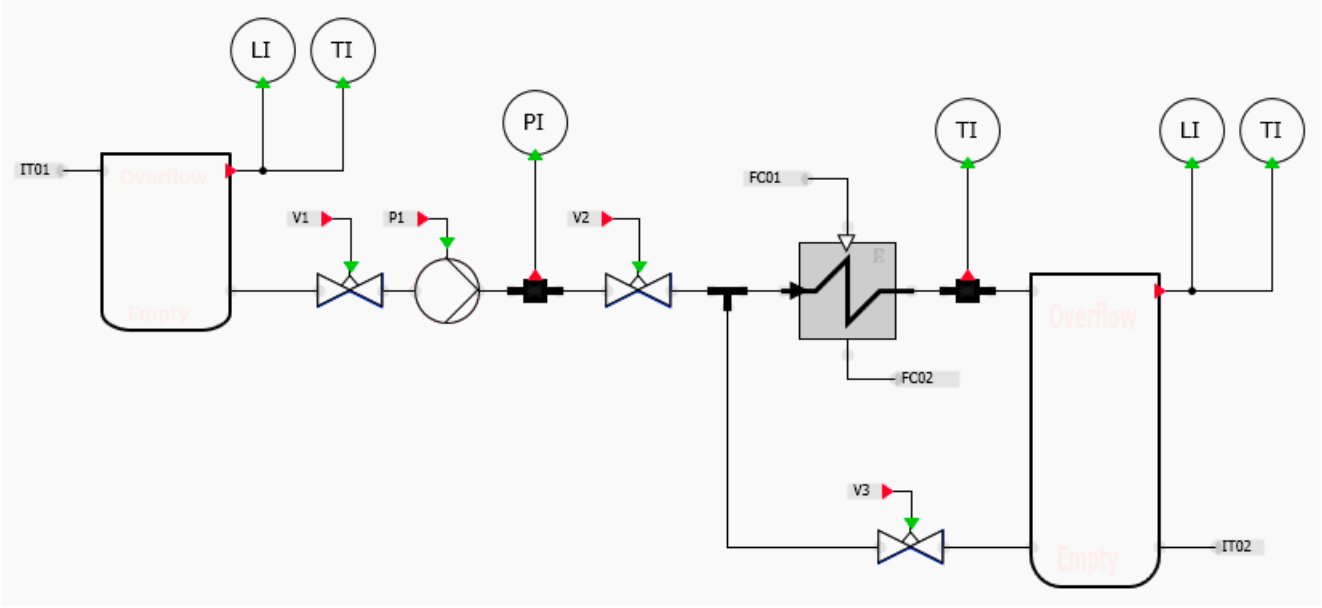
Separator – Abscheider (Seite 701)

StorageTankLiquid – Speichertank für flüssige Medien (Seite 725)

8.2.4 Flussnetze

Ein Flussnetz in SIMIT ist eine Verschaltung von Flussnetzkomponenten zur Simulation von thermodynamischen Vorgängen in Rohrleitungsnetzen. Die Simulation von Flussnetzen basiert auf einem speziellen Lösungsverfahren, das über Flussnetzkomponenten konfiguriert und parametriert wird. Der weiter unten beschriebene Modellierungsansatz schränkt die Flussnetze zwar auf homogene Medien ein, ist aber anwendbar für Flüssigkeiten, ideale Gase oder Wasser in den Aggregatzuständen Wasser und Dampf.

Die Bibliothek FLOWNET stellt Komponententypen zur Verfügung, mit denen Flussnetze konfiguriert, d. h. modelliert werden können. Die Modellierung von Flussnetzen erfolgt – wie in SIMIT üblich – im Diagrammeditor. Das dabei benutze Symbol der Flussnetzkomponententypen wie beispielsweise Ventil oder Pumpe hat eine Form, wie sie üblicherweise in Rohrleitungsschemata verwendet wird. Somit lässt sich ein Flussnetzmodell, wie in der folgenden Abbildung gezeigt, einfach in Form eines Rohrleitungsschemas aus den Symbolen der Komponententypen zusammensetzen:



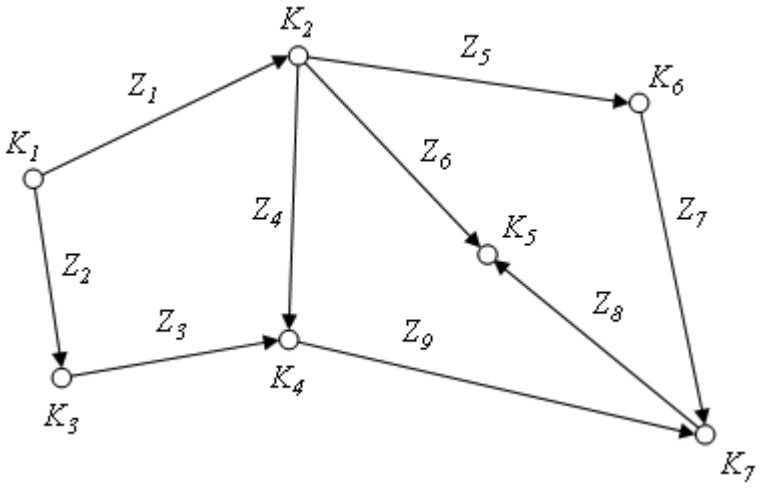
Aus der Zusammenschaltung von Flussnetzkomponenten wird die Topologie des Flussnetzes zur Strukturierung des Flussnetzlösers abgeleitet. Zur Laufzeit der Simulation tauschen die Flussnetzkomponenten und der Flussnetzlöser dann Daten aus: berechnete Werte oder auch Flussnetzparameter.

8.2.4.1 Grundlagen der Flussnetze

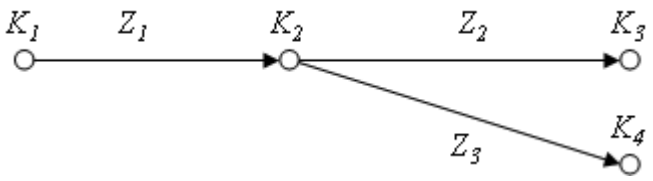
Das Verfahren zur Simulation von Rohrleitungsnetzen beruht darauf, dass das Flussnetz, das aus der Verschaltung von Flussnetzkomponenten resultiert, auf einen Graphen aus Knoten und Zweigen abgebildet wird. Die Zweige modellieren Fließstrecken, die Knoten die Abzweige, d. h. die Zusammenführungen oder Verzweigungen von Fließstrecken. Bestimmende Größen für die Knoten sind die Drücke und spezifischen Enthalpien und für die Zweige die Strömung (Massenströme).

Alle physikalischen Größen mit Vektorcharakter, wie beispielsweise Materieströme werden im Flussnetz als eindimensionale Größen mit Richtungsinformation, d. h. als gerichtete Größen (Linienströme) behandelt. Die Richtung wird durch das Vorzeichen ausgedrückt. Das Bezugssystem zur numerischen Beschreibung der Größen kann willkürlich gewählt werden.

Flussnetze können durch einen Graphen dargestellt werden, dessen Zweige (Kanten) aus den Rohrleitungen mit ihren Armaturen (Ventile, Pumpen etc.) und dessen Knoten aus den Rohrverzweigungen gebildet werden. Der Graph ist gerichtet, d. h. die Richtung seiner Zweige gibt die Bezugsrichtung für die Strömung vor. Weiter ist der Graph wie das Flussnetz zusammenhängend. Der Graph repräsentiert damit die Topologie des Flussnetzes auf der Ebene von Rohrleitungen und Rohrverzweigungen. In der folgenden Abbildung ist beispielhaft ein Graph mit 7 Knoten K_i und 9 Zweigen Z_i dargestellt:



Ein Graph, wie in der folgenden Abbildung zu sehen, mit drei Zweigen und vier Knoten ergibt sich aus dem Beispiel im Kapitel: Flussnetze (Seite 614)



Die Knoten K_1, K_3 und K_4 sind Knoten, über die Randbedingungen für das Flussnetz gesetzt werden. Hier wird über diese sogenannten externen Knoten beispielsweise der Druck an den Anschlüssen der beiden Behälter vorgegeben. Der Knoten K_2 ist im Unterschied dazu ein interner Knoten, für den die relevanten Größen, wie beispielsweise der Druck, mit Hilfe des Flussnetzlösungsverfahrens berechnet werden.

Das Lösungsverfahren für Flussnetze beruht darauf, dass die Strömung in den Zweigen abhängig vom Druck mit Hilfe der Impulsbilanz angesetzt wird und in den Knoten die Materie- und Enthalpieströme bilanziert werden. Die Zustandsgrößen für das so angesetzte System sind folglich die Massenströme in den Zweigen und die Drücke und spezifischen Enthalpien in den Knoten. Entsprechend dem jeweils betrachteten Medium können daraus weitere Größen, wie beispielsweise Dichten und Temperaturen im Flussnetz abgeleitet werden.

Falls weder der Durchfluss noch die Enthalpie der Strömung in einem Zweig durch die in diesem Zweig liegenden Flussnetzkomponenten verändert werden, wird dieser Zweig im Flussnetzgraphen eliminiert, d. h. seine beiden Knoten werden in einem Knoten zusammengefasst.

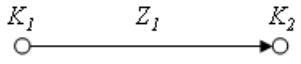
Das Lösungsverfahren für Flussnetze ist ein zyklisches Lösungsverfahren mit äquidistanten Zeitscheiben, um das Standardverfahren zu ergänzen. Flussnetzkomponententypen können also ergänzend zu anderen Komponententypen, wie beispielsweise den Komponententypen der Basisbibliothek eingesetzt werden. Für den Datenaustausch sind Flussnetzkomponenten an den Flussnetzlöser über spezifische Verbindungen angebunden. Sie erhalten über diese

Verbindungen beispielsweise Werte, die vom Flussnetzlöser berechnet worden sind und geben ihrerseits daraus berechnete Größen wieder an den Flussnetzlöser zurück.

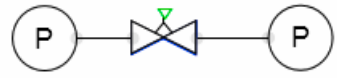
Hinweis

Abhängig von der Struktur des Flussnetzes und der Parametrierung der Flussnetzkomponenten kann der Fall eintreten, dass die Berechnung des Flussnetzes während der Simulation instabil wird und infolgedessen Werte von Zustandsgrößen des Flussnetzes über alle Grenzen wachsen können. Eine Prüfung der Stabilität von Flussnetzen erfolgt nicht in SIMIT. Sie können in diesen Fällen mit einer Verringerung der Zykluszeit für das Flussnetz und/oder einer Anpassung der Bilanzierungsparameter Stabilität herstellen.

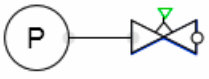
Offensichtlich muss ein Flussnetz mindestens aus einem Zweig mit zwei Knoten bestehen. Den minimalen Graphen zeigt die folgende Abbildung.



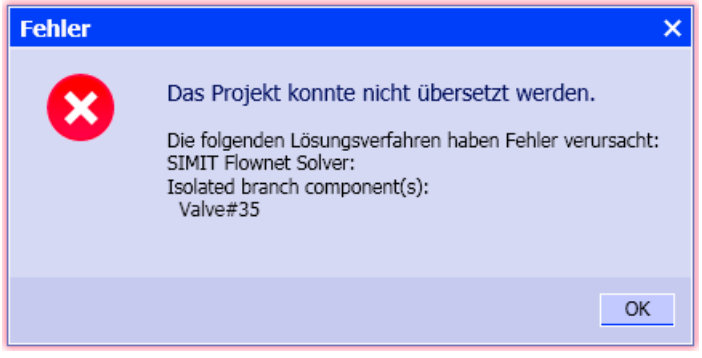
Die folgende Abbildung zeigt ein dazu korrespondierendes minimales Flussnetz. Die Knoten können externe (wie in der Abbildung) oder interne Knoten sein.



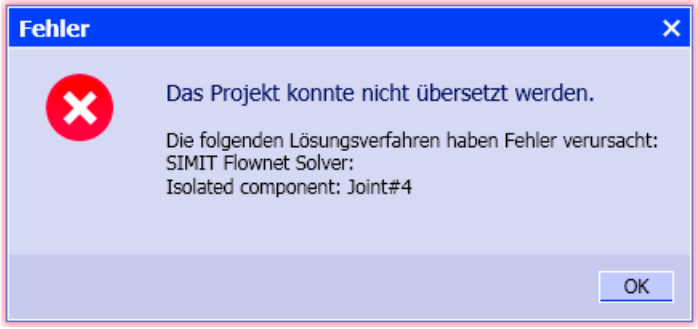
Falls, wie in der folgenden Abbildung zu sehen, ein Zweig nicht mit Knoten abgeschlossen ist, wird beim Starten der Simulation eine Fehlermeldung ausgegeben und die Simulation wird nicht gestartet.



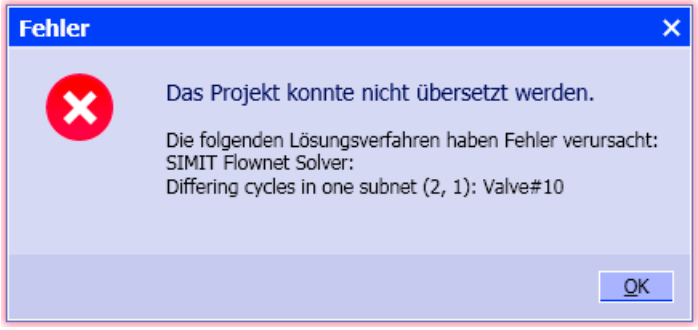
Die folgende Abbildung zeigt diese Fehlermeldung "*Isolated branch component(s)*" mit Angabe der Komponenten in diesem Zweig.



Eine entsprechende Fehlermeldung erscheint auch, falls für das Flussnetz lediglich eine isolierte Komponente existiert. Der damit korrespondierende Graph besteht dann nur aus einem Zweig oder Knoten und ist folglich kein minimaler Flussnetzgraph.



Auch die Flussnetzkomponenten und der Flussnetzlöser werden zyklisch in SIMIT abgearbeitet. Den Flussnetzkomponenten muss dazu eine Zeitscheibe zugeordnet sein. Dabei müssen alle Komponenten eines Flussnetzes mit der gleichen Zeitscheibe parametriert sein. Andernfalls wird der Start der Simulation mit einer Fehlermeldung wie in der folgenden Abbildung abgebrochen.



8.2.4.2 In Flussnetzen verwendete Größen

In Flussnetzen werden als physikalische Basisgrößen die Massenströme, Drücke und spezifischen Enthalpien sowie daraus abgeleitet die Dichten und Temperaturen betrachtet. In der folgenden Tabelle sind diese Größen mit den hier verwendeten Formelzeichen und Einheiten angegeben.

Tabelle 8-41 Flussnetzgrößen

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s
Druck	p	bar (absolut)
spezifische Enthalpie	h	kJ/kg
Dichte	ρ	kg/m³
Temperatur	T	°C

8.2.4.3 Modellierung der Zweige von Flussnetzen

Es wird vorausgesetzt, dass Zweige keine Masse einspeichern. Der Massenstrom und die Dichte sind dann einheitlich im gesamten Zweig.

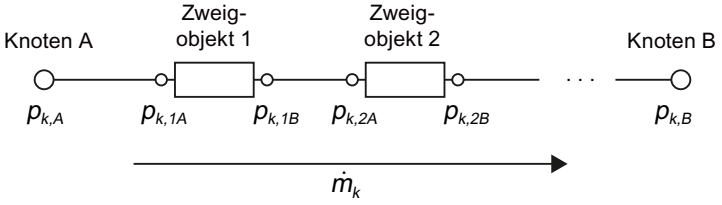
Diese Voraussetzungen sind für inkompressible Medien stets gegeben. Für kompressible Medien wird keine Masse im Zweig gespeichert, wenn die Zweige "kein" Volumen besitzen. Die Dichte nimmt zwar für kompressible Medien in Richtung fallender Drücke, also in Fließrichtung ab, für schwache Drosselung ist die Dichteänderung aber gering, die Annahme konstanter Dichte ist damit tragbar.

Für Zweigobjekte sind folglich lediglich die Drücke an den Anschlusspunkten zu betrachten. Die Beziehungen zwischen den Drücken beiderseits eines Zweigobjekts und dem Massenstrom im Zweig, wie z. B. $\rho\,\Delta p \sim \dot{m}^2$, sind dabei rein algebraischer Natur.

Für jeden Zweig *k* wird die Impulsbilanz mit Druckkräften angesetzt. Reibungskräfte, Beschleunigungskräfte und die Schwerkraft werden vernachlässigt. Mit einem einheitlichen Querschnitt A_k im Zweig der Länge L_k gilt dann

$$L_k \frac{d\dot{m}_k}{dt} = A_k \Delta p_k - A_k \sum_{\kappa} \Delta p_{k,\kappa}$$

wobei $\Delta p_k = p_{k,A} - p_{k,B}$ die über dem gesamten Zweig *k* anliegende Druckdifferenz ist und $\Delta p_{K,k} = p_{K,kA} - p_{K,kB}$ die Druckdifferenzen der einzelnen Zweigelemente sind (s. folgende Abbildung).



Wird der Druck in bar eingesetzt, dann ergibt sich die Änderung des Massenstromes $\dot{m}_k$ zu

$$\frac{d\dot{m}_k}{dt} = 10^5 \frac{A_k}{L_k} \left(\frac{\Delta p_k}{\text{bar}} - \sum_{\kappa} \frac{\Delta p_{k,\kappa}}{\text{bar}} \right) \frac{\text{kg}}{\text{ms}^2}$$

Die Länge und der Querschnitt eines Zweiges sind im Allgemeinen unbekannt und müssen plausibel geschätzt werden. Nimmt man beispielsweise einen Querschnitt von 0,05 m^2 und eine Länge von 10 m an, dann ergibt sich ein im weiteren als Impulsfaktor bezeichneter Faktor

$$A_k = 10^5 \frac{A_k}{L_k}$$

von 500 m.

8.2.4.4 Modellierung der Knoten von Flussnetzen

In jedem Knoten werden dynamisch die Zu- und Abflüsse des Mediums bilanziert. Jedem Knoten wird dazu ein Bilanzraum, d. h. ein Volumen zugeordnet. Bilanziert werden die zu- und abfließenden Massenströme und, als Maß für die Energieumsetzung, die zu- und abfließenden Enthalpieströme.

Massenbilanz für den Knoten

Der Druck in einem Knoten *i* wird über die Massenbilanz, d. h. die Bilanzierung der Zu- und Abflüsse aus mit dem Knoten verbundenen Zweigen ermittelt:

$$V_i \frac{d\rho_i}{dt} = \sum_{\kappa} \dot{m}_{\kappa}$$

V_i ist das Volumen des dem Knoten zugeordneten Bilanzraums, ρ_i die Dichte des Mediums im Bilanzraum und die $\dot{m}_{\kappa}$ sind die Zu-/Abflüsse. Zuflüsse werden positiv ($\dot{m}_{\kappa} > 0$) angesetzt, Abflüsse negativ ($\dot{m}_{\kappa} < 0$).

Mit der Zustandsgleichung ρ= ρ(*p,h*) folgt unter Annahme einer isenthalpen Zustandsänderung für die Drücke in den Knoten

$$\frac{dp_i}{dt} = \left(V_i \frac{d\rho_i}{dp_i}\bigg|_{h=\text{const}} \right)^{-1} \sum_{\kappa} \dot{m}_{\kappa}$$

Der Term

$$V_i \frac{d\rho_i}{dp_i}$$

ist ein Maß für die Kompressibilität des Mediums. Mit dem Kompressionsmodul K_i

$$K_i = M_i \left(V_i \frac{d\rho_i}{dp_i} \right)^{-1}$$

gilt somit für die Massenbilanz

$$\frac{dp_i}{dt} = \frac{K_i}{M_i} \sum_{\kappa} \dot{m}_{\kappa} = c_i \sum_{\kappa} \dot{m}_{\kappa}$$

In der Voreinstellung des Flussnetzlösungsverfahrens wird das spezifische Kompressionsmodul $c_i = K_i / M_i$ für alle Knoten eines Flussnetzes gleichgesetzt. Dabei wird aber die gegenüber Flüssigkeiten größere Kompressibilität von Gasen und Dämpfen durch unterschiedliche Faktoren c_i für Medien mit hoher und niedriger Dichte berücksichtigt. Weitere Informationen hierzu finden Sie im Kapitel: Parametrierung von Flussnetzen (Seite 623).

Enthalpiebilanz für den Knoten

Die spezifische Enthalpie wird als weitere Zustandsgröße eines Knotens behandelt. Im Prinzip wird aus der konvektiv zufließenden Enthalpie und der im Knoten bereits enthaltenen Enthalpie eine Mischenthalpie gebildet, mit der die abfließenden Strömungen ausgestattet werden. Die Bilanzierung für einen Knoten *i* erfolgt gemäß

$$\frac{d(h_i M_i)}{dt} = \sum_{\kappa} h_\kappa \dot{m}_\kappa$$

wobei M_i die Masse und h_i die spezifische Enthalpie des Mediums im Bilanzraum sind. Es ergibt sich daraus gemäß

$$\frac{dh_i}{dt} = \frac{1}{M_i} \sum_{\kappa \in Z_i} \dot{m}_\kappa (h_\kappa - h_i)$$

dass lediglich über die Enthalpiedifferenz der Zuflüsse in den Knoten $i\,(k \in Z_i)$ summiert werden muss.

Wie die Faktoren c_i der Massenbilanzen werden die thermischen Faktoren $m_i = 1 / M_i$ in der Voreinstellung des Flussnetzlösungsverfahrens einheitlich für alle Knoten gesetzt, aber mit unterschiedlichen Werten für Medien mit hoher und niedriger Dichte.

Bestimmung der Dichte des Mediums in den Knoten

Die Werte für die Dichte in den Knoten werden aus den Werten für den Druck und der spezifischen Enthalpie berechnet. Die verwendeten Beziehungen sind abhängig vom Medium im Flussnetz.

Wasser/Dampf als Medium

Für Wasser/Dampf wird die Dichte in den Knoten aus dem Druck *p* und der spezifischen Enthalpie *h* aus der Zustandsgleichung für Wasser/Dampf berechnet:

$\rho = \rho(p,h)$.

Flüssigkeit als Medium

Bei Flüssigkeiten wird mit einer konstanten Dichte im gesamten Flussnetz gerechnet. Voreingestellt ist eine Dichte von 997,337 kg/m^3.

Berechnung

Für ideales Gas wird die Gasgleichung $pV = mR_s T$ angesetzt. Mit der spezifischen Wärmekapazität c_p folgt daraus für die Dichte

$$\rho = \frac{m}{V} = \frac{p \cdot 10^2 \frac{\text{Pa}}{\text{bar}}}{R_S\left(\frac{h - h_0}{c_p} + T_0\right)}$$

mit dem Druck p in bar, der spezifische Wärmekapazität c_p in kJ/kgK und der spezifischen Gaskonstante R_S in kJ/kgK. Wird für den Nullpunkt (T_0, h_0) der Tripelpunkt von Wasser angesetzt, dann können T_0 = 273,15 K und h_0 = 0 gesetzt werden. Die Dichte berechnet sich dann aus der Beziehung

$$\rho = \frac{p \cdot 10^2 \frac{\text{Pa}}{\text{bar}}}{R_S\left(\frac{h}{c_p} + 273{,}15\ \text{K}\right)}$$

Im Flussnetzlösungsverfahren ist die spezifische Gaskonstante mit dem Wert von 0,287 kJ/kgK (spezifische Gaskonstante von trockener Luft) eingestellt.

Bestimmung der Temperatur des Mediums in den Knoten

Auch die in den Komponenten der FLOWNET-Bibliothek verwendeten Temperaturen werden aus den Werten für den Druck p und spezifischer Enthalpie h berechnet. Für Wasser/Dampf als Medium wird dazu die Zustandsgleichung

$T = T(p,h)$

verwendet. Für ideale Gase und Flüssigkeiten wird die Temperatur mit Hilfe der spezifischen Wärmekapazität c_p aus der spezifischen Enthalpie gemäß

$T = h / c_p$

bestimmt.

8.2.4.5 Wärmeaustausch mit der Umgebung

Der Wärmeaustausch mit der Umgebung wird durch einen entsprechenden Term in der Enthalpiebilanz für den Knoten gemäß folgender Formel berücksichtigt:

$$\frac{dh_i}{dt} = \frac{1}{M_i}\left(\sum_{K \in Z_i} \dot{m}_K (h_K - h_i) - c_i (T_i - T_{Env})\right)$$

Dabei ist T_{Env} die Umgebungstemperatur, T_i die Temperatur des Knotens und

$c_i = \alpha_i A_i$

der den Wärmeübergang bestimmende Faktor als Produkt aus dem Wärmeübergangskoeffizienten α_i und der Wärmeaustauschfläche A_i.

Zur Berechnung des Wärmeaustausches mit der Umgebung verwenden Sie als Knoten die Komponente *JointParam*.

8.2.4.6 Parametrierung von Flussnetzen

Die Größen für die Zweige und Knoten eines Flussnetzes werden unter Zuhilfenahme der oben beschriebenen Parameter berechnet. Diese Parameter sind im Flussnetzlösungsverfahren mit entsprechenden Werten voreingestellt, können aber mit Hilfe von speziellen Komponententypen der Bibliotheken FLOWNET und CHEM-BASIC verändert werden.

Komponententypen

Zur Parametrierung von Netzen können Sie die folgenden Komponententypen verwenden:

- *NetParam* zur allgemeinen Parametrierung von Flussnetzen in FLOWNET (NetParam – Netzparametrierung (Seite 751)) und in CHEM-BASIC (NetParam – Netzparametrierung (Seite 704))
- *NetWS* zur Parametrierung von Netzen mit Wasser/Dampf als Medium in FLOWNET (NetWS – Netzparametrierung (Seite 760)),
- *NetLiquid* zur Parametrierung von Netzen mit Flüssigkeiten als Medium in FLOWNET (NetLiquid – Netzparametrierung (Seite 777)) und
- *NetGas* zur Parametrierung von Netzen mit idealem Gas als Medium in FLOWNET (NetGas – Netzparametrierung (Seite 790)).

Zur gezielten Parametrierung von einzelnen Knoten können Sie die folgenden Komponententypen verwenden:

- *JointParam* zur allgemeinen Parametrierung eines Knotens in FLOWNET (JointParam – parametrierbare Verzweigung (Seite 753)) und in CHEM-BASIC (JointParam – parametrierbarer Rohrleitungsknoten (Seite 634))
- *JointParamWS* zur Parametrierung eines Knotens in einem Netz mit Wasser/Dampf als Medium in FLOWNET (JointParamWS – parametrierbare Verzweigung (Seite 763)),
- *JointParamLiquid* zur Parametrierung eines Knotens in einem Netz mit Flüssigkeit als Medium in FLOWNET (JointParamLiquid – parametrierbare Verzweigung (Seite 780)),
- *JointParamGas* zur Parametrierung eines Knotens in einem Netz mit idealem Gas als Medium in FLOWNET (JointParamGas – parametrierbare Verzweigung (Seite 793)).

Zur Parametrierung von einzelnen Zweigen können Sie den Komponententyp verwenden:

- *BranchParam* in FLOWNET (BranchParam – Zweigparametrierung (Seite 752)) und in CHEM-BASIC (BranchParam – Zweigparametrierung (Seite 703))

Verbindungstypen

In den folgenden Abschnitten sind die Verbindungstypen beschrieben, mit denen Sie in FLOWNET in Ihren selbst erstellten Flussnetzkomponenten Parametrierungen nutzen können:

- Verbindungstyp FLN5 für Parameter eines Flussnetzes (Seite 811)
- Verbindungstyp FLN6 zur Parametrierung eines Zweiges (Seite 813)
- Verbindungstyp FLN7 zur Parametrierung eines internen Knotens (Seite 813)

Das Medium im Flussnetz

Für ein Flussnetz können folgende Medien vorgegeben werden:

- Wasser/Dampf (Water/Steam),
- Ideales Gas (Ideal Gas) oder
- Flüssigkeit (Liquid)

Voreingestellt ist Wasser/Dampf als Medium.

Parameter für Zweige

Für die Zweige im Flussnetz kann der Impulsfaktor A gesetzt werden. Im Flussnetzlösungsverfahren ist ein gleicher Faktor A = 450 m für alle Zweige voreingestellt. Ist der Impulsfaktor in einem Zweig mehrfach gesetzt, dann wird der im Zweig wirksame Faktor A gemäß

$$\frac{1}{A} = \sum_{k} \frac{1}{A_{\kappa}}$$

aus den *k* Faktoren A_k ermittelt.

Parameter für Knoten

Für die Knoten in einem Flussnetz können folgende Größen parametriert werden:

- Spezifischer Kompressionsmodul $c = K / M$
- Thermischer Faktor $m = 1 / M$

Beide Größen können getrennt für Wasser und Dampf sowie für Flüssigkeit und Gas vorgegeben werden.

Für Wasser/Dampf als Medium erfolgt eine Umschaltung zwischen den Parametern c_L und m_L und den Parametern c_G und m_G bei Berechnungen im Flussnetzlösungsverfahren anhand der Dichte des Mediums nach folgendem Schema:

$$c = \begin{cases} c_L & \text{für} \quad \rho \geq 500\ \text{m}^3/\text{kg} \\ c_G & \text{für} \quad \rho < 500\ \text{m}^3/\text{kg} \end{cases}$$

und

$$m = \begin{cases} m_L & \text{für} \quad \rho \geq 500\ \text{m}^3/\text{kg} \\ m_G & \text{für} \quad \rho < 500\ \text{m}^3/\text{kg} \end{cases}$$

Der Übergang zwischen den beiden Parameterwerten kann auch mit einer linearen Übergangsfunktion angesetzt werden, gemäß

$$c = \begin{cases} c_L & \text{für} \quad \rho > \rho_L \\ c_G & \text{für} \quad \rho < \rho_G \\ c_L + (c_G - c_L)\dfrac{\rho_G(\rho_L - \rho)}{\rho(\rho_L - \rho_G)} & \text{sonst} \end{cases}$$

und

$$m = \begin{cases} m_L & \text{für} \quad \rho > \rho_L \\ m_G & \text{für} \quad \rho < \rho_G \\ m_L + \dfrac{m_G - m_L}{v_G - v_L}(v - v_L) & \text{sonst} \end{cases}$$

mit den beiden Eckwerten ρ_G = 5 kg/m³ und ρ_G = 1000 kg/m³.

Für Flüssigkeit als Medium im Flussnetz sind immer die entsprechenden Größen c_L und m_L wirksam, für ideales Gas als Medium die Größen c_G und m_G.

Für den Wärmeaustausch mit der Umgebung können die Umgebungstemperatur T_{Env} und der Wärmeübergangsfaktor $c = \propto A / M$ parametriert werden. Voreingestellt ist die Temperatur T_{Env} = 20 °C und ein Faktor c = 0, d. h. in der Voreinstellung erfolgt kein Wärmeaustausch.

Parameter für Flüssigkeit als Medium

Wenn Flüssigkeit als Medium für das Flussnetz eingestellt ist, wird mit konstanter Dichte gerechnet. Der zu verwendende Dichtewert kann als Parameter vorgegeben werden.

Als weitere Größe kann die spezifische Wärmekapazität c_p der Flüssigkeit für das Flussnetz wie auch spezifisch für jeden einzelnen Knoten vorgegeben werden. Voreingestellt ist ein Wert von 4,18 kJ/kgK.

Parameter für Ideales Gas als Medium

Für ideales Gas als Medium des Flussnetzes können die Gaskonstante R_S und die spezifische Wärmekapazität c_p vorgegeben werden. Voreingestellt sind die Werte R_S = 0,287 kJ/kgK und c_p = 1 kJ/kgK.

Initialisierung von Größen

Mit jedem Start der Simulation werden die Zustandsgrößen des Flussnetzes wie folgt initialisiert:

Alle Zweigmassenströme werden mit null initialisiert ($\dot{m} = 0$). Diese Initialisierung kann nicht verändert werden. Die Drücke in den Knoten werden in der Voreinstellung mit p = 1 bar angesetzt, können aber über einen Parameter auf andere Werte gesetzt werden. Der Startwert h für die spezifische Enthalpie ist unterschieden nach dem Medium im Flussnetz wie folgt gegeben:

- h = 100 kJ/kg für Wasser/Dampf,
- h = 83,6 kJ/kg für Flüssigkeit und
- h = 20 kJ/kg für ideales Gas.

Diese Startwerte können durch einen Parameter verändert werden.

Parameterübersicht

In der folgenden Tabelle sind die zur Verfügung stehenden Flussnetzparameter zusammengefasst. Die Spalte Signal zeigt den Namen des Ein- oder Ausgangssignals der Komponente, die Spalte Bezeichnung den Namen, unter dem der Parameter im Eigenschaftsfenster von Komponenten zu finden ist. In der letzten Spalte wird die Vorbelegung nach Wert und Einheit angezeigt. Sind explizit Parameter nicht gesetzt, dann ist für diese die Vorbelegung für das Flussnetz wie oben wirksam.

Tabelle 8-42 Liste der Flussnetzparameter

Signal	Bezeichnung	Beschreibung	Vorbelegung	
			Wert	Einheit
MEDIUM	Medium	Kennzahl für das Medium im Flussnetz: • Wasser/Dampf (Water/Steam): 0 • Flüssigkeit (Liquid): 2 • Ideales Gas (Ideal Gas): 1	0	–
CG	sCompressionGas	Spezifischer Kompressionsmodul $c = K / M$ für ideales Gas oder Dampf	10	bar/kg
CL	sCompressionLiquid	Spezifischer Kompressionsmodul $c = K / M$ für Flüssigkeiten	100	bar/kg
MG	FactorThermalGas	Thermischer Faktor $m = 1 / M$ für ideales Gas oder Dampf	100	kg^{-1}
ML	FactorThermalLiquid	Thermischer Faktor $m = 1 / M$ für Flüssigkeiten	0,1	kg^{-1}
P_INIT	PressureInit	Initialwert für Druck	1	bar
H_INIT	sEnthalpyInit	Initialwert für spezifische Enthalpie	20 / 83,6 / 100	kJ/kg
DENSITY	Density(Liquid)	Dichte (nur wirksam für Flüssigkeit)	997,337	kg/m^3
T_ENV	TemperatureEnvironment	Umgebungstemperatur T_{Env}	20	°C
C_ENV	FactorHeatExchangeEnv	Wärmeübergangsfaktor $c = \alpha A$	0	kW/K
L_CR	sHeatCapLiquid	spezifische Wärmekapazität c_p für Flüssigkeiten	4,18	kJ/kgK
IG_R	GasConstant	Gaskonstante R_S	0,287	kJ/kgK
IG_CR	sHeatCapGas	spezifische Wärmekapazität c_p für Gase	1,0	kJ/kgK

Signal	Bezeichnung	Beschreibung	Vorbelegung	
ST	SmoothTransition	Umschaltung auf linearen Übergang des Kompressionsmoduls und thermischen Faktor für Wasser/Dampf	False	
AL	FactorMomentum	Impulsfaktor A	450	m

8.2.4.7 Kopplung Simulationsmodell mit Aktor-Sensor-Ebene

Kopplung Simulationsmodell mit Aktor-Sensor-Ebene

Das auf der Basis der Komponententypbibliotheken CHEM-BASIC und FLOWNET erstellte physikalische Simulationsmodell wird mit einer Leittechnik über eine Aktor-Sensor-Ebene gekoppelt.

In der Aktor-Sensor-Ebene werden Signale aus der Leittechnik über spezielle Antriebskomponenten zur Bildung der von der Leittechnik benötigten Rückmeldungen benutzt. Hierzu können beispielsweise die Antriebskomponententypen der Basisbibliothek von SIMIT eingesetzt werden.

Zusätzlich berechnen diese Antriebskomponenten beispielsweise die Stellung für ein Ventil oder die normierte Drehzahl für eine Pumpe.

Die physikalischen Werte von Messstellen werden von der Aktor-Sensor-Ebene aus der Simulationsebene gelesen. Diese Werte werden dann in die von der Leittechnik benötigten Formate aufbereitet und weitergegeben.

Die Aktor-Sensor-Ebene wird somit wie folgt mit der Simulationsebene verbunden:

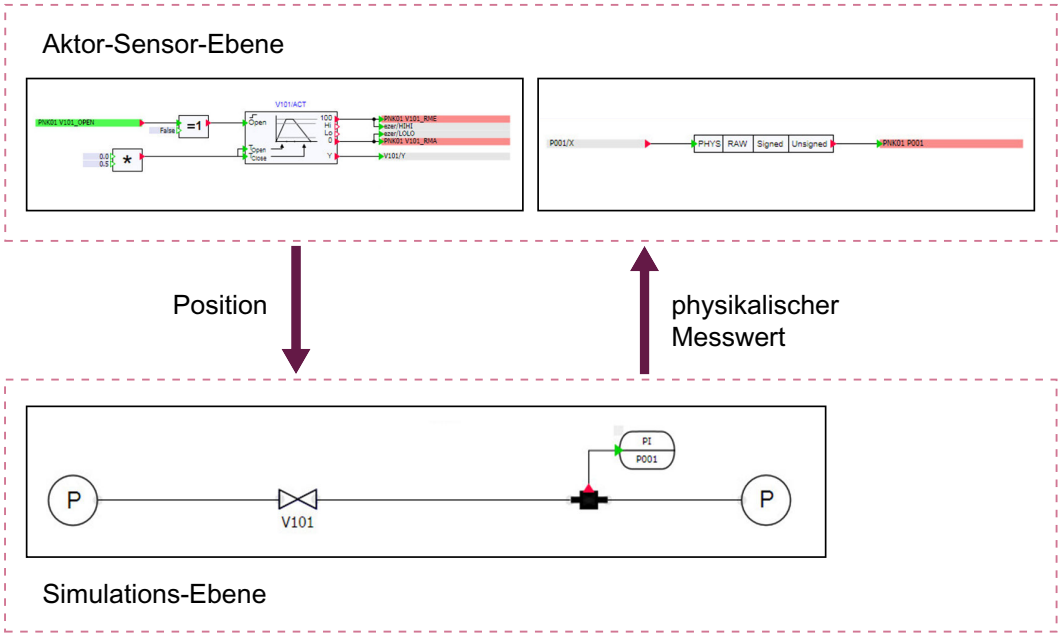


Bild 8-5 Schematische Kopplung der Aktor-Sensor-Ebene mit der Simulationsebene

Zur Kopplung der Simulationsebene mit der Aktor-Sensor-Ebene bestehen folgende Möglichkeiten:

- Direkte Verschaltung über Signallinien (nur möglich wenn Aktor-Sensor-Ebene und Simulationsebene in einem Plan realisiert werden)
- Verschaltung der Komponenten über Signallinien unter Nutzung von globalen Konnektoren
- Implizite Verschaltung der Aktor-Sensor-Ebene mit der Simulationsebene

Direkte Verschaltung über Signallinien

Die direkte Verschaltung der Aktor-Sensor-Ebene mit der Simulationsebene über Signallinien ist die einfachste Art der Verschaltung. Bei dieser Verschaltung müssen alle zueinander gehörenden Teile beider Ebenen jeweils auf einem Plan liegen. Dies begrenzt die Anwendbarkeit dieses Verschaltungstyps.

Eine Generierung der Aktor-Sensor-Ebene ist aufgrund der engen Verknüpfung beider Ebenen auf einem Plan nicht praktikabel.

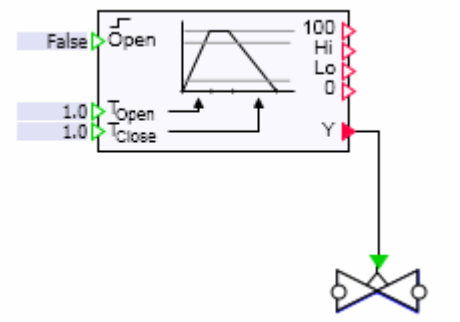


Bild 8-6 Kopplung des Komponententyps Valve mit einem Antrieb

Direkte Verschaltung über Signallinien auf unterschiedlichen Plänen

Eine performantere Lösung der Verkopplung beider Ebenen ist die Verschaltung über Signallinien unter Nutzung von Konnektoren auf unterschiedlichen Plänen. Hierbei ist eine automatisierte Generierung der Aktor-Sensor-Ebene in Verbindung mit einer manuell erstellten Simulationsebene möglich. Die automatisierte Generierung kann beispielsweise über einen der in SIMIT integrierten Tabellenimporte unter Nutzung der SIMIT-Standard-Templates erfolgen. Gespeist werden diese Tabellenimporte aus PCS 7-Planungsdaten.

Der wesentliche Vorteil im Vergleich zur direkten Verschaltung ohne Konnektoren liegt darin, dass die Aktor-Sensor-Ebene in einer komplett eigenen Ordnerstruktur liegen kann und dort leicht ausgetauscht oder ergänzt werden kann.

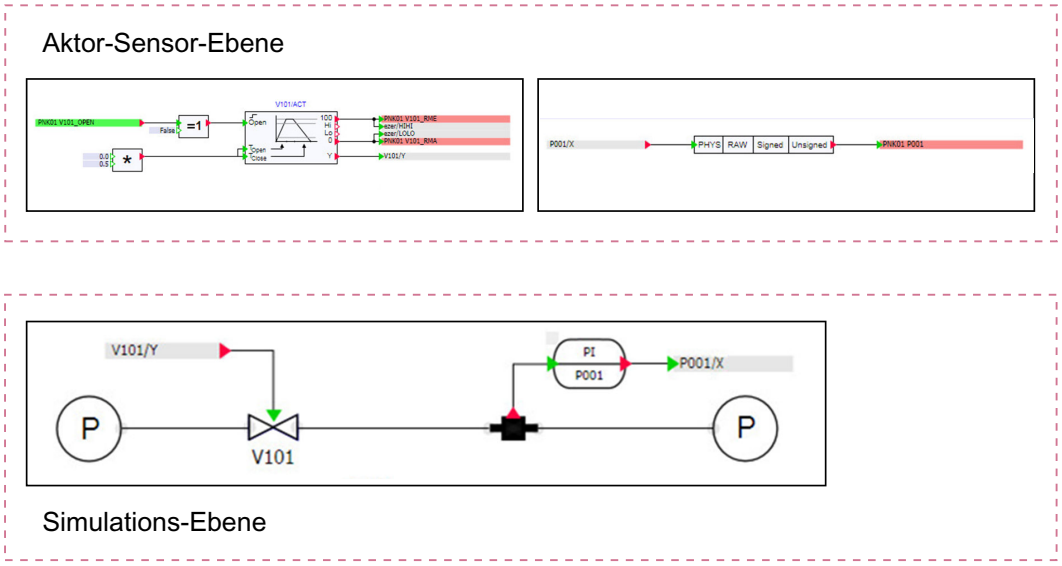


Bild 8-7 Kopplung der Aktor-Sensor-Ebene mit der Simulationsebene mittels Signallinien und Konnektoren

Verschaltung über implizite Kopplung

Bei der Verschaltung der beiden Ebenen über implizite Kopplung können diese ebenfalls auf unterschiedlichen Plänen liegen. Es ist sinnvoll, diese Pläne auch in unterschiedlichen Ordnern zu verwalten.

Die wesentlichen Vorteile dieser Art der Kopplung im Vergleich zur Kopplung mit Signallinien und Konnektoren sind folgende:

- Bessere Übersichtlichkeit der Pläne innerhalb der Simulationsebene durch einfachere Darstellung (keine Konnektoren notwendig)
- Die Eingabe des Anlagenkennzeichens der Komponenten ist nur an einer Stelle notwendig (Name des Ventils, der Pumpe oder der Messstelle)
- Im Modell eingestellte Parameter (z. B. Ventilparameter) sind automatisch mit dem entsprechenden AKZ verknüpft.

Die Verschaltung über implizite Kopplung wird durch Namensgleichheit erreicht. Am lesenden Eingang einer Komponente muss hierzu der Name und der Ausgang der schreibenden Komponente angegeben werden.

Zur Unterstützung dieser Kopplungsart bei Pumpen und Ventilen wird der Eingang *Position* implizit vorbelegt. Diese Vorbelegung ist wie folgt definiert:

Name der zu koppelnden Komponente:	"Ventilname"/ACT
Name des zu koppelnden Ausgangs:	Y

Zur Nutzung der beschriebenen impliziten Kopplung wird somit die Vergabe des Namens "Ventilname/ACT" für den entsprechenden Antrieb in der Aktor-Sensor-Ebene empfohlen. Dazu werden die Templates entsprechend gestaltet.

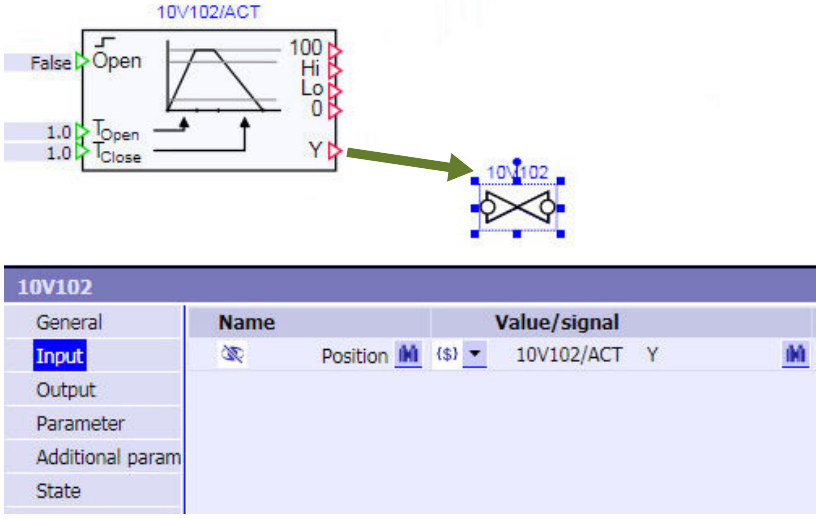


Bild 8-8 Implizite Kopplung des Komponententyps Valve mit einem Antrieb

Bei der impliziten Kopplung von Messstellen müssen die entsprechenden lesenden Bausteine in der Aktor-Sensor-Ebene über Templates initialisiert werden.

8.2.5 Komponenten der Bibliothek CHEM-BASIC

8.2.5.1 Burner

Symbol

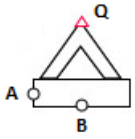


Bild 8-9 Symbol des Komponententyps Burner

Der Komponententyp *Burner* bildet vereinfacht die Funktion eines Brenners nach.

Die Komponente wird von einem Brennstoff durchströmt.

Die Komponente berechnet die bei einer Verbrennung freiwerdende Brennleistung und stellt diese am Ausgang Q zur Verfügung. Diese Brennleistung kann zur Erwärmung des in anderen Komponenten (z. B. *Storage TankLiquid* oder *PressureStrainer*) gespeicherten Inhalts verwendet werden.

Funktion

Die Komponente *Burner* ist über die Anschlüsse *A* und *B* mit einem Flussnetz verbunden.

Im Fall einer Zündung des Brennstoffes (über Eingang *Ignation* oder Button "Ignation" im Bedienfenster) wird der im Falle einer vollständigen Verbrennung des Brennstoffs

freiwerdende Wärmestrom am Ausgang *Q* ausgegeben. Der Wärmestrom wird aus dem anzugebenden Heizwert und dem Massenstrom berechnet. Typische Heizwerte sind:

Erdgas	53 MJ/kg
Heizöl	45 MJ/kg
Benzin	34 MJ/kg
Rohbraunkohle	10 MJ/kg

Eine tatsächliche stoffliche Umsetzung des Brennstoffs erfolgt nicht. Der Massenstrom und die Enthalpie des Brennstoffs im Brenner werden nicht beeinflusst.

Sobald der Brennstoffmassenstrom auf null fällt, erlischt die Flamme, und es wird kein Wärmestrom mehr abgegeben.

Der Komponententyps *Burner* besitzt unterschiedliche Symbole in Abhängigkeit vom Flammenstatus:

Brenner ohne Flamme



Brenner mit Flamme



Parameter

Parametername	Beschreibung	Einheit	Standardwert
HeatingValue	Heizwert des Brennstoffs in MJ/kg	MJ/kg	10.0
Kvs	Durchflusskoeffizient k_{VS} mit $k_{VS} \geq 10^{-6}$ m^3/h	m^3/h	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m^3/h	m^3/h	0.000001

8.2.5.2 Fittings

ChangeCrossSection – Querschnittsänderung

Symbol



Bild 8-10 Symbol des Komponententyps ChangeCrossSection

Funktion

Die Komponente *ChangeCrossSection* berechnet den Druckverlust in einer Rohrleitung bei Querschnittsänderung. Parametriert wird die Komponente über den Durchmesser 1 auf der Anschlussseite D1 und Durchmesser 2 auf der Anschlussseite D2.

Bei der Druckverlustrechnung wird von einer stetigen Querschnittsänderung mit einem Winkel von 4° ausgegangen. Je nach Größenverhältnis der Durchmesser arbeitet die Komponente daher als Düse (Durchmesser 1 > Durchmesser 2) oder Diffusor (Durchmesser 1 < Durchmesser 2).

Darstellung

Das Symbol passt sich entsprechend an (Strömungsrichtung von D1 nach D2).

Düse



Diffusor



Parameter

Parametername	Beschreibung	Einheit	Standardwert
Diameter1	Rohrdurchmesser auf Anschlussseite D1	mm	200
Diameter2	Rohrdurchmesser auf Anschlussseite D2	mm	300

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckverlust	Δp	mbar

Nozzle – Stutzen

Symbol

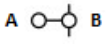


Bild 8-11 Symbol des Komponententyps Nozzle

Funktion

Mit dem Komponententyp *Nozzle* wird der Druckverlust eines Stutzens aufgrund von Querschnittsveränderung oder Einlaufeffekten an Apparaten abgebildet. Sie können diesen Typ zum Anschluss von Rohrleitungen an folgende Komponententypen nutzen:

- Tanks
- Wärmetauscher
- Zentrifuge
- Abscheider

Sie können Rohrleitungen an die genannten Komponenten prinzipiell auch ohne den Komponententyp *Nozzle* anschließen. Die Nozzle bietet Ihnen jedoch die Möglichkeit, den Druckverlust des Stutzens über den Parameter *Kvs* zu parametrieren.

Der im Komponententyp *Nozzle* berechnete Druckabfall ist abhängig von:

- Größe der Nozzle (k_{VS}-Wert)
- Durchströmung (Volumenstrom, Medium)

Der Druckabfall über der Nozzle wird gemäß folgender Formel berechnet.

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Dabei ist

$\Delta p = p_B - p_A$	der Druckabfall über der Nozzle in bar
$\dot{m}$	der Durchfluss bzw. Massenstrom in kg/s
ρ	die Dichte des Mediums in kg/m³
k_{VS}	der Durchflusskoeffizient der Nozzle in m³/h

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Durchflusskoeffizient k_{VS} mit $k_{VS} \geq k_{V0} + 10^{-6}$ m³/h	m³/h	10.0

Beispielverschaltung

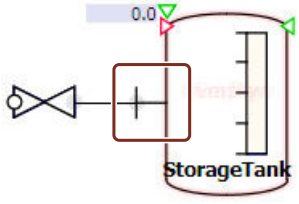


Bild 8-12 Beispielverschaltung Komponententyp Nozzle mit StorageTankLiquid

Joint – Rohrleitungsknoten

Symbol



Bild 8-13 Symbol des Komponententyps Joint

Funktion

Über den Komponententyp *Joint* können Sie die drei an seinen Anschlüssen *A*, *B* und *C* angeschlossenen Zweige in einem Knoten zusammenführen. Mit der Komponente *Joint* wird ein interner Knoten in das Flussnetz eingefügt.

Die zur Parametrierung dieses Knotens notwendigen Parameter werden vom entsprechenden Baustein *NetParam* des Flussnetzes gelesen. Somit haben alle im selben Flussnetz befindlichen Knoten dieses Typs die gleiche Parametrierung.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³
Temperatur	T	°C

JointParam – parametrierbarer Rohrleitungsknoten

Symbol



Bild 8-14 Symbol des Komponententyps JointParam

Funktion

Über den Komponententyp *JointParam* können Sie die drei an seinen Anschlüssen *A*, *B* und *C* angeschlossenen Zweige in einem Knoten zusammenführen. Mit der Komponente *JointParam* wird ein interner Knoten in das Flussnetz eingefügt. Im Gegensatz zum Komponententyp *Joint* wird jede Instanz des Komponententyps *JointParam* einzeln parametriert.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
sCompressionGas	Spezifischer Kompressionsmodul für Medien niedriger Dichte (Gase/Dämpfe, ρ < 500 kg/m³)	bar/kg	10.0
sCompressionLiquid	Spezifischer Kompressionsmodul für Medien hoher Dichte (Flüssigkeiten, ρ > 500 kg/m³)	bar/kg	100.0
FactorThermalGas	Faktor für die Enthalpiebilanzierung für Medien niedriger Dichte (Gase/Dämpfe, ρ < 500 kg/m³)	1/kg	100.0
FactorThermalLiquid	Faktor für die Enthalpiebilanzierung für Medien hoher Dichte (Flüssigkeiten, ρ > 500 kg/m³)	1/kg	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck im Knoten	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie im Knoten	kJ/kg	100.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums im Knoten mit der Umgebung; bei einem Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³
Temperatur	T	°C

Orifice – Blende

Symbol



Bild 8-15 Symbol des Komponententyps Orifice

Funktion

Die Komponente *Orifice* berechnet den Druckverlust einer Blende in einer Rohrleitung. Parametriert wird die Komponente über den Durchmesser der Rohrleitung *DiameterPipe* und den Durchmesser der Blende *DiameterOrifice*.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
DiameterPipe	Durchmesser der Rohrleitung	mm	200
DiameterOrifice	Durchmesser der Blende	mm	40

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckverlust	Δp	mbar

Pipe – Rohrleitung

Symbol



Bild 8-16 Symbol des Komponententyps Pipe

Funktion

Die Komponente *Pipe* berechnet den Druckverlust in einer Rohrleitung. Parametriert wird die Rohrleitung über die Rohrlänge *l*, den Rohrdurchmesser *d*, Anzahl der Rohrkrümmer *n*, Verhältnis von Biegeradius zu Rohrdurchmesser und die Wandrauigkeit *k*. Maßgebend für den Druckverlust ist die Reynoldszahl *Re*, daher muss darüber hinaus noch die Viskosität *η* des Mediums vorgegeben werden.

Grundsätzlich werden bei der Berechnung der Rohrströmung drei Bereiche unterschieden:

- hydraulisch glatte Rohre (Re < 65 d/k)
- Übergangsbereich
- hydraulisch raue Rohre (Re > 1300 d/k)

Hinweis

Die Dichte wird pro Komponente nur einmal berechnet. Daher erfolgt die Berechnung über die gesamte Rohrstrecke mit konstanter Dichte. Soll eine Dichteänderung berücksichtigt werden, muss die Rohrleitung gestückelt und die Komponenten hintereinander verschaltet werden.

Bei Rohrkrümmern wird immer von einer 90°-Biegung ausgegangen. Ist das Verhältnis von Biegeradius zu Rohrdurchmesser RatioRD größer als 10, werden Rohrkrümmer wie ein gerades Rohr gehandhabt. Sie erhöhen somit die Länge der Rohrleitung Length.

Anhaltswerte für die Wandrauigkeit *k*

Werkstoff	Zustand der Rohre	*k* in mm
Asbest-Zement		0,05 – 0,1
Gusseisen	Neu	0,25 – 0,5
	Angerostet	1,0 – 1,5
	Verkrustet	1,5 – 5,0
Kupfer, Messing, Bronze, Aluminium, Glas, Kunststoff	Technisch glatt	0,001 – 0,0015
Stahl, genietet		0,5 – 10
Stahl, geschweißt	Neu	05 – 0,1
	Verrostet	0,4
	Leicht verkrustet	1,0 – 1,5
	Stark verkrustet	2,0 – 4,0
Stahl, gezogen oder gewalzt	Neu	0,02 – 0,1
	Leicht verkrustet	0,4
	Stark verkrustet	3,0
Stahl, verzinkt	Glatt	0,07
	Normal galvanisiert	0,15

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Viscosity	Viskosität des Mediums	mPas	0.891
Roughness	Wandrauigkeit	mm	0.05
Diameter	Durchmesser der Rohrleitung	mm	200
Length	Länge der Rohrleitung	m	1.0
NbrOfElbows	Anzahl der Rohrkrümmer	–	0
RatioRD	Verhältnis von Biegeradius zu Rohrdurchmesser	–	5.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckverlust	Δp	mbar

Im erweiterten Bedienfenster wird zusätzlich angezeigt:

Größe	Formelzeichen	Einheit
Reynoldszahl	Re	–

8.2.5.3 Graphics

Membrane

Symbol



Funktion

Die Komponente Membrane hat keine Funktion und dient nur der grafischen Darstellung.

Motor

Symbol



Funktion

Die Komponente Motor hat keine Funktion und dient nur der grafischen Darstellung.

MotorThreePhase

Symbol



Funktion

Die Komponente MotorThreePhase hat keine Funktion und dient nur der grafischen Darstellung.

8.2.5.4 Heatexchanger

Condenser – Kondensator

Symbol

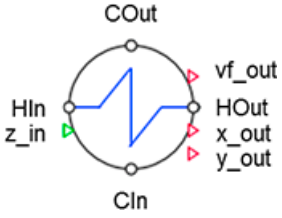


Bild 8-17 Symbol des Komponententyps Condenser

Funktion

Die Komponente *Condenser* ist ein Modell eines Wärmeübertragers, mit dem der auf der warmen Seite (*HIn*, *HOut*) angeschlossene Strom (teilweise) kondensiert werden kann. Die notwendige Wärmemenge zum Kondensieren wird dem Strom auf der kalten Seite (*CIn*, *COut*) entzogen. Die Zusammensetzung des warmen Stroms (Anteil des Leichtsieders) wird über den Eingang *z_in* vorgegeben. Der Dampfanteil (*vf* = VaporFraction) sowie die Zusammensetzung in der flüssigen (*x_out*) als auch in der Gas-Phase (*y_out*) werden berechnet und am Ausgang zur Verfügung gestellt.

Parametriert wird der Kondensator über die Wärmeaustauschfläche *A* und den Wärmeübergangskoeffizienten *k*. Das zu verdampfende Gemisch wird über die Siedetemperatur des Leicht- und Schwersieders (*TS_LB*, *TS_HB*) sowie über die Verdampfungsenthalpien (*hv_LB*, *hv_HB*) definiert. Die Größe der Siedelinse kann über den Parameter *S* variiert werden. Dabei bedeutet:

- $S = 0$, maximale Größe der Siedelinse (siehe Abbildung)
- $S = T_{HB} - T_{LB}$, keine Siedelinse (physikalisch unsinnig)

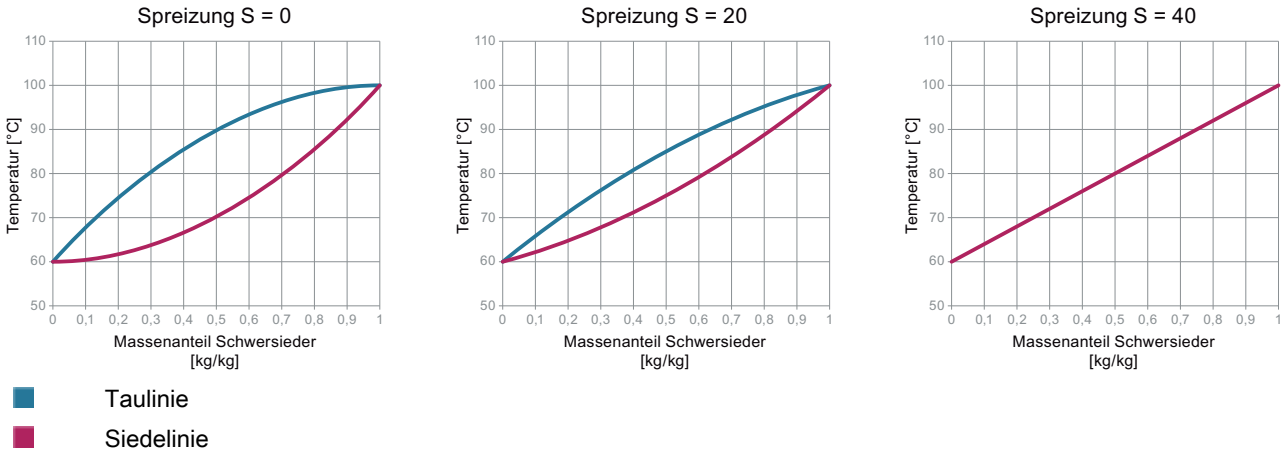


Bild 8-18 Siedelinse in Abhängigkeit des Parameters *S* (Spreizung)

Eine Druckabhängigkeit der Siedetemperaturen der Reinstoffe wird vernachlässigt.

ACHTUNG
Auf der warmen Seite ist die Stromrichtung fest vorgegeben. Ein Durchströmen ist immer nur vom Eingang (*HIn*) zum Ausgang (*HOut*) möglich. Eine Strömung aufgrund einer Druckdifferenz vom Ausgang zum Eingang wird über eine modellierte Rückschlagklappe verhindert. Auf der kalten Seite ist die Stromrichtung fest vorgegeben. Ein Durchströmen ist immer nur vom Eingang (*CIn*) zum Ausgang (*COut*) möglich. Eine Strömung aufgrund einer Druckdifferenz vom Ausgang zum Eingang wird über eine modellierte Rückschlagklappe verhindert. An dem Eingang auf der warmen Seite (*HIn*) muss ein Flussnetz mit dem Medium *Ideales Gas* angeschlossen sein. Am Ausgang auf der warmen Seite (*HOut*) muss ein Flussnetz mit dem Medium *Flüssig* angeschlossen sein.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Area	Wärmeübertragerfläche	m²	10
K	Wärmeübergangskoeffizient	W/(m²K)	800
cP_Wand	Wärmekapazität der Apparatewand (Stahl)	kJ/kg	0.5

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
T_HB	Siedetemperatur Schwersieder	°C	100
T_LB	Siedetemperatur Leichtsieder	°C	60
S	Spreizung Siedelinse	–	10
hv_HB	Verdampfungsenthalpie Schwersieder	kJ/kg/K	2300

Parametername	Beschreibung	Einheit	Standardwert
hc_LB	Verdampfungsenthalpie Leichtsieder	kJ/kg/K	1800
kvs_H	Durchflusskoeffizient warme Seite	m³/h	10
kvs_C	Durchflusskoeffizient kalte Seite	m³/h	10
kv0	Durchflusskoeffizient k_V	m³/h	0.000001
T1	Dämpfungszeit	s	1
T2	Initialisierungszeitintervalle	–	50
geoHeight	Geodätische Höhe	m	0
zH_Init	Zusammensetzung auf der heißen Seite zum Initialisieren	kg/kg	0.8
T_Init	Temperatur zum Initialisieren	°C	30

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Eingangstemperaturen	*T*	°C
Eingangskonzentration Leichtsieder	*z*	kg/kg
Ausgangstemperaturen	*T*	°C
Ausgangskonzentration Leichtsieder Flüssig	*x*	kg/kg
Ausgangskonzentration Leichtsieder Gas	*y*	kg/kg
Dampfanteil im Ausgang	*vf*	–

Siehe auch

Geodätische Höhe (Seite 612)

Evaporator – Verdampfer

Symbol

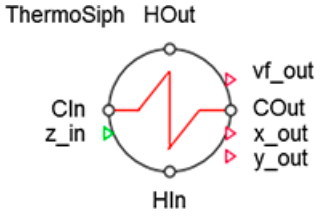


Bild 8-19 Symbol des Komponententyps Evaporator

Funktion

Die Komponente *Evaporator* ist ein Modell eines Wärmeübertragers, mit dem der auf der kalten Seite (*CIn*, *COut*) angeschlossene Strom (teilweise) verdampf werden kann. Die notwendige Wärme zum Verdampfen wird dem Strom auf der warmen Seite (*HIn*, *HOut*) entzogen. Die Zusammensetzung des kalten Stroms (Anteil des Leichtsieders) wird über den Eingang *z_in* vorgegeben. Der Dampfanteil (*vf* = VaporFraction) sowie die Zusammensetzung in der flüssigen (*x_out*) als auch in der Gas-Phase (*y_out*) werden berechnet und am Ausgang zur Verfügung gestellt.

Der Verdampfer kann als Verdampfer einer Kolonne mit Naturumlauf verwendet werden. Dazu muss der Anschluss *ThermoSyph* mit dem entsprechenden Anschluss der Kolonne verbunden werden. Über diesen Anschluss werden die notwendigen Informationen (Wärmestrom, Dampfanteil, Zusammensetzungen, k_{VS}-Wert der kalten Seite am Auslegungspunkt) für den Naturumlauf mit der Kolonne ausgetauscht. Für den Naturumlauf wird der aktuelle k_{VS}-Wert auf der kalten Seite des Verdampfers durch die Kolonne angepasst, so dass sich der Massenstrom des Naturumlaufes in Abhängigkeit der Dichteunterschiede einstellt.

Parametriert wird der Verdampfer über die Wärmeaustauschfläche und den Wärmeübergangskoeffizienten. Das zu verdampfende Gemisch wird über die Siedetemperatur des Leicht- und Schwersieders (*TS_LB*, *TS_HB*) sowie über deren Verdampfungsenthalpien (*hv_LB*, *hv_HB*) definiert. Die Größe der Siedelinse kann über den Parameter *S* variiert werden. Dabei bedeutet:

- $S = 0$, maximale Größe der Siedelinse (siehe Abbildung).
- $S = T_{HB} - T_{LB}$, keine Siedelinse (physikalisch unsinnig)

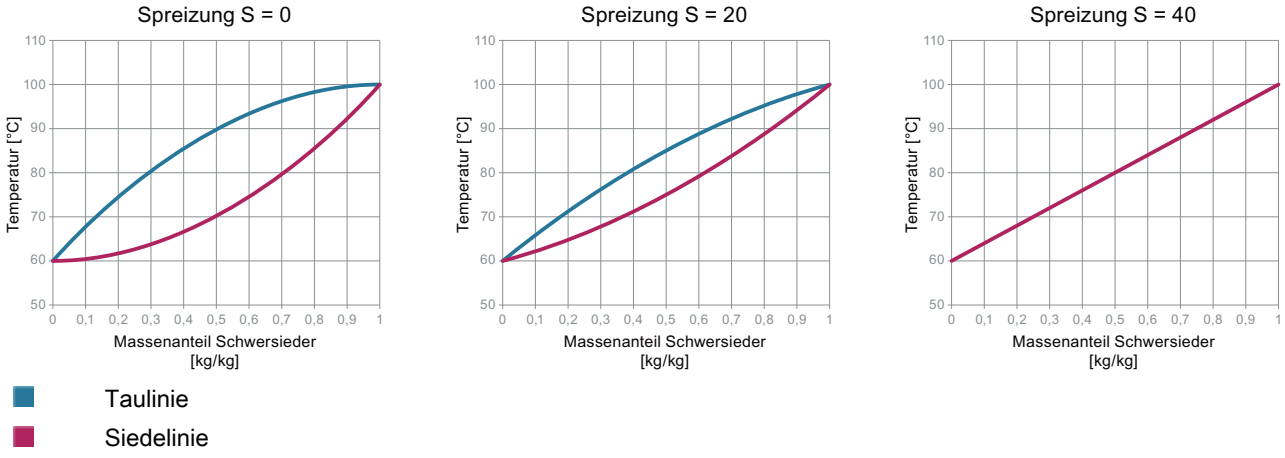


Bild 8-20 Siedelinse in Abhängigkeit des Parameters *S* (Spreizung)

Eine Druckabhängigkeit der Siedetemperaturen der Reinstoffe wird vernachlässigt.

ACHTUNG
Auf der kalten Seite ist die Stromrichtung fest vorgegeben. Ein Durchströmen ist immer nur vom Eingang (*CIn*) zum Ausgang (*COut*) möglich. Eine Strömung aufgrund einer Druckdifferenz vom Ausgang zum Eingang wird über eine modellierte Rückschlagklappe verhindert. Auf der warmen Seite ist die Stromrichtung fest vorgegeben. Ein Durchströmen ist immer nur vom Eingang (*HIn*) zum Ausgang (*HOut*) möglich. Eine Strömung aufgrund einer Druckdifferenz vom Ausgang zum Eingang wird über eine modellierte Rückschlagklappe verhindert. An Eingang auf der kalten Seite (*CIn*) muss ein Flussnetz mit dem Medium *Flüssig* angeschlossen sein. Am Ausgang auf der kalten Seite (*COut*) muss ein Flussnetz mit dem Medium *Ideales Gas* angeschlossen sein.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Area	Wärmeübertragerfläche	m²	10
K	Wärmeübergangskoeffizient	W/(m²K)	800
cP_Wand	Wärmekapazität der Apparatewand (Stahl)	kJ/kg	0.5

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
T_HB	Siedetemperatur Schwersieder	°C	100
T_LB	Siedetemperatur Leichtsieder	°C	60
S	Spreizung Siedelinse	–	10
hv_HB	Verdampfungsenthalpie Schwersieder	kJ/kg/K	2300
hc_LB	Verdampfungsenthalpie Leichtsieder	kJ/kg/K	1800
kvs_H	Durchflusskoeffizient warme Seite	m³/h	10
kvs_C	Durchflusskoeffizient kalte Seite	m³/h	10
kv0	Durchflusskoeffizient k_V	m³/h	0.000001
T1	Dämpfungszeit	s	1
T2	Initialisierungszeitintervalle	–	50
geoHeight	Geodätische Höhe	m	0
zH_Init	Zusammensetzung auf der heißen Seite zum Initialisieren	kg/kg	0.2
T_Init	Temperatur zum Initialisieren	°C	60

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Eingangstemperaturen	*T*	°C
Eingangskonzentration Leichtsieder	*z*	kg/kg
Ausgangstemperaturen	*T*	°C
Ausgangskonzentration Leichtsieder Flüssig	*x*	kg/kg
Ausgangskonzentration Leichtsieder Gas	*y*	kg/kg
Dampfanteil im Ausgang	*vf*	–

ElectricHeatExchanger – Elektrisch beheizter Wärmeübertrager

Symbol

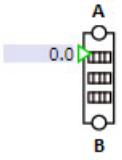


Bild 8-21 Symbol des Komponententyps ElectricHeatExchanger

Funktion

Der Komponententyp *ElectricHeatExchanger* dient zur Simulation einer elektrischen Heizung im Flussnetz. Über den Anschluss *P_EL* wird die elektrische Heizleistung P_{el} in kW vorgegeben.

Es wird angenommen, dass die zugeführte elektrische Energie P_{el} vollständig in Wärme umgewandelt wird. Es treten keine Verluste auf.

$$\frac{dh}{dt} = \frac{1}{\rho V}\,\dot{Q}$$

Die Komponente *ElectricHeatExchanger* wird über die Anschlüsse *A* und *B* in ein Flussnetz eingebaut. Der Komponententyp kann beidseitig durchströmt werden.

Die Größe des Wärmeübertragers kann über den Parameter *Volume* parametriert werden. Hierdurch ergibt sich auf der Ausgangsseite des Mediums bei Ein- und Ausschaltvorgängen der Heizleistung eine entsprechend verzögerte Temperaturänderung.

Dieses Volumen wird immer als vollständig gefüllt vorausgesetzt, d. h. es erfolgt keine Masseneinspeicherung bzw. Massenausspeicherung.

Der Druckverlust über der Komponente wird über den Parameter *Kvs* eingestellt.

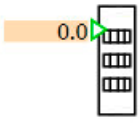
Zur Vermeidung von unerwünschten Zuständen der Simulation kann über den Parameter *MaxEnthalpy* die maximal erreichbare Enthalpie des ausfließenden Mediums festgelegt werden.

Hinweis

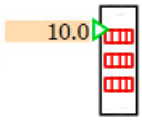
Durch Vorgabe einer negativen Heizleistung P_{el} am Eingang *P_EL* können Sie diese Komponente auch zur Kühlung verwenden.

Wenn sich der *ElectricHeatExchanger* im Heizmodus befindet, wird dies durch einen Farbumschlag im Symbol der Komponente *ElectricHeatExchanger* dargestellt.

ElectricHeatExchanger ohne Heizleistung



ElectricHeatExchanger mit Heizleistung



ACHTUNG

Der *ElectricHeatExchanger* berechnet nur Wärmeübergänge und keine damit verbundenen Dichteänderungen. Wenn der *ElectricHeatExchanger* in einem Wasser-Dampf-Flussnetz betrieben wird, werden diese Dichteänderungen erst im nächsten durchströmten Knoten berechnet.

Eine direkt nach dem *ElectricHeatExchanger* positionierte Messstelle im gleichen Zweig arbeitet daher noch mit der unveränderten Dichte. Die innerhalb dieser Messstelle berechnete Temperatur $T = f(p, h)$ ist somit nicht exakt.

Für eine exakte Berechnung ist zwischen *ElectricHeatExchanger* und Messstelle ein Knoten notwendig.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Speichervolumen des elektrischen Wärmeübertragers	m^3	0.1
Kvs	Kennwert zur Berechnung des Druckabfalls	m^3/h	360.0
InitialEnthalpy	Initialisierungsenthalpie des Mediums	kJ/kg	80.0
MaxEnthalpy	Maximale Enthalpie des Mediums	kJ/kg	3000.0
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Siehe auch

Geodätische Höhe (Seite 612)

HeatExchangerGeneral – Allgemeiner Wärmetauscher

Symbol

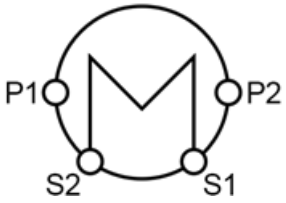


Bild 8-22 Symbol des Komponententyps HeatExchangerGeneral

Funktion

Die Komponente *HeatExchangerGeneral* ist ein Modell eines allgemeinen Wärmeübertragers. Wärme wird zwischen der Produktseite (PS) und der Serviceseite (SS) ausgetauscht. Parametriert wird der Wärmetauscher über die Wärmeaustauschfläche, den Wärmeübergangskoeffizienten und die minimale Temperaturdifferenz zwischen der Produkt- und Serviceseite. Die Berechnung der übertragenen Wärmemenge erfolgt über die mittlere logarithmische Temperaturdifferenz. Wird hierdurch die minimale Temperaturdifferenz unterschritten, erfolgt eine Korrektur des Wärmestroms, so dass die minimale Temperaturdifferenz eingehalten wird.

Die beiden Medien werden auf der Prozessseite über die Anschlüsse P1 und P2 verschaltet, auf der Serviceseite über die Anschlüsse S1 und S2. Je nach Stromführung (s. Tabelle) arbeitet der Wärmeübertrager im Gegenstrom oder im Gleichstrom.

	Prozessseite (PS)	
Serviceseite (SS)	1 → 2	2 → 1
1 → 2	Gegenstrom	Gleichstrom
2 → 1	Gleichstrom	Gegenstrom

ACHTUNG

Das Modell ist nicht geeignet, um bei Anschluss des Mediums Wasser/Dampf auf der Produkt- oder Serviceseite den Phasenwechsel zu berechnen. Hierbei kann es zu Abweichungen bei der Berechnung des Wärmestroms und damit bei den Austrittstemperaturen kommen. Somit ergeben sich drei Bereiche, in denen eine Komponente berechnet werden kann:

- Unterkühlte Flüssigkeit – siedende Flüssigkeit
- Siedende Flüssigkeit – gesättigter Dampf
- Gesättigter Dampf – überhitzter Dampf

Darstellung

Während der Simulation wird im Symbol der Komponente die Richtung des Wärmestroms farbig dargestellt. Dies ergibt sich aus dem Vorzeichen des Wärmestroms aus der Berechnung des Wärmetauschers.

		
Keine Wärmeübertragung	Wärmeübertragung PS → SS Kühlung Produktseite	Wärmeübertragung SS → PS Erwärmung Produktseite

Ist der berechnete Wärmestrom durch einen Phasenwechsel limitiert, färben sich die Linien grau:



Tritt ein Fehler bei der Berechnung des Wärmestroms auf, dann färben sich die Linien rot:



Falls das Symbol kurzfristig auftritt, z. B. während eines Lastwechsels, ist dies unproblematisch. Wenn es jedoch über einen längeren Zeitraum auftritt, so überprüfen Sie die eingestellten Parameter oder die entsprechenden Lastpunkte.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Area	Wärmeübertragerfläche	m²	10
K	Wärmeübergangskoeffizient	W/(m²K)	500

Zusatzparameter

Die Verdampfung/Kondensation macht häufig den Großteil des Wärmestroms aus. Bei der Auswahl von Wasser/Dampf als Medium kann über die Optionen *PhaseChangePS* (für die Prozessseite) und *PhaseChangeSS* (für die Serviceseite) die Berechnung der Kondensation/ Verdampfung erzwungen werden. Hierdurch wird eine Berechnung im Nassdampfbereich (Bereich 2) durchgeführt. Mögliche Wärmemengen aus überhitztem Dampf oder unterkühlter Flüssigkeit werden hier nicht berücksichtigt.

Voraussetzung für die Berechnung ist, dass die nötigen physikalischen Gegebenheiten durch die Eingangstemperaturen am Wärmeübertrager erfüllt sind. Wird die Option nicht gewählt,

wird maximal bis zum Erreichen des Nassdampfbereichs, also gesättigter Dampf bzw. gesättigte Flüssigkeit, gerechnet. Es kommt zu keinem Phasenwechsel.

Parametername	Beschreibung	Einheit	Standardwert
DTmin	Minimale Temperaturdifferenz	°C	10
KvsPS	Prozessseitiger Durchflusskoeffizient k_{VS}	m³/h	100
KvsSS	Serviceseitiger Durchflusskoeffizient k_{VS}	m³/h	100
PhaseChangePS	Erzwungene Berechnung im Nass-dampfbereich für die Prozessseite (nur relevant bei Wasser/Dampf als Medium)	–	False
PhaseChangeSS	Erzwungene Berechnung im Nass-dampfbereich für die Serviceseite (nur relevant bei Wasser/Dampf als Medium)	–	False
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Eingangstemperaturen	T_in	°C
Ausgangstemperaturen	T_out	°C
Wärmestrom (von Service- zu Prozessseite)	Q	kW
Gegenstrombetrieb	Countercurrent	–

Siehe auch

Geodätische Höhe (Seite 612)

HeatExchangerShellTube – Rohrbündelwärmeübertrager

Symbol

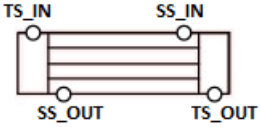


Bild 8-23 Symbol des Komponententyps HeatExchangerShellTube

Funktion

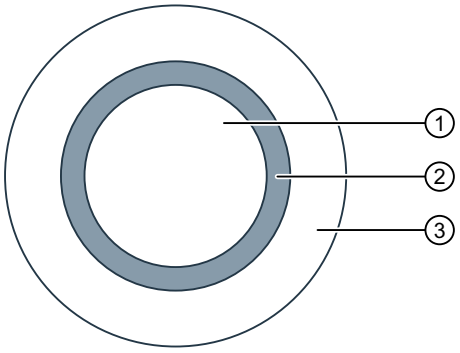
Bei der Komponente *HeatExchangerShellTube* handelt es sich um das Modell eines eingängigen Rohrbündelwärmeübertragers im Gegenstrom. Daher wird sie über entsprechende geometrische Daten parametriert.

Die beiden Medien werden rohrseitig über die Anschlüsse *TS_IN* und *TS_OUT* geführt, mantelseitig über die Anschlüsse *SS_IN* und *SS_OUT*.

Wenn die Strömungsrichtung auf einer Seite umgedreht wird, arbeitet der Wärmeübertrager entsprechend im Gleichstrom.

Aus den parametrierten geometrischen Daten werden je ein Gesamtvolumen für die Rohrinnenseite und für die Mantelseite berechnet. Die Übertragerfläche ergibt sich aus der Summe der Übertragerflächen der Einzelrohre.

Das sich somit ergebene Einrohrmodell ist in der folgenden Abbildung dargestellt.



① Rohrseitiger Raum
② Rohr
③ Mantelseitiger Raum

Bild 8-24 Aufbau des Einrohrmodells im Komponententyp HeatExchangerShellTube

Zur vereinfachten Berechnung wird dieses Modell segmentiert. Die Anzahl der Segmente wird über den Parameter *NbrOfSegments* vorgegeben. Innerhalb eines Segments wird von einer idealen Durchmischung des jeweiligen Mediums ausgegangen. Aus diesem Ansatz ergibt sich somit ein Diskretisierungsfehler.

Die Anzahl der Segmente kann zwischen 2 und 20 gewählt werden.

Die Wärmeübergänge zwischen Rohrinnenraum, Rohrmaterial und Mantelraum können über die Parameter *HTC_TS* (Rohrraum → Rohr) bzw. *HTC_SS* (Rohr → Mantelraum) eingestellt werden.

Wärmeverluste über den Mantel an die Umgebung können über den Parameter *HTC_Shell* parametriert werden. Der Wärmeverlust wird in Abhängigkeit von der Umgebungstemperatur (Eingang *AmbientTemperature*) berechnet.

Der Zustand "stable condition" im Bedienfenster kann in diesem Fall nicht mehr erreicht werden, da im Wärmestrom Q (SS-Tube) auch die Wärmeverluste durch die Wand enthalten sind.

Hinweis

Wählen Sie die Segmentanzahl optimal.

Eine höhere Segmentzahl führt zur Reduzierung des Diskretisierungsfehlers.

Eine zu hohe Segmentzahl kann aber auch zur Erhöhung der Rechenlast und zu Instabilitäten führen.

Im instabilen Zustand zeigt der Wärmeübertrager folgendes Symbol an:



Bild 8-25 Instabile Zustände im HeatExchangerShellTube, Autokorrektur aktiv

Falls das Symbol kurzzeitig auftritt, z. B. während eines Lastwechsels, ist dies unproblematisch. Wenn es jedoch über einen längeren Zeitraum auftritt, so überprüfen Sie die eingestellten Parameter oder die entsprechenden Lastpunkte. Zu größerer Stabilität führen z. B. eine Reduzierung der Segmentzahl oder eine Reduzierung des Massenstroms.

Die Art des Mediums auf der Mantel- oder Rohrseite ergibt sich aus der Verschaltung des Wärmeübertragers im Flussnetz und muss nicht gesondert parametriert werden.

Die Wärmekapazität des Metalls (Rohre und Mantel) ist mit 0,46 kJ/(kg·K) fest vorgegeben.

ACHTUNG
Der *HeatExchangerShellTube* berechnet nur Wärmeübergänge und keine damit verbundenen Dichteänderungen. Wenn der *HeatExchangerShellTube* in einem Wasser-Dampf-Flussnetz betrieben wird, werden diese Dichteänderungen erst im nächsten durchströmten Knoten berechnet. Eine direkt nach dem *HeatExchangerShellTube* positionierte Messstelle im gleichen Zweig arbeitet daher noch mit der unveränderten Dichte. Die innerhalb dieser Messstelle berechnete Temperatur $T = f(p, h)$ ist somit nicht exakt. Für eine exakte Berechnung ist zwischen *HeatExchangerShellTube* und Messstelle ein Knoten notwendig.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
NbrOfSegments	Anzahl der Segmente des Wärmeübertragers	–	6

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
D_Tube	Rohraußendurchmesser der Einzelrohre	m	0.016
L_Tube	Länge der Rohrbündel	m	2.0
N_Tube	Anzahl der Rohre	–	150
S_Tube	Wandstärke der Rohre	m	0.0022
Rho_Tube	Dichte der Rohre	kg/m³	7800.0
KvsSS	Mantelseitiger Durchflusskoeffizient k_{VS}	m³/h	100.0
KvsTS	Rohrseitiger Durchflusskoeffizient k_{VS}	m³/h	100.0
HTC_TS	Wärmeübergangskoeffizient an der Rohrinnenseite	–	1.0

Parametername	Beschreibung	Einheit	Standardwert
HTC_SS	Wärmeübergangskoeffizient an der Rohraußenseite	–	1.0
HTC_Shell	Wärmeübergangskoeffizient an der Mantelseite	–	0.0
Shell_Volume_Factor	Faktor zur Berechnung des Mantelvolumens in Abhängigkeit vom Rohrvolumen V_{Shell} = *Shell_Volume_Factor* · V_{Tube}	–	3.0
InitialEnthTS	Initialisierungsenthalpie des Mediums im Rohrraum	kJ/kg	120.0
InitialEnthSS	Initialisierungsenthalpie des Mediums im Mantelraum	kJ/kg	30.0
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Beschreibung	Einheit
Massflow SS	Massenstrom auf der Mantelseite des Wärmeübertragers	kg/s
DeltaT SS	Temperaturänderung des Mediums zwischen Eintritt und Austritt auf der Mantelseite	K
Massflow TS	Massenstrom auf der Rohrseite des Wärmeübertragers	kg/s
DeltaT TS	Temperaturänderung des Mediums zwischen Eintritt und Austritt auf der Rohrseite	K
Q (Tube->TS)	Wärmestrom vom Rohr auf das Medium im Rohrraum	kW
Q (SS->Tube)	Wärmestrom vom Medium im Mantelraum auf das Rohr und den Mantel	kW

Ein gefülltes Kästchen ("stable condition") zeigt an, dass eine konstante Durchströmung mit stabilem Wärmeübergang erreicht ist.

Erweitertes Bedienfenster

Im erweiterten Bedienfenster werden aktuell berechnete Werte (Temperaturen im Mantelraum, im Rohrraum und der Rohre) der einzelnen Segmente dargestellt. Zusätzlich werden die aus den parametrierten geometrischen Daten intern berechneten Kenngrößen des Wärmeübertragers dargestellt.

Im oberen Teil werden die Segmente mit den entsprechenden Temperaturen im Rohrraum (links), Mantelraum (rechts) und Rohrmaterial (Mitte) angezeigt. Nicht genutzte Segmente werden als inaktiv dargestellt. Im Beispiel werden 6 Segmente genutzt, maximal können Sie 20 Segmente nutzen.

Im unteren Teil des erweiterten Bedienfensters werden die aus den parametrierten geometrischen Daten berechneten Kenngrößen angezeigt.

Siehe auch

Geodätische Höhe (Seite 612)

HeatPipe – elektrisch beheiztes Rohr

Symbol



Bild 8-26 Symbol des Komponententyps HeatPipe

Funktion

Der Komponententyp *HeatPipe* dient zur Simulation einer elektrisch beheizten Rohrleitung. Über den Anschluss *P_EL* wird die elektrische Heizleistung P_{el} in kW vorgegeben.

Dabei gilt die Annahme, dass die zugeführte elektrische Energie P_EL vollständig in Wärme umgewandelt wird. Es treten keine Verluste auf.

$$\frac{dh}{dt} = \frac{1}{\rho V}\,\dot{Q}$$

Die Komponente *HeatPipe* wird über die Anschlüsse *A* und *B* in ein Flussnetz eingebaut. Der Komponententyp kann beidseitig durchströmt werden.

Die Größe des beheizten Rohrs können Sie über den Parameter *Volume* parametrieren. Hierdurch ergibt sich auf der Ausgangsseite des Mediums bei Ein- und Ausschaltvorgängen der Heizleistung eine entsprechend verzögerte Temperaturänderung. Hierbei wird das Rohrvolumen jedoch als ideal durchmischt angesehen, so dass die Temperatur im gesamten Rohrvolumen gleich ist und entsprechend der Temperaturänderung langsam erhöht oder abgesenkt wird.

Dieses Volumen wird immer als vollständig gefüllt vorausgesetzt, d. h. es erfolgt keine Masseneinspeicherung bzw. Massenausspeicherung.

Der Druckverlust über der Komponente wird über den Parameter *Kvs* eingestellt.

Zur Vermeidung von unerwünschten Zuständen der Simulation können Sie über den Parameter *MaxEnthalpy* die maximal erreichbare Enthalpie des ausfließenden Mediums festlegen.

Hinweis

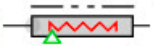
Durch Vorgabe einer negativen Heizleistung P_{el} am Eingang *P_EL* können Sie diese Komponente auch zur Kühlung verwenden.

Wenn sich das Heizrohr im Heizmodus befindet, wird dies durch Darstellung einer roten Heizwendel im Symbol der Komponente *HeatPipe* dargestellt.

HeatPipe ohne Heizleistung



HeatPipe mit Heizleistung



ACHTUNG
Die Komponente *HeatPipe* berechnet nur Wärmeübergänge und keine damit verbundenen Dichteänderungen. Wenn die *HeatPipe* in einem Wasser-Dampf-Flussnetz betrieben wird, werden diese Dichteänderungen erst im nächsten durchströmten Knoten berechnet. Eine direkt nach der *HeatPipe* positionierte Messstelle im gleichen Zweig arbeitet daher noch mit der unveränderten Dichte. Die innerhalb dieser Messstelle berechnete Temperatur $T = f(p, h)$ ist somit nicht exakt. Für eine exakte Berechnung ist zwischen *HeatPipe* und Messstelle ein Knoten notwendig.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Speichervolumen des Heizrohrs	m^3	0.1
Kvs	Durchflusskoeffizient zur Berechnung des Druckabfalls	m^3/h	360.0
InitialEnthalpy	Initialisierungsenthalpie des Mediums	kJ/kg	80.0
MaxEnthalpy	Maximale Enthalpie des Mediums	kJ/kg	3000.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
geoHeight	Geodätische Höhe; online änderbar	m	0.0

8.2.5.5 Measurements

DifferencePressureIndicator – Differenzdruckanzeige

Symbol

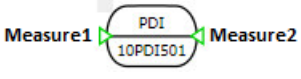


Bild 8-27 Symbol des Komponententyps DifferencePressureIndicator

Funktion

Der Komponententyp *DifferencePressureIndicator* dient zur Anzeige der Differenz der über seine Eingänge *Measure1* und *Measure2* vorgegebenen Drücke.

Über den Parameter *SwitchValue_Plus* oder *SwitchValue_Minus* kann eine Ober- bzw. Untergrenze für den Messwert angegeben werden. Mit den Schaltern *Use_Plus* und *Use_Minus* kann die Überwachung der Grenzwerte aktiviert werden. Das Erreichen der Ober- bzw. Untergrenze wird im Symbol durch eine grüne Binäranzeige angezeigt.

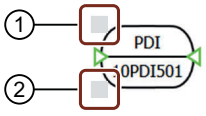
Über den Parameter *UoM* wird die Einheit des analogen Ausgangs sowie der Schaltgrenzen festgelegt.

Wenn mehrere Schaltwerte benötigt werden, so müssen Sie auch mehrere Messstellen verwenden. Alternativ können Sie zusätzliche Binärwerte auch über Logik projektieren.

Zur weiteren Verkopplung der Messwerte können Sie folgende Ausgänge der Komponente sichtbar schalten:

- *AOUT*
  Verschaltung des Analogwerts mit der Einheit, die über Parameter *UoM* eingestellt ist
- *BOUT_PLUS*
  Verschaltung des Binärwerts der Obergrenze
- *BOUT_MINUS*
  Verschaltung des Binärwerts der Untergrenze

Der Zustand dieser Schaltwerte wird im Grafiksymbol dynamisch dargestellt:



① Darstellung des Zustandes des Schaltwerts der Obergrenze

② Darstellung des Zustandes des Schaltwerts der Untergrenze

Bild 8-28 Schaltwerte im Grafiksymbol des Komponententyps DifferencePressureIndicator

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code, Art der Messung	–	PDI
Use_Plus	Aktivierung der Obergrenze	–	False
SwitchValue_Plus	Schaltwert für Obergrenze	s. UoM	0.0
Use_Minus	Aktivierung der Untergrenze	–	False
SwitchValue_Minus	Schaltwert für Untergrenze	s. UoM	0.0
UoM	Gibt den Messwert in der gewählten Einheit aus. Schaltwerte beziehen sich ebenfalls auf die gewählte Einheit. Zur Verfügung stehen folgende Einheiten: atm, bar, inHg, inWC, kPa, mbar, mmHg, MPa, mWS, Pa, psi, Torr		bar

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Druckdifferenz der Messeingänge	Δp	bar mbar kPa Pa psi

DriveIndicator – Motoransteuerung

Symbol



Bild 8-29 Symbol des Komponententyps DriveIndicator

Funktion

Der Komponententyp *DriveIndicator* hat ausschließlich grafische Funktionen. Er dient zur Verdeutlichung des Signalwegs zur Ansteuerung von Antrieben.

Bei Bedarf können Sie diese Komponente auch direkt zur Ansteuerung von Antrieben nutzen. Hierzu verbinden Sie die Komponente über den Eingang *DI* mit der Aktor-Sensor-Ebene. Der Ausgang *DO* wird dann beispielsweise auf den Stelleingang des Ventils oder einer Pumpe verschaltet.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code	–	YS

FlowIndicator – Durchflussanzeige

Symbol



Bild 8-30 Symbol des Komponententyps FlowIndicator

Funktion

Der Komponententyp *FlowIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Durchflusses.

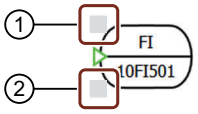
Über den Parameter *SwitchValue_Plus* oder *SwitchValue_Minus* kann eine Ober- bzw. Untergrenze für den Messwert angegeben werden. Mit den Schaltern *Use_Plus* und *Use_Minus* kann die Überwachung der Grenzwerte aktiviert werden. Das Erreichen des entsprechenden Schaltwerts wird im Symbol durch eine grüne Binäranzeige angezeigt.

Über den Parameter *UoM* wird die Einheit des analogen Ausgangs sowie der Schaltgrenzen festgelegt.

Zur weiteren Verkopplung der Messwerte können Sie folgende Ausgänge der Komponente sichtbar schalten:

- *AOUT*
  Verschaltung des Analogwerts mit der Einheit, die über Parameter *UoM* eingestellt ist
- *BOUT_Plus*
  Verschaltung des Binärwerts der Obergrenze
- *BOUT_Minus*
  Verschaltung des Binärwerts der Untergrenze

Der Zustand dieser Schaltwerte wird im Grafiksymbol dynamisch dargestellt:



① Darstellung des Zustandes des Schaltwerts der Obergrenze

② Darstellung des Zustandes des Schaltwerts der Untergrenze

Bild 8-31 Schaltwerte im Grafiksymbol des Komponententyps FlowIndicator

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code, Art der Messung	–	FI
Use_Plus	Aktivierung der Obergrenze	s. UoM	False
SwitchValue_Plus	Schaltwert für Obergrenze	–	0.0
Use_Minus	Aktivierung der Untergrenze	s. UoM	False
SwitchValue_Minus	Schaltwert für Untergrenze	–	0.0
UoM	Gibt den Messwert in der gewählten Einheit aus. Schaltwerte beziehen sich ebenfalls auf die gewählte Einheit. Zur Verfügung stehen folgende Einheiten: g, kg, l, lb, m³, ml, t jeweils pro s, min, h, d	–	kg/s

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s kg/h t/h
Volumenstrom	$\dot{V}$	m^3/h l/h

LevelIndicator – Füllstandsanzeige

Symbol

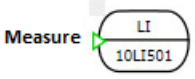


Bild 8-32 Symbol des Komponententyps LevelIndicator

Funktion

Der Komponententyp *LevelIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Füllstandes.

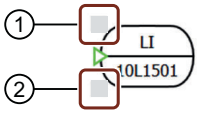
Über den Parameter *SwitchValue_Plus* oder *SwitchValue_Minus* kann eine Ober- bzw. Untergrenze für den Messwert angegeben werden. Mit den Schaltern *Use_Plus* und *Use_Minus* kann die Überwachung der Grenzwerte aktiviert werden. Das Erreichen der Ober- bzw. Untergrenze des entsprechenden Schaltwerts wird im Symbol durch eine grüne Binäranzeige angezeigt.

Über den Parameter *UoM* wird die Einheit des analogen Ausgangs sowie der Schaltgrenzen festgelegt.

Zur weiteren Verkopplung der Messwerte können Sie folgende Ausgänge der Komponente sichtbar schalten:

- *AOUT*
  Verschaltung des Analogwerts
- *BOUT_Plus*
  Verschaltung des Binärwerts der Obergrenze
- *BOUTMinus*
  Verschaltung des Binärwerts der Untergrenze

Der Zustand dieser Schaltwerte wird im Grafiksymbol dynamisch dargestellt:



① Darstellung des Zustandes des Schaltwerts der Obergrenze
② Darstellung des Zustandes des Schaltwerts der Untergrenze

Bild 8-33 Schaltwerte im Grafiksymbol des Komponententyps LevelIndicator

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code, Art der Messung	–	LI
Use_Plus	Aktivierung der Obergrenze	s. UoM	False
SwitchValue_Plus	Schaltwert für Obergrenze	–	0.0
Use_Minus	Aktivierung der Untergrenze	s. UoM	False
SwitchValue_Minus	Schaltwert für Untergrenze	–	0.0
UoM	Gibt den Messwert in der gewählten Einheit aus. Schaltwerte beziehen sich ebenfalls auf die gewählte Einheit. Zur Verfügung stehen folgende Einheiten: ft, in, m, mm, %		m

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Füllstand	l	m %

NozzleMeasurement – Stutzenmessstelle

Symbol



Bild 8-34 Symbol des Komponententyps NozzleMeasurement

Funktion

Die Komponente *NozzleMeasurement* dient zum Anschluss von Messstellen (z. B. *LevelIndicator* oder *TemperatureIndicator*) an eine andere Komponente (z. B. *StorageTankLiquid* oder *PressureStrainer*).

Die Komponente reicht die am Eingang *A* anstehenden Werte an den Ausgang *B* weiter. Die Werte bleiben unverändert.

PipeMeasure – Rohrmessstelle

Symbol



① Measure

Bild 8-35 Symbol des Komponententyps PipeMeasure

Funktion

Der Komponententyp *PipeMeasure* bildet eine Messstelle im Rohr. Sie fügen ihn mit seinen Anschlüssen *A* und *B* an der gewünschten Messstelle im Flussnetz ein.

Über den Anschluss *Measure* werden folgende Messgrößen ausgegeben:

- Absolutwert $|\dot{m}|$ des Durchflusses
- Druck p_A am Anschluss *A*
- Temperatur *T*

Alle anderen Signale des Anschlusses *Measure* werden in dieser Komponente nicht gesetzt.

Hinweis

In dieser Komponente werden die Messverfahren für die verschiedenen Größen nicht mit geeigneten Modellen simuliert. Es werden lediglich die im Flussnetzlöser berechneten Größen ausgegeben.

Die Richtung des Medienflusses wird bei laufender Simulation durch einen Pfeil im Symbol angezeigt (siehe Abbildung).



Bild 8-36 Medienflussanzeige des Komponententyps PipeMeasure

Parameter

Parametername	Beschreibung	Einheit	Standardwert
geoHeight	Einbauhöhe der Messstelle	m	0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Massenstrom	$\dot{m}$	kg/s
Temperatur	T	°C

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Enthalpie	h	kJ/kg
Dichte	r	kg/m³

Siehe auch

Geodätische Höhe (Seite 612)

PressureIndicator – Druckanzeige

Symbol



Bild 8-37 Symbol des Komponententyps PressureIndicator

Funktion

Der Komponententyp *PressureIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Druckwerts.

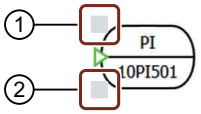
Über den Parameter *SwitchValue_Plus* oder *SwitchValue_Minus* kann eine Ober- bzw. Untergrenze für den Messwert angegeben werden. Mit den Schaltern *Use_Plus* und *Use_Minus* kann die Überwachung der Grenzwerte aktiviert werden. Das Erreichen der Ober- bzw. Untergrenze des entsprechenden Schaltwerts wird im Symbol durch eine grüne Binäranzeige angezeigt.

Über den Parameter *UoM* wird die Einheit des analogen Ausgangs sowie der Schaltgrenzen festgelegt.

Zur weiteren Verkopplung der Messwerte können Sie folgende Ausgänge der Komponente sichtbar schalten:

- *AOUT*
  Verschaltung des Analogwerts mit der Einheit, die über Parameter *UoM* eingestellt ist
- *BOUT_Plus*
  Verschaltung des Binärwerts der Obergrenze
- *BOUT_Minus*
  Verschaltung des Binärwerts der Untergrenze

Der Zustand dieser Schaltwerte wird im Grafiksymbol dynamisch dargestellt:



① Darstellung des Zustandes des Schaltwerts der Obergrenze
② Darstellung des Zustandes des Schaltwerts der Untergrenze

Bild 8-38 Schaltwerte im Grafiksymbol des Komponententyps PressureIndicator

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code, Art der Messung	–	PI
Use_Plus	Aktivierung der Obergrenze	s. UoM	False
SwitchValue_Plus	Schaltwert für Obergrenze	–	0.0
Use_Minus	Aktivierung der Untergrenze	s. UoM	False
SwitchValue_Minus	Schaltwert für Untergrenze	–	0.0
UoM	Gibt den Messwert in der gewählten Einheit aus. Schaltwerte beziehen sich ebenfalls auf die gewählte Einheit. Zur Verfügung stehen folgende Einheiten: atm, bar, inHg, inWC, kPa, mbar, mmHg, MPa, mWS, Pa, psi, Torr		bar

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar mbar kPa Pa psi

TemperatureIndicator – Temperaturanzeige

Symbol



Bild 8-39 Symbol des Komponententyps TemperatureIndicator

Funktion

Der Komponententyp *TemperatureIndicator* dient zur Anzeige der über seinen Eingang *Measure* vorgegebenen Temperatur.

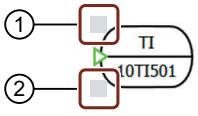
Über den Parameter *SwitchValue_Plus* oder *SwitchValue_Minus* kann eine Ober- bzw. Untergrenze für den Messwert angegeben werden. Mit den Schaltern *Use_Plus* und *Use_Minus* kann die Überwachung der Grenzwerte aktiviert werden. Das Erreichen der Ober- bzw. Untergrenze des entsprechenden Schaltwerts wird im Symbol durch eine grüne Binäranzeige angezeigt.

Über den Parameter *UoM* wird die Einheit des analogen Ausgangs sowie der Schaltgrenzen festgelegt.

Zur weiteren Verkopplung der Messwerte können Sie folgende Ausgänge der Komponente sichtbar schalten:

- *AOUT*
  Verschaltung des Analogwerts mit der Einheit, die über Parameter *UoM* eingestellt ist
- *BOUT_Plus*
  Verschaltung des Binärwerts der Obergrenze
- *BOUT_Minus*
  Verschaltung des Binärwerts der Untergrenze

Der Zustand dieser Schaltwerte wird im Grafiksymbol dynamisch dargestellt:



① Darstellung des Zustandes des Schaltwerts
② Darstellung des Zustandes des Schaltwerts

Bild 8-40 Schaltwerte im Grafiksymbol des Komponententyps TemperatureIndicator

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code, Art der Messung	–	TI
Use_Plus	Aktivierung der Obergrenze	–	False
SwitchValue_Plus	Schaltwert für Obergrenze	–	0.0
Use_Minus	Aktivierung der Untergrenze	–	False

Parametername	Beschreibung	Einheit	Standardwert
SwitchValue_Minus	Schaltwert für Untergrenze	–	0.0
UoM	Gibt den Messwert in der gewählten Einheit aus. Schaltwerte beziehen sich ebenfalls auf die gewählte Einheit. Zur Verfügung stehen folgende Einheiten: °C, °F, K		°C

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Temperatur	T	°C °F K

WeightIndicator – Gewichtsanzeige

Symbol



Bild 8-41 Symbol des Komponententyps WeightIndicator

Funktion

Der Komponententyp *WeightIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Gewichts.

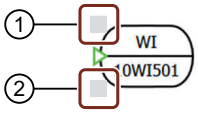
Über den Parameter *SwitchValue_Plus* oder *SwitchValue_Minus* kann eine Ober- bzw. Untergrenze für den Messwert angegeben werden. Mit den Schaltern *Use_Plus* und *Use_Minus* kann die Überwachung der Grenzwerte aktiviert werden. Das Erreichen der Ober- bzw. Untergrenze des entsprechenden Schaltwerts wird im Symbol durch eine grüne Binäranzeige angezeigt.

Über den Parameter *UoM* wird die Einheit des analogen Ausgangs sowie der Schaltgrenzen festgelegt.

Zur weiteren Verkopplung der Messwerte können Sie folgende Ausgänge der Komponente sichtbar schalten:

- *AOUT*
  Verschaltung des Analogwerts mit der Einheit, die über Parameter *UoM* eingestellt ist
- *BOUT_Plus*
  Verschaltung des Binärwerts der Obergrenze
- *BOUT_Minus*
  Verschaltung des Binärwerts der Untergrenze

Der Zustand dieser Schaltwerte wird im Grafiksymbol dynamisch dargestellt:



① Darstellung des Zustandes des Schaltwerts der Obergrenze
② Darstellung des Zustandes des Schaltwerts der Untergrenze

Bild 8-42 Schaltwerte im Grafiksymbol des Komponententyps WeightIndicator

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ICA_Code	MSR-Code, Art der Messung	–	TI
Use_Plus	Aktivierung der Obergrenze	–	False
SwitchValue_Plus	Schaltwert für Obergrenze	–	0.0
UseMinus	Aktivierung der Untergrenze	–	False
SwitchValue_Minus	Schaltwert für Untergrenze	–	0.0
UoM	Gibt den Messwert in der gewählten Einheit aus. Schaltwerte beziehen sich ebenfalls auf die gewählte Einheit. Zur Verfügung stehen folgende Einheiten: g, kg, lb, t	–	kg

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Gewicht	m	kg

8.2.5.6 Mixing Apparatuses

Mixer – Mischer

Symbol

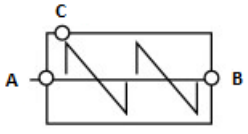


Bild 8-43 Symbol des Komponententyps Mixer

Funktion

Der Mixer dient der Zumischung eines Teilmassenstroms *C* (z. B. eines Feststoffmassenanteils oder eines Flüssigmassenanteils) in einen Hauptstrom *AB*.

Im Mixer wird der Massenanteil des zugeführten Teilmassenstroms im Hauptstrom berechnet. Dies kann z. B. in einem Umwälzkreislauf sinnvoll sein, wenn ein zweiter Stoff zugeführt wird (z. B. Aufkonzentration einer Säure).

Das Flussnetz des zuführenden Teilmassenstroms *C* ist topologisch nicht mit dem Flussnetz des Hauptmassenstroms *AB* verbunden. Die Veränderung der Stoffeigenschaften (z. B. Wärmekapazität) des austretenden Hauptstroms durch die Zumischung des Teilstroms *C* wird vernachlässigt.

Im Fall der Zumischung eines Feststoffs wird dieser wie eine Flüssigkeit behandelt. Für einen Stofftransport ist somit ein Druckunterschied notwendig. Dies ergibt sich daraus, dass in SIMIT das Flussnetzwerk prinzipiell nur den Transport von Gasen oder Flüssigkeiten zulässt. Eventuelle Fördereinrichtungen werden durch Ersatzverschaltungen (bestehend aus Pumpen, Ventilen, Quellen und Senken) ersetzt.

Der Hauptmassenstrom kann in beide Richtungen strömen. Der zugeführte Teilmassenstrom führt immer in den Mixer hinein. Voraussetzung hierfür ist ein entsprechend höherer Druck im Flussnetz des Teilmassenstroms als im Hauptmassenstrom.

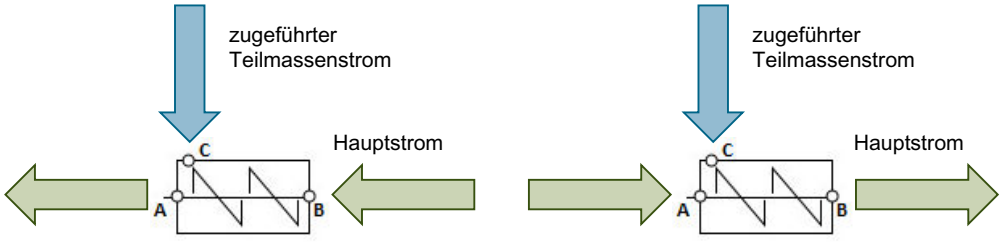


Bild 8-44 Strömungsmöglichkeiten im Komponententyp Mixer

Die Hauptstromanschlüsse *A* und *B* sind gleichwertig ausgelegt. Der zuzumischende Teilmassenstrom wird immer über den Anschluss *C* zugeführt.

Mit der Komponente *Mixer* ist es möglich, für jeden einzelnen Stoffstromanschluss einen Massenanteil für die Teilmassen an den Eingängen *RateA*, *RateB* oder *RateC* vorzugeben.

Wenn keiner der drei Anschlüsse vorgegeben (Voreinstellung) ist, gilt Folgendes:

- 0 % Massenanteil im Hauptstromeintritt
- 0 % Massenanteil im Teilmassenstromeintritt

Der Massenanteil des an *A* oder *B* austretenden Gesamtstroms wird berechnet und am Ausgang *Rate* angezeigt.

Für den Hauptmassenstrom wird ein Druckabfall berechnet. Dieser Druckabfall ist abhängig von:

- Durchflusskoeffizient des Mixers (k_{VS}-Wert)
- Durchströmung (Volumenstrom, Medium)

Der Druckabfall über dem Mixer wird gemäß folgender Formel berechnet:

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Dabei ist

$\Delta p = p_B - p_A$	der Druckabfall über dem Mixer in bar
$\dot{m}$	der Durchfluss bzw. Massenstrom in kg/s
ρ	die Dichte des Mediums in kg/m³
k_{VS}	der Durchflusskoeffizient des Mixers in m³/h

ACHTUNG
Das Flussnetz des zuführenden Massenstroms am Anschluss *C* wird in der Komponente mit einer Massenstromquelle abgeschlossen. Aufgrund dieser Massenstromquelle ist der sich am Rand des Flussnetzes einstellende Druck abhängig vom zugeführten Massenstrom und dem Druckabfall im zuführenden Zweig. Zu hohe Druckabfälle im zuführenden Zweig (d. h. beispielsweise zu geringe k_{VS}-Werte im Hauptstrom) können zu physikalisch nicht möglichen Zuständen führen.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Durchflusskoeffizient des Mixers k_{VS} mit $k_{VS} \geq 10^{-6}$ m³/h	m³/h	10.0
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen für folgende Massenströme angezeigt:

- Eintretender Hauptmassenstrom *A* oder *B*
- Zugemischter Massenstrom *C*
- Austretender Gesamtmassenstrom *A* oder *B*

Größe	Formelzeichen	Einheit
Massenanteil	w	%
Massenstrom	$\dot{m}$	kg/s

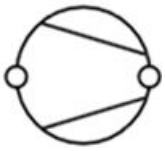
Siehe auch

Geodätische Höhe (Seite 612)

8.2.5.7 Pump

Compressor – Verdichter

Symbol



Symbol des Komponententyps Compressor

Funktion

Der Komponententyp *Compressor* berechnet die Druckerhöhung eines Gasstroms durch einen Kompressor in Abhängigkeit vom Förderstrom und der Drehzahl gemäß folgender Formel:

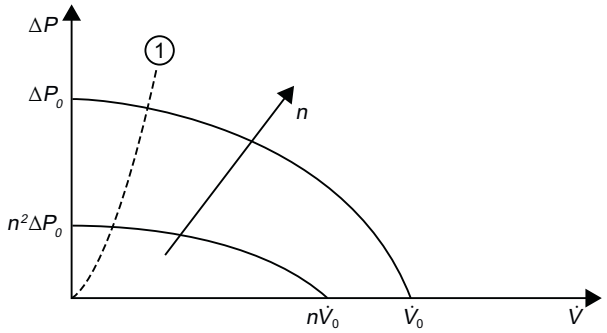
$$\Delta P = n^2 \Delta P_0 + \left(\Delta P^* - \Delta P_0\right) \frac{\dot{V}^2}{\left(\dot{V}^*\right)^2} \quad \text{für} \quad \dot{V} > 0$$

ΔP	aktuelle Druckerhöhung des Kompressors [bar]
ΔP^*	Nenndruckerhöhung des Kompressors am Auslegungspunkt [bar]
ΔP_0	Nullförderhöhe des Kompressors [bar]
n	aktuelle Drehzahl normiert [0..1]
$\dot{v}$	aktueller Förderstrom des Kompressors [Nm³/s]
$\dot{v}^*$	Nennförderstrom des Kompressors am Auslegungspunkt [Nm³/s]

Für den Förderstrom $\dot{v}$ ist die Bezugsrichtung vom Anschluss *A* zum Anschluss *B* definiert, d. h. für einen Förderstrom in Bezugsrichtung gilt $\dot{v} > 0$.

Die Drehzahl wird als prozentualer Wert am Eingang *Speed* vorgegeben. Der Drehzahlwert wird begrenzt auf den Bereich $0 \le N \le 100\ \%$. Folglich gilt n = N / 100 % und damit $0 \le n \le 1$. Dieser Drehzahlwert kann z. B. über eine I/O-Ebene wie in Kopplung Simulationsmodell mit Aktor-Sensor-Ebene (Seite 627) beschrieben mit einer Leittechnik unter Verwendung der Antriebe aus der Basisbibliothek von SIMIT gekoppelt werden.

Die folgende Abbildung veranschaulicht den oben definierten quadratischen Zusammenhang zwischen Druckerhöhung und Förderstrom beim Betrieb des Kompressors im Normalbereich, d. h. für $\dot{v} > 0$, $\Delta P > 0$.



① Kompressorgrenze

Kompressorkennlinie qualitativ

Für die Temperaturerhöhung durch die Kompression des Gases wird eine isentrope Zustandsänderung angenommen:

$$T_B = T_A \left(\frac{P_B}{P_A} \right)^{\frac{K-1}{K}}$$

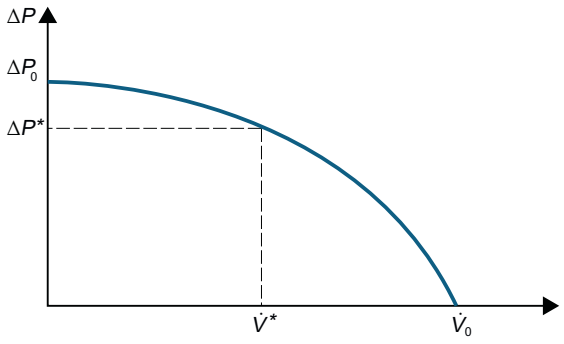
Bei einer Flussumkehr ($\dot{v} < 0$) gibt es keine Druckerhöhung. Der Kompressor verhält sich in diesem Fall wie ein Rückschlagventil. Die Drosselwirkung stellen Sie in diesem Fall über den Parameter *Kv0* ein.

Hinweis

Die Komponente *Compressor* verdichtet nur Gase. Bei Flüssigkeiten und Wasser fungiert die Pumpe nur als Strömungswiderstand mit dem Widerstandsbeiwert *Kv0*. Wasser/Dampf wird bis zu einem Flüssiganteil von 5 Ma.% verdichtet.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ZeroFlowHead	Nullförderhöhe ΔP_0, $\Delta P_0 > \Delta P^*$	bar	5
NominalPressure	Nenndruck ΔP^*; $\Delta P^* > 0$	bar	4
NominalVolumeflow	Nennmassenstrom $\dot{v}^*$, $\dot{v}^* > 0$	Nm³/s	1



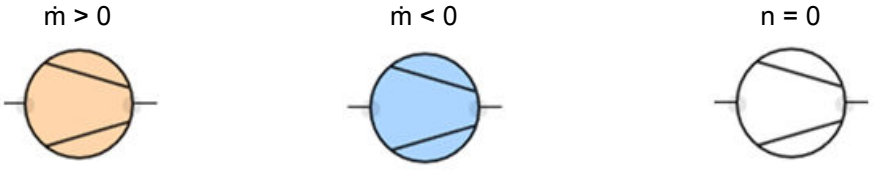
ΔP_0 Nullförderhöhe [bar]
ΔP^* Nenndruck [bar]
$\dot{v}^*$ Nennvolumenstrom [kg/s]

Grafische Darstellung der Parameter des Komponententyps *Compressor*

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Durchflusskoeffizient k_v	m³/h	0.001
Pmin	Minimal erzeugbarer Druck auf der Saugseite	bar	0.01
InitialEnth	Enthalpie des Gases bei Initialisierung	kJ/kg	100
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Der Betriebszustand im Symbol des Kompressors wird wie folgt angezeigt:



Darstellung des Betriebszustands des Kompressors am Symbol

Bedienfenster

Im Bedienfenster schalten Sie zwischen automatisierter Fahrweise (Steuerung über die Aktor-Sensor-Ebene) und manueller Fahrweise um. Voreingestellt ist immer die automatisierte Fahrweise. Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Local control	–	–
"Slider"	–	%

Im erweiterten Bedienfenster des Komponententyps *Compressor* wird die Kompressorkennlinie grafisch dargestellt. Darüber hinaus werden folgende Größen angezeigt:

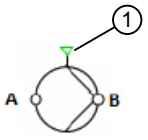
Größe	Formelzeichen	Einheit
Druckerhöhung	ΔP	bar
Förderstrom	v̇	Nm³/s
Nenndruck	Ordinate	bar
Nennvolumenstrom	Abszisse	Nm³/s

Siehe auch

Geodätische Höhe (Seite 612)

Pump – Pumpe

Symbol



① Speed

Bild 8-45 Symbol des Komponententyps Pump

Funktion

Der Komponententyp *Pump* berechnet die Druckerhöhung durch eine Pumpe in Abhängigkeit vom Förderstrom und der Drehzahl gemäß folgender Formel:

$$\Delta p = n^2 \Delta p_0 + \left(\Delta p^* - \Delta p_0\right) \frac{\dot{m}^2}{\left(\dot{m}^*\right)^2} \quad \text{für} \quad \dot{m} > 0$$

Δp	aktuelle Druckerhöhung der Pumpe [bar]
Δp^*	Nenndruckerhöhung der Pumpe am Auslegungspunkt [bar]
Δp_0	Nullförderhöhe der Pumpe [bar]
n	aktuelle Drehzahl normiert [0..1]
$\dot{m}$	aktueller Förderstrom der Pumpe [kg/s]
$\dot{m}^*$	Nennförderstrom der Pumpe am Auslegungspunkt [kg/s]

Für den Förderstrom $\dot{m}$ ist die Bezugsrichtung vom Anschluss *A* zum Anschluss *B* definiert, d. h. für einen Förderstrom in Bezugsrichtung gilt $\dot{m} > 0$.

Die Drehzahl wird als prozentualer Wert am Eingang *Speed* vorgegeben. Der Drehzahlwert wird begrenzt auf den Bereich $0 \le N \le 100\ \%$. Folglich gilt n = N / 100 % und damit $0 \le n \le 1$. Dieser Drehzahlwert kann z. B. über eine I/O-Ebene wie in Kopplung Simulationsmodell mit

Aktor-Sensor-Ebene (Seite 627) beschrieben mit einer Leittechnik unter Verwendung der Antriebe aus der Basisbibliothek von SIMIT gekoppelt werden.

Die folgende Abbildung veranschaulicht den oben definierten quadratischen Zusammenhang zwischen Druckerhöhung und Förderstrom beim Betrieb der Pumpe im Normalbereich, d. h. für $\dot{m} > 0$, $\Delta p > 0$.

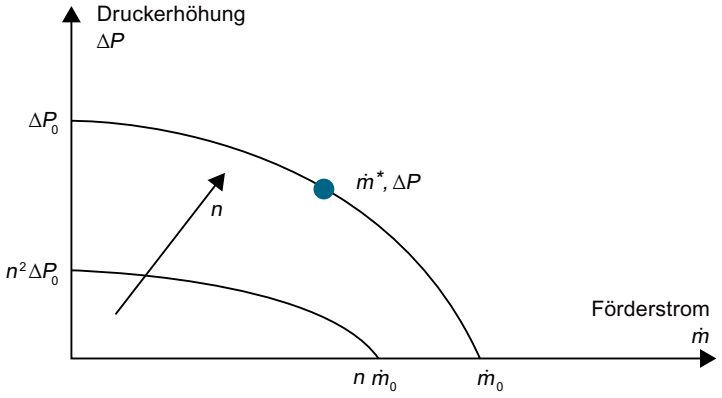


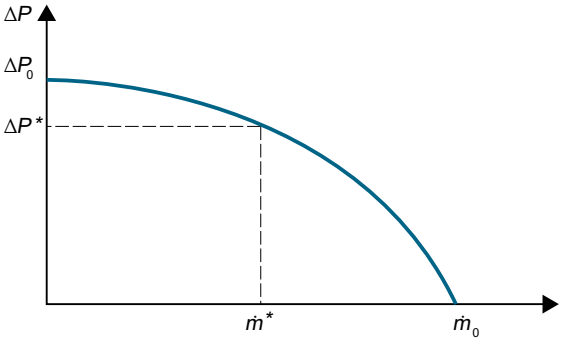
Bild 8-46 Pumpenkennlinie qualitativ

Bei einer Flussumkehr ($\dot{m} < 0$) gibt es keine Druckerhöhung. Die Pumpe verhält sich in diesem Fall wie ein Rückschlagventil. Die Drosselwirkung stellen Sie in diesem Fall über den Parameter *Kv0* ein.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ZeroFlowHead	Nullförderhöhe Δp_0, $\Delta p_0 > \Delta p^*$; online änderbar	bar	5.0
NominalPressure	Nenndruck Δp^*; $\Delta p^* > 0$; online änderbar	bar	4.0
NominalMassflow	Nennmassenstrom $\dot{m}^*$, $\dot{m}^* > 0$; online änderbar	kg/s	1.0

Mit Parametern werden die bestimmenden Größen der Pumpenkennlinie eingestellt:



Δp_0 Nullförderhöhe [bar]

Δp^* Nenndruck [bar]

$\dot{m}^*$ Nennmassenstrom [kg/s]

Bild 8-47 Grafische Darstellung der Parameter des Komponententyps Pump

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Drosselwirkung der Pumpe im Falle der Flussumkehr	m^3/h	0.001
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Der Betriebszustand im Symbol der Pumpe wird wie folgt angezeigt (siehe Abbildung):

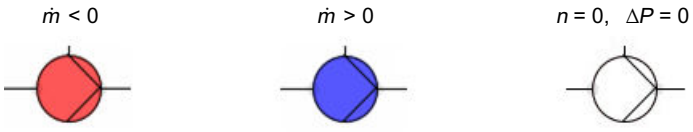


Bild 8-48 Darstellung des Betriebszustands der Pumpe am Symbol

Bedienfenster

Im Bedienfenster schalten Sie zwischen automatisierter Fahrweise (Steuerung über die Aktor-Sensor-Ebene) und manueller Fahrweise um. Voreingestellt ist immer die automatisierte Fahrweise.

Bei manueller Fahrweise geben Sie die prozentuale Pumpendrehzahl über einen Slider vor.

Im erweiterten Bedienfenster wird die Pumpenkennlinie grafisch dargestellt. Die charakteristischen Werte der Kennlinie und der aktuelle Betriebspunkt werden über Digitalanzeigen dargestellt. Zusätzlich werden die aktuelle Druckerhöhung Δp und der Förderstrom $\dot{m}$ angezeigt.

Siehe auch

Geodätische Höhe (Seite 612)

8.2.5.8 Separators

Centrifuge – Zentrifuge

Symbol

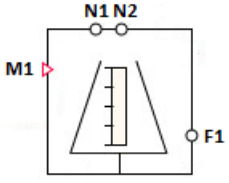


Bild 8-49 Symbol des Komponententyps Centrifuge

Funktion

Der Komponententyp *Centrifuge* dient zur Trennung eines aus zwei Stoffen bestehenden Gemischs (Suspension oder Emulsion) in seine Bestandteile. Zur Trennung von Gasgemischen kann diese Komponente nicht verwendet werden.

Die Trennung in einer Zentrifuge erfolgt aufgrund der Massenträgheit während des Betriebs der Zentrifuge (Rotation der Zentrifugenkammer). Hierbei gelangen die Partikel mit höherer Dichte nach außen, oder es wird ein Feststoff aufgrund von Einbauten (z. B. Sieb oder Filtertuch) zurückgehalten.

Im vorliegenden Modell wird die Trennung durch Definition eines abscheidbaren und eines nicht abscheidbaren Gemischbestandteils simuliert. Die Summe beider Anteile ergibt 100 %. Dichteunterschiede werden bei der Abtrennung nicht berücksichtigt.

Im Folgenden wird der nach außen gelangte Teilstoff als Filtrat bezeichnet (abscheidbarer Gemischbestandteil). Der in der Zentrifugenkammer verbliebene Teilstoff wird im Folgenden als Filterkuchen bezeichnet (nicht abscheidbarer Gemischbestandteil).

Der Filterkuchen kann durch Zugabe von Waschwasser resuspendiert und gewaschen werden. Durch Zugabe von unter Druck stehendem Gas kann Flüssigkeit aus dem Filterkuchen gepresst werden.

Beschaltung

Die Zentrifuge besitzt zwei Typen von Anschlüssen für Stoffströme und somit zur Verschaltung von Rohrleitungen:

- **N1 ... Nn – allgemeine Anschlussstutzen**
  Diese Anschlusspunkte dienen zum Befüllen und Entleeren der Zentrifuge und als Gasanschlüsse.
  Alle Anschlüsse des Vektors *N1..Nn* sind zunächst gleichwertig und können für alle Anschlusstypen verwendet werden (Wasser/Dampf, ideales Gas, Liquid)
- **F1 … Fn – Filtratablauf**
  Diese Anschlusspunkte sind ein Sonderfall. Über diese Anschlüsse erfolgt ausschließlich der Austritt des abgeschiedenen Filtrats. Alle anderen Anschlüsse erfolgen über den bereits beschriebenen Vektor *N1..Nn.*

Die Anzahl der Anschlusspunkte hängt vom zu simulierenden System ab und wird über die Parameter *NbrOfInletsOutlets* für die Anschlüsse *N1 ... Nn* und *NbrOfFiltrateOutlets* für die Anschlüsse *F1 ... Fn* vorgegeben.

Das Stoffmedium in den angrenzenden Flussnetzen wird über die Komponente *NetParam* definiert.

Die Mindestverschaltung ist:

- Emulsions-/Suspensionseintritt (*Nn*)
- Filterkuchenaustritt (Gemischanteil mit niederer Dichte oder in der Zentrifugenkammer zurückgehalten) (*Nn*)
- Filtrataustritt (aus der Zentrifugenkammer herausgeschleuderter Gemischanteil) (*Fn*)

Zusätzliche Verschaltungen können beispielsweise sein:

- Ein-/Austritt für Reinigungsmedien
- Ein-/Austritt für Waschlösungen

- Gasein- und Gasaustritte
- Belüftung

Beim Anschlusstyp Wasser/Dampf wird das einströmende Medium intern im Tank mit einem Liquid-Medium vermischt. Die einströmende Masse und Enthalpie wird berücksichtigt, im Modell bildet sich jedoch kein entsprechendes Phasengleichgewicht aus. Einströmender Dampf wird in der Massenbilanz der Flüssigphase zugerechnet und trägt in der Gasphase nicht zum Druckaufbau bei. Wasser wird im Gemisch wie ein Liquid-Medium behandelt.

Wenn Wasser/Dampf als Austrittsmedium verwendet wird, so verlässt es die Zentrifuge am Ausgang mit der Gemischenthalpie der Flüssigphase. Alle weiteren Berechnungen innerhalb dieses Flussnetzes erfolgen dann unter Nutzung der Wasser/Dampf-Stoffeigenschaften (z. B. Wärmekapazität).

Aufgrund der fehlenden Dampfeigenschaften ist derzeit z. B. eine Dampfsterilisation auch im Stillstand nicht möglich.

Parameter ConnectorHeight

Zu jedem der Anschlusspunkte *N1..Nn* und *F1..Fn* wird die Einbauhöhe des Stutzens an der Zentrifuge benötigt (Parameter *ConnectorHeight* für Anschlüsse *N1..Nn* bzw. *ConnectorHeightF* für den Anschlüsse *F1..Fn*). Die Angabe ist in Metern. Die Voreinstellung ist jeweils 0 m und somit am Boden der Zentrifuge. Für die Nachbildung von tiefer liegenden Anschlüssen sind negative Werte zulässig.

Eingangsvektor ScreeningRate

Über den Eingangsvektor *ScreeningRate* kann für jeden der Anschlüsse *N1..Nn* der Anteil des nicht abscheidbaren Gemischbestandteils des einströmenden Mediums in Prozent vorgegeben werden. Damit wird für den Emulsions- oder Suspensionseintritt der nach der Abscheidung zurückbleibende Anteil definiert. Anschlüsse ohne Abscheideanteile (z. B. Waschmittel) werden mit dem Wert "0" (Voreinstellung) belegt. Gasanschlüsse erhalten ebenfalls den Wert "0".

Die hier vorgegebenen Werte gelten ausschließlich für eintretende Stoffströme. Alle austretenden Stoffströme strömen mit dem in der Zentrifuge aktuell gespeicherten nicht abscheidbaren Gemischbestandteil der Gesamtmasse *Rate* aus. Dieser Bestandteil wird aus den ein- und austretenden Massenströmen und der jeweiligen *ScreeningRate* laufend berechnet.

Im Fall von austretenden Massenströmen wird eine gegebenenfalls eingetragene *ScreeningRate* nicht ausgewertet.

Die Filtrataustritte *F1..Fn* erfolgen immer als reines Filtrat (d. h. *Rate* = 0.0). Zum Filtratanschluss *F1..Fn* gibt es somit keinen Anschluss *ScreeningRate*.

Der Eingangsvektor *ScreeningRate* ist als default nicht sichtbar.

Alternativ können Sie den nicht abscheidbaren Gemischbestandteil *Rate* der **befüllten Zentrifuge** nach dem Befüllvorgang auch manuell über das Bedienfenster setzen.

Bilanzierung der Flüssigphase

Alle definierten Eingänge in den Anschlussvektoren *N1 .. Nn* und *F1..Fn* müssen auch verschaltet werden. Während der Simulation darf keiner dieser Anschlüsse offen bleiben, da hier sonst offene Flussnetze angelegt werden. Dies führt bei der Codegenerierung zu einer Fehlermeldung.

Die Flüssigphase wird bezüglich der ein- und austretenden Massen (m_{Liquid}) und Enthalpien (h_{Liquid}) bilanziert. Zusätzlich erfolgt eine Massenbilanz des nicht abscheidbaren Gemischbestandteils ($m_{Screening}$).

$$\frac{dm_{Liquid}}{dt} = \sum_{i=1}^{n} \dot{m}_{Liquid,i}$$

$$\frac{dm_{Screening}}{dt} = \sum_{i}^{n} \dot{m}_{Screening,i}$$

$$\frac{dh_{Liquid}}{dt} = \frac{1}{m_{Liquid}} \left(\sum_{i}^{n} \dot{m}_{Liquid,i}\, h_{Liquid,i} + \sum \dot{Q} \right)$$

$$T_{Liquid} = \frac{h_{Liquid}}{c_P}$$

Gasbilanzierung

Für den Fall, dass mindestens ein Gasanschluss angeschlossen ist, wird auch ein entsprechender Gasdruck berechnet. In diesem Fall wird von einer geschlossenen, also druckdichten Zentrifuge ausgegangen. Durch Ein- oder Ausspeichern von Gas oder Veränderung des Gasvolumens, z. B. durch Füllen des Tanks, wird der Gasdruck verändert. Der berechnete Gasdruck wirkt ebenfalls auf die Flüssigkeitsanschlüsse.

Die Gasbilanzierung bezieht sich ausschließlich auf von außen zugegebene (Inert-)Gase. Es werden z. B. keine Dampfdrücke von Lösemitteln berücksichtigt.

Innerhalb des Gasraums werden die ein- und austretenden Massen (m_{Gas}) und Enthalpien (h_{Gas}) bilanziert:

$$\frac{dm_{Gas}}{dt} = \sum_{i}^{n} \dot{m}_{Gas,i}$$

$$\frac{dh_{Gas}}{dt} = \frac{1}{m_{Gas}} \left(\sum_{i}^{n} \dot{m}_{Gas,i}\, h_{Gas,i} + \sum \dot{Q} \right)$$

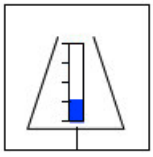
Der Gasdruck wird dann entsprechend der Zustandsgleichung idealer Gase berechnet:

$$p_{Gas} = \frac{m_{Gas}\, R_S\, T_{Gas}}{V_{Gas}}$$

Falls kein Gasanschluss verbunden ist, entfällt die Gasbilanzierung. In diesem Fall wird der Gasdruck durch den Umgebungsdruck ersetzt.

Der Komponententyp *Centrifuge* besitzt unterschiedliche Symbole für den Betrieb mit und ohne Gasdruck.

Zentrifuge ohne Gasdruck



Zentrifuge mit Gasdruck



Innerhalb des Komponententyps wird nicht zwischen Suspension und Emulsion unterschieden. Auch im Fall der Suspension wird intern mit zwei Flüssigkeiten gerechnet. Der entsprechende Austrag des Filterkuchens einer Suspension erfolgt somit ebenfalls als "flüssiges Medium". Entsprechende Geräte zum Austrag (Schaber, Förderschnecken o.ä.) müssen somit im Modell durch Ventile und Pumpen ersetzt werden.

Betriebsarten der Zentrifuge

Die Zentrifuge ist gleichermaßen für den kontinuierlichen wie für den diskontinuierlichen Betrieb ausgelegt.

Der Betrieb der Zentrifuge wird durch Aufgabe einer entsprechenden normierten Winkelgeschwindigkeit (0 % … 100 %) am Eingang *Omega* gestartet. Filtrat kann die Zentrifuge erst verlassen, wenn der entsprechende Filtratablauf geöffnet ist. Die Erhöhung der Winkelgeschwindigkeit führt zu einer Erhöhung des Austrittsdrucks des Filtrats. Der Austrittsdruck wird vereinfacht aus dem Radius der Zentrifuge (aus geometrischen Daten ermittelt) und der Winkelgeschwindigkeit berechnet.

Im Fall einer stillstehenden Zentrifuge kann trotzdem Filtrat austreten, da hier kein Unterschied zwischen Filtrat und z. B. Reinigungsmittel gemacht wird. Falls dies verhindert werden soll, kann z. B. ein zusätzliches Ventil in das Modell eingefügt werden, dessen Stellung in Abhängigkeit von der Drehzahl verfahren wird.

Der absolute Wert des Filtratmassenstroms kann über k_V-Werte im Filtratablauf eingestellt werden (z. B. Ventil oder Nozzle).

Überschreiben von Zustandsgrößen

Über das Bedienfenster der Komponente kann der Bediener während der Simulation folgende Zustandsgrößen auf beliebige (zulässige) Werte setzen:

- Gespeicherte Masse / Füllstand
- Temperatur
- Nicht abscheidbarer Massenanteil *Rate*

Hierzu muss der Bediener zunächst den Sollwert in die entsprechende Digitaleingabe *SetValues* eingeben und mit dem Button "Set" übernehmen. Der Zustand der gespeicherten

Gesamtmasse kann hierbei sowohl über den Wert *MassTotal* als auch über den Füllstand (*Level*) vorgegeben werden.

Über diese Beeinflussung von Zustandsgrößen können Sie z. B. Aufwärmvorgänge oder Füllvorgänge beschleunigen. Ebenso können Sie als Bediener die Zentrifuge entleeren, um beispielsweise in einer Schrittkette zurückzuspringen.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Füllvolumen der Zentrifuge	m^3	1.0
Height	Höhe des Füllvolumens der Zentrifuge	m	1.0
NbrOfInletsOutlets	Anzahl von allgemeinen Flussnetzanschlüssen	–	2
NbrOfFiltrateOutlets	Anzahl von Filtratanschlüssen	–	1
NbrOfMeasurements	Anzahl Messstellenanschlüsse	–	1
MeasurementHeight [Nr]	Einbauhöhe der Messstellenanschlüsse	m	0.0
ConnectorHeight [Nr]	Einbauhöhe der allgemeinen Flussnetzanschlüsse	m	0.0
ConnectorHeightF [Nr]	Einbauhöhe der Filtratanschlüsse	m	0.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureOutside	Umgebungsdruck	bar	1.0
LevelInit	Initialisierungswert Füllstand der Zentrifuge	%	3.0
TemperatureInit	Initialisierungswert Temperatur der Suspension	°C	20.0
PressureInit	Initialisierungswert Druck im Fall einer Gasdruckzentrifuge	bar	1.0
InitScreeningRateTotal	Initialisierungswert Feststoffanteil in der Suspension (nicht abscheidbarer Anteil)	%	0.0
Density	Dichte der Suspension	kg/m^3	997.337
RemainderGas	Restvolumen, das beim Füllen der Zentrifuge nicht mit Flüssigkeit gefüllt werden kann	m^3	0.05
RemainderLiquid	Restvolumen, das beim Entleeren der Zentrifuge in der Zentrifuge verbleibt	m^3	0.005
CpInit	Initialwärmekapazität für das während der Initialisierung in der Zentrifuge gespeicherte Medium	–	4.18
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Länge	L	m	Ja[1]
Temperatur	T	°C	Ja[1]
Gewicht	M	kg	Ja[1]
Druck	p_B	bar	Nein
Gasdruck	p_G	bar	Nein
Anteil Feststoff im Feststoff/Flüssiggemisch	w	%	Ja[1]

[1] Eingabe mit "Set" aktivieren

Beispielverschaltung

Eine beispielhafte Verschaltung der Zentrifuge entnehmen Sie dem folgenden Bild:

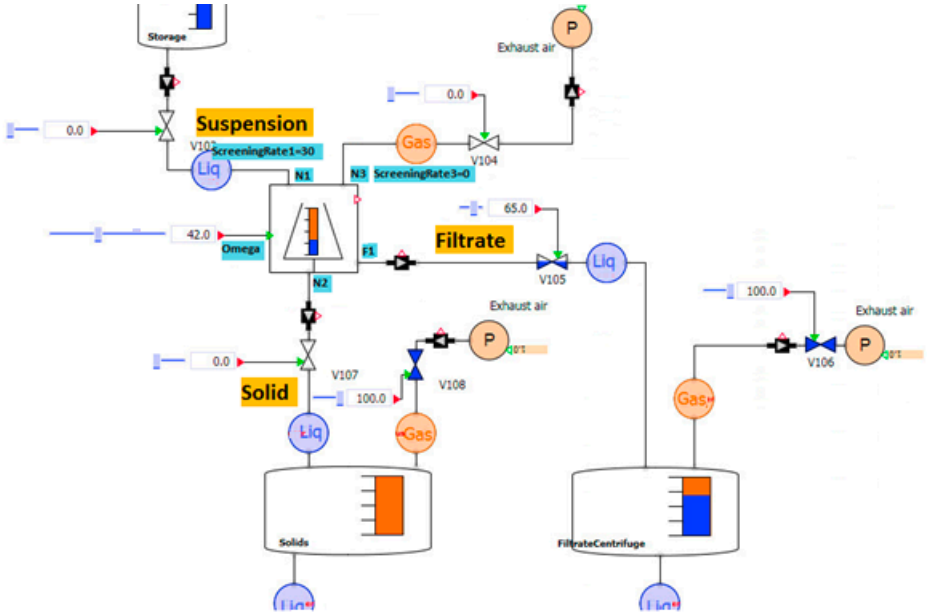


Bild 8-50 Beispielverschaltung des Komponententyps Centrifuge

Hierzu nehmen Sie folgende Parametrierungen vor:

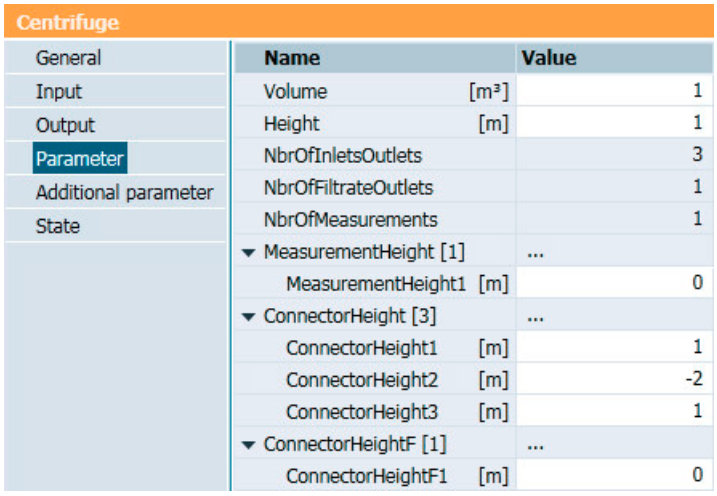


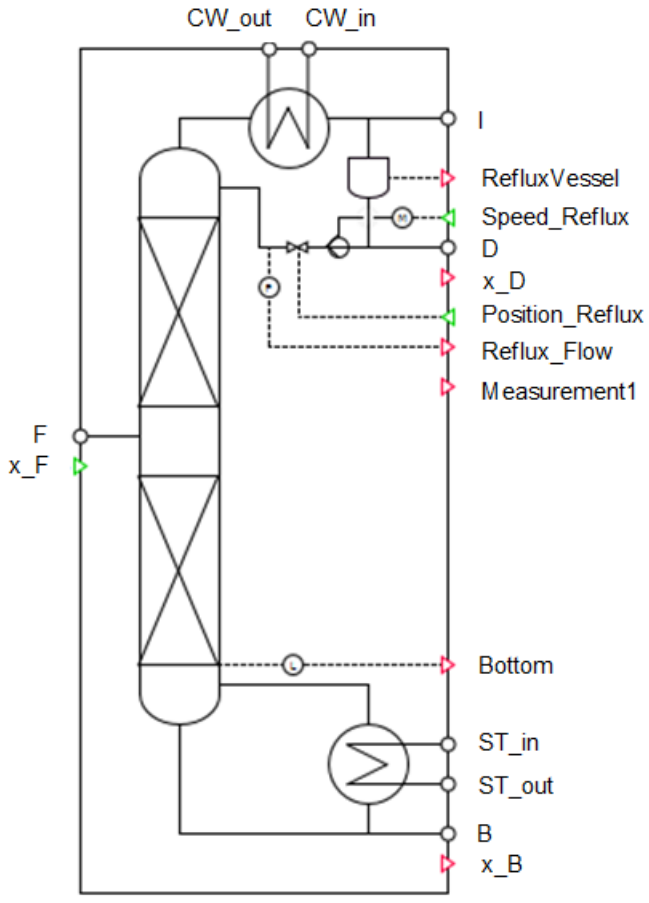
Bild 8-51 Parametrierung des Komponententyps Centrifuge in der Beispielverschaltung

Siehe auch

Geodätische Höhe (Seite 612)

Column – Kolonne

Symbol



Symbol des Komponententyps Column

Funktion

Der Komponententyp *Column* berechnet vereinfacht die Destillation eines binären Zulauf-Stroms in ein leichtsiedendes Gemisch (Kopfprodukt) und ein schwersiedendes Gemisch (Sumpfprodukt) in einer Bodenkolonne.

Der Zulauf wird über den Anschluss *F* mit der Konzentration an Leichtsieder *x_F* der Kolonne zugeführt. Das Kopfprodukt (Destillat) wird der Kolonne am Anschluss *D* mit der Konzentration an Leichtsiedern *x_D* entnommen. Das Sumpfprodukt (Bottom) wird am Anschluss *B* mit der Konzentration an Leichtsiedern *x_B* der Kolonne entnommen.

Am Kopf der Kolonne ist ein Kondensator mit Rücklaufbehälter vorhanden. Der Kondensator hat zwei Anschlüsse für Kühlwasser (*CW_in* und *CW_out*). Aus dem Rücklaufbehälter wird mit einer Pumpe (entweder im Ablauf vom Rücklaufbehälter oder in der Rücklaufleitung) der Rücklauf zur Kolonne gefördert. Nicht kondensierte Bestandteile des Kopfstroms der Kolonne werden über den Anschuss *I* der Kolonne entnommen.

Im Sumpf wird die Kolonne über einen Naturumlaufverdampfer (Reboiler) beheizt. Der Reboiler hat zwei Anschlüsse für Heizdampf (*ST_in* und *ST_out*).

Die Trennung des Leichtsieders und Schwersieders wird idealisiert über so genannte Split-Faktoren berechnet. Die Split-Faktoren werden mit empirischen Formeln in Abhängigkeit von Rücklaufmenge, Rückdampfmenge, Zulaufmenge und Zulaufkonzentration im Vergleich zum (stationären) Auslegungspunk (Operation Point OP) ermittelt. Der stationäre Arbeitspunkt der Kolonne wird mit folgenden Parametern spezifiziert:

- Zulaufmenge: *OP_mflF*
- Zulaufkonzentration: *OP_xF*
- Kopfkonzentration: *OP_xD*
- Sumpfkonzentration: *OP_xB*
- Rücklaufmenge: *OP_mflR*
- Rückdampfmenge: *OP_mflV*
- Verdampferleistung: *OP_Qreb*

Mit dem Parameter *Split_Max* werden die Split-Faktoren begrenzt. Der maximale Split-Faktor beträgt *Split_Max*, der minimale (1 − *Split_Max*). Je größer der einzelne Split-Faktor ist, desto mehr wird vom Leicht- oder Schwersiederüber den Kopfproduktstrom abgetrennt.

Die Temperaturen in der Kolonne werden in Abhängigkeit der Konzentration des Leichtsieders *x* und der Siedepunkte des Leicht- und Schwersieders (*TS_LB* und *TS_HB*) bei Kolonnenauslegungsdruck berechnet.

$$T = \left(x T_{S_{LB}}\right) + \left((1-x) T_{S_{HB}}\right)$$

Die Verdampfungsenthalpie *hV* wird in Abhängigkeit der Konzentration des Leichtsieders *x* und den Verdampfungsenthalpien des Leicht- und Schwersieders (*hV_LB* und *hV_LB*) berechnet.

$$h_V = \left(x h_{V_{LB}}\right) + \left((1-x) h_{V_{HB}}\right)$$

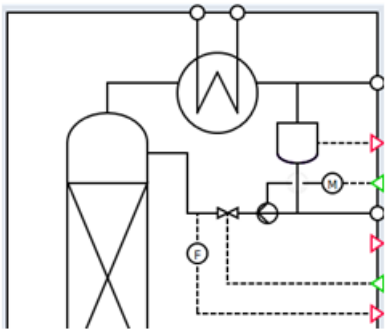
Der Naturumlauf im Reboiler wird mit Hilfe der Verdampferleistung *Q_Reb* im Vergleich zu der Rücklaufmenge *OP_mflV* und Verdampferleistung *OP_Qreb* am stationären Arbeitspunkt berechnet. Die Verdampferleistung ergibt sich aus der Heizleistung, die vom Heizmedium im Reboiler übertragen wird.

Die Größe der Kolonne wird über folgende Parameter festgelegt:

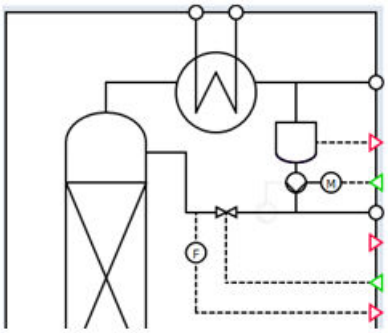
- Durchmesser der Kolonne: *Diameter*
- Anzahl Böden: *n_tray*
- Abstand der Böden: *Tray_Height*

Der Zulaufboden der Kolonne wird mit dem Parameter *Feed_Tray* festgelegt.

Mit dem Parameter *Pump_Top* kann ausgewählt werden, ob sich die Pumpe am Rücklaufbehälter nur in der Rücklaufleitung befindet (*Reflux*) oder ob die Pumpe im Ablauf des Rückflussbehälters für Rücklauf und Destillat befindet (*Reflux+Distillate*).



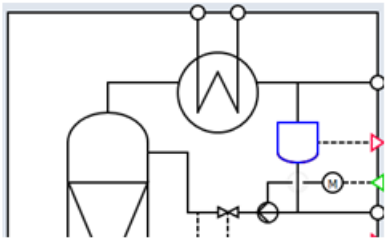
Darstellung: Pumpe im Rücklauf



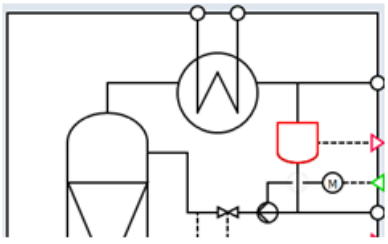
Pumpe für Rücklauf und Destillat

Der Kopfkondensator wird über die Parameter Austauschfläche *A_Cond* und Wärmedurchgangskoeffizient *k_Cond* konfiguriert.

Der Rücklaufbehälter wird über Parameter Durchmesser *Diameter_RefluxVessel* und Volumen *Volume_RefluxVessel* konfiguriert. Wenn der Rücklaufbehälter leerläuft, wird der Behälter mit blauen Linien dargestellt. Wenn der Behälter vollständig gefüllt ist und überläuft, wird der Behälter mit roten Linien dargestellt.



Darstellung: Rücklaufbehälter leer



Rücklaufbehälter voll

Der Reboiler wird über die Parameter Austauschfläche *A_Reb* und Wärmedurchgangskoeffizient *k_Reb* konfiguriert.

Über den Messstellenanschluss *RefluxVessel* kann der Füllstand, Druck und Temperatur im Rücklaufbehälter ausgelesen werden.

Über den Messstellenanschluss *Reflux_Flow* kann der Durchfluss vom Rücklauf zum Kolonnenkopf ausgelesen werden.

Über den Messstellenanschluss *Bottom* kann der Füllstand Druck und Temperatur im Sumpf der Kolonne ausgelesen werden.

Messungen (Druck und Temperatur) an der Kolonne können über Messstellenanschlüsse *Measurement[i]* ausgelesen werden. Über den Parameter *NbrOfMeasurements* kann die Anzahl festgelegt werden. Über die Parameter *MeasurementTray* kann für jede Messstelle der Boden, auf dem die Messstelle liegt, angegeben werden.

Die Pumpendrehzahl kann über den Eingang *Speed_Reflux* vorgegeben werden. Die Drehzahl kann zwischen 0 (Pumpe steht) und 100 (voll Drehzahl) vorgegeben werden.

Die Ventilstellung des Ventils in der Rücklaufleitung kann über den Eingang *Position_Reflux* vorgegeben werden. Die Ventilstellung kann zwischen 0 (Ventil ganz geschlossen) und 100 (Ventil ganz geöffnet) vorgegeben werden.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Diameter	Durchmesser Kolonne	m	2
N_tray	Anzahl der Böden	–	10
Tray_Height	Abstand zwischen den Böden	mm	50
Feed_tray	Zulaufboden	–	10
A_Cond	Austauschfläche Kopfkondensator	m²	50
K_Cond	Wärmedurchgangskoeffizient Kopfkondensator	W/m²/K	1000
ZeroFlowHead_Reflux	Nullförderhöhe Pumpe am Rücklaufbehälter	bar	5
NominalPressure_Reflux	Druck am Arbeitspunk der Pumpe am Rücklaufbehälter	bar	4
NominalMassflow_Reflux	Massenstrom am Arbeitspunkt der Pumpe am Rücklaufbehälter	kg/s	5
Diamter_RefluxVessel	Durchmesser Rücklaufbehälter	m	1
Volume_RefluxVessel	Volumen Rücklaufbehälter	m³	1
geoHeight_RefluxVessel	Geodätische Höhe Rücklaufbehälter	m	0
Pump_Top	Pumpenkonfiguration am Rücklaufbehälter	Reflux	
OP_mflF	Zulaufmassenstrom am Arbeitspunkt	kg/s	6
OP_xF	Zulaufkonzentration am Arbeitspunkt	kg/kg	0.4
OP_xD	Destillatkonzentration am Arbeitspunkt	kg/kg	0.9
OP_xB	Sumpfproduktkonzentration am Arbeitspunkt	kg/kg	0.1
OP_mflR	Rücklaufmassenstrom am Arbeitspunkt	kg/s	4.5
OP_mflV	Rückdampfmassenstrom am Arbeitspunkt	kg/s	17
OP_QReb	Heizleistung Reboiler am Arbeitspunkt	kW	7000
A_Reb	Austauschfläche Reboiler	m²	400
K_Reb	Wärmedurchgangskoeffizient Reboiler	W/m²/K	1000
geoHeight_Colum	Geodätische Höhe Kolonne (am Sumpf)	m	0
NbrOfMeasuremnsts	Anzahl Messstellen an der Kolonne	–	1
MeasurementTray	Boden auf dem sich die Messstelle befindet	–	10

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
LiqLevTray	Flüssigkeitshöhe auf Kolonnenboden	mm	50
hV_LB	Verdampfungsenthalpie Leichtsieder	kJ/kg/K	1100
hV_HB	Verdampfungsenthalpie Schwersieder	kJ/kg/K	2200
TS_LB	Siedetemperatur Leichtsieder bei Kolonnendruck	°C	65
TS_HB	Siedetemperatur Schwersieder bei Kolonnendruck	°C	100
DTmin	Minimale Temperaturdifferenz Wärmetauscher	K	5
Split_Max	Maximaler Trennfaktor	–	0.9999
LevelPercent_RefluxVessel_Init	Initialisierung Füllstand Rücklaufbehälter	%	50
LevelPercent_Bottom_Init	Initialisierung Füllstand Kolonnensumpf	%	50
kvs_CW	Widerstandsbeiwert Strömung Kopfkondensator Servicemedium	m³/h	200
kvs_Reflux	Widerstandsbeiwert Ventil Rücklauf	m³/h	10
pInit_Top	Initialisierung Druck Kolonnenkopf	bar	1
PInit_Bottom	Initialisierung Druck Kolonnensumpf	bar	1.1

Bedienfenster

Im Bedienfenster werden die wichtigsten Kenngrößen der Kolonne angezeigt. Es werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Kopfdruck	p_Kopf	bar
Kopftemperatur	T_Kopf	°C
Kühlleistung Kopfkondensator	Q_Cond	kW
Rücklaufmassenstrom	mfl_R	kg/s
Gasmassenstrom Kopf	mfl_I	kg/s
Kopfproduktstrom	mfl_D	kg/s
Konzentration Kopfprodukt	x_D	kg/kg
Rücklaufverhältnis	RR	–
Zulaufmassenstrom	mfl_F	kg/s
Zulaufkonzentration	x_F	kg/kg
Rückdampfverhältnis	BR	–
Rückdampfmassenstrom	mfl_V	kg/s
Verdampferleistung	Q_Reb	kW
Sumpftemperatur	T_Sumpf	°C
Sumpfdruck	p_sumpf	bar

Größe	Formelzeichen	Einheit
Sumpfproduktstrom	mfl_B	kg/s
Konzentration Sumpfprodukt	x_B	kg/kg

Column Tower – Kolonnenturm

Symbol

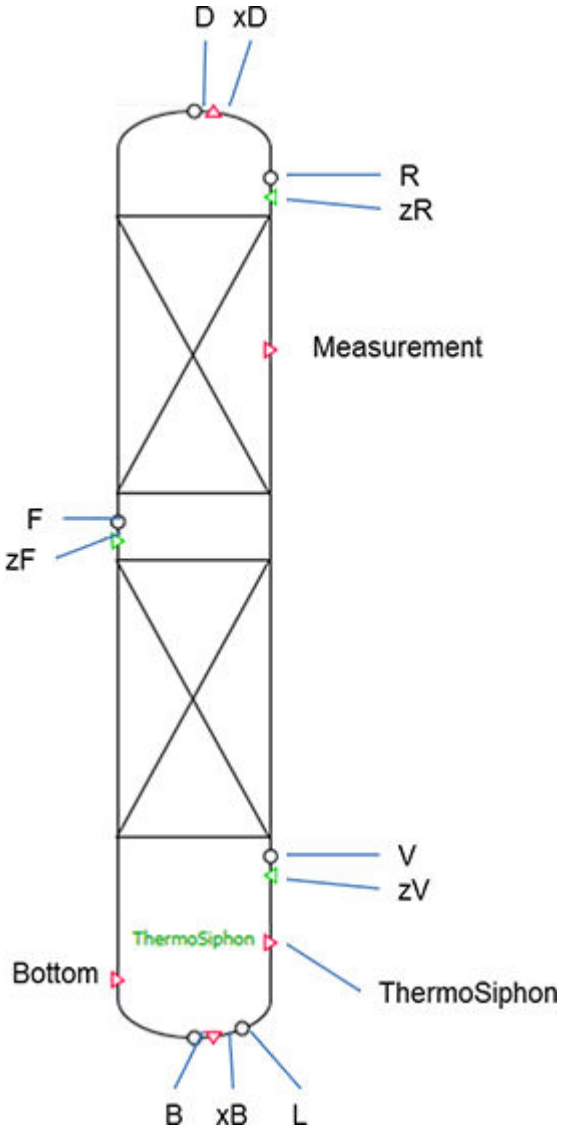


Bild 8-52 Symbol des Komponententyps Column Tower

Funktion

Der Komponententyp *Column Tower* berechnet vereinfacht die Destillation eines binären Zulauf-Stroms in ein leichtsiedendes Gemisch (Kopfprodukt) und ein schwersiedendes Gemisch (Sumpfprodukt) in einer Bodenkolonne.

Der Zulauf wird über den Anschluss *F* mit der Konzentration an Leichtsieder *zF* der Kolonne zugeführt. Das Kopfprodukt (Destillat) wird der Kolonne am Anschluss *D* mit der Konzentration an Leichtsiedern *xD* entnommen. Das Sumpfprodukt (Bottom) wird am Anschluss *B* mit der Konzentration an Leichtsiedern *xB* der Kolonne entnommen. Der Rücklauf wird am Anschluss *R* mit der Konzentration *zR* zugeführt. Der Sumpfstrom zum Reboiler wird am Anschluss *L* abgezogen. Der erwärmte und teilweise verdampfte Strom vom Reboiler wird am Anschluss *V* der Kolonne wieder zugeführt. Die Konzentration an Leichtsieder für diesen Strom wird über den Anschluss *zV* eingegeben.

Die Trennung des Leichtsieders und Schwersieders wird idealisiert über sogenannte Split-Faktoren berechnet. Die Split-Faktoren werden mit empirischen Formeln in Abhängigkeit von Rücklaufmenge, Rückdampfmenge, Zulaufmenge und Zulaufkonzentration im Vergleich zum (stationären) Auslegungspunk (Operation Point OP) ermittelt. Der stationäre Arbeitspunkt der Kolonne wird mit folgenden Parametern spezifiziert:

- Zulaufmenge *OP_mflF*
- Zulaufkonzentration *OP_xF*
- Kopfkonzentration *OP_xD*
- Sumpfkonzentration *OP_xB*
- Rücklaufmenge *OP_mflR*
- k_{VS}-Wert Verdampfer *OP_kvReb* (für Naturumlauf) bzw.
- Umlaufmenge *OP_mflCircForced* (für Zwangsumlauf)

Mit dem Parameter *Split_Max* werden die Split-Faktoren begrenzt. Der maximale Split-Faktor beträgt *Split_Max*, der minimale (1 − *Split_Max*). Je größer der einzelne Split-Faktor ist, desto mehr wird vom Leicht- oder Schwersieder über den Kopfproduktstrom abgetrennt.

Die Temperaturen in der Kolonne werden in Abhängigkeit der Konzentration des Leichtsieders *x* und der Siedepunkte des Leicht- und Schwersieders (*TS_LB* und *TS_HB*) sowie der Spreizung *S* der Siedelinse bei Kolonnenauslegungsdruck berechnet.

$T = f(TS_{LB}, TS_{HB}, S, x)$

Der Anschluss *ThermoSiph* der Kolonne kann mit dem Anschluss *ThermoSiph* des Verdampfers/Reboilers verbunden werden. Dann werden die notwendige Informationen wie Dampfmassengehalt *vf*, und Zusammensetzung der Dampf- und Flüssig-Phase (*x* und *y*) am Ausgang des Reboilers sowie die Sumpfkonzentration der Kolonne *xB* direkt übertragen. Ist dieser Anschluss verbunden, ist eine Vorgabe des Eingangs *zV* nicht mehr notwendig.

Über den Parameter *ThermoSiphon* kann eingestellt werden, dass der Reboiler als Naturumlaufverdampfer betrieben wird. In diesem Fall muss der Anschluss *ThermoSiph* der Kolonne mit dem Anschluss *ThermoSiph* des Verdampfers verbunden werden. Der Naturumlauf zwischen Kolonne und Verdampfer wird dann entsprechend realisiert. Wird der Parameter *ThermoSiphon* nicht gesetzt, wird von einem Zwangsumlauf mit Pumpe ausgegangen. Der Anschluss *ThermoSiph* kann auch hier mit dem Verdampfer verbunden werden.

Die Größe der Kolonne wird über folgende Parameter festgelegt:

- Durchmesser der Kolonne *Diameter*
- Anzahl der Böden *n_tray*
- Abstand der Böden *Tray_Height*

Der Zulaufboden der Kolonne wird mit dem Parameter *Feed_Tray* festgelegt.

Der Rücklaufbehälter wird über Parameter Durchmesser *Diameter_RefluxVessel* und Volumen *Volume_RefluxVessel* konfiguriert. Läuft der Rücklaufbehälter leer, wird der Behälter mit blauen Linien dargestellt. Ist der Behälter vollständig gefüllt und läuft über, wird der Behälter mit roten Linien dargestellt (siehe Abbildung).

Über den Messstellenanschluss *Bottom* kann der Füllstand Druck und Temperatur im Sumpf der Kolonne ausgelesen werden.

Messungen (Druck und Temperatur) an der Kolonne können über Messstellenanschlüsse *Measurement[i]* ausgelesen werden. Über den Parameter *NbrOfMeasurements* kann die Anzahl festgelegt werden. Über die Parameter *MeasurementsTray* kann für jede Messstelle der Boden, auf dem die Messstelle liegt, angegeben werden.

Parameter

Folgende Parameter stehen zur Verfügung:

Parametername	Beschreibung	Einheit	Standardwert
Diameter	Durchmesser Kolonne	m	2
n_Tray	Anzahl der Böden	–	10
Tray_Height	Abstand zwischen den Böden	mm	50
Feed_tray	Zulaufboden	–	10
NbrOfMeasuremnsts	Anzahl der Messstellen an der Kolonne	–	1
MeasurementTray	Boden auf dem sich die Messstelle befindet	–	10
OP_mflF	Zulaufmassenstrom am Arbeitspunkt	kg/s	6
OP_xF	Zulaufkonzentration am Arbeitspunkt	kg/kg	0.4
OP_xD	Destillatkonzentration am Arbeitspunkt	kg/kg	0.9
OP_xB	Sumpfproduktkonzent-ration am Arbeitspunkt	kg/kg	0.1
OP_mflR	Rücklaufmassenstrom am Arbeitspunkt	kg/s	4.5
geoHeight_Colum	Geodätische Höhe der Kolonne (am Sumpf)	m	0
H_Reb	Länge des Reboilers	mm	2000
ThermoSiphon	Schalter für Naturum-lauf	–	True

Parametername	Beschreibung	Einheit	Standardwert
OP_TF	Zulauftemperatur am Arbeitspunkt	°C	60
OP_TR	Temperatur Rücklauf am Arbeitspunkt	°C	60
OP_xR	Konzentration Rücklauf am Arbeitspunkt	kg/kg	0.9
OP_mflCircForced	Umlaufmenge Zwangsumlauf am Arbeitspunkt	kg/s	20

Zusatzparameter

Folgende Zusatzparameter stehen zur Verfügung:

Parametername	Beschreibung	Einheit	Standardwert
T_LB	Siedetemperatur Leichtsieder bei Kolonnendruck	°C	65
T_HB	Siedetemperatur Schwersieder bei Kolonnendruck	°C	100
hV_LB	Verdampfungsenthalpie Leichtsieder	kJ/kg/K	1100
hV_HB	Verdampfungsenthalpie Schwersieder	kJ/kg/K	2200
S	Spreizung der Siedelinse	–	10
pInit_Top	Initialisierung Druck Kolonnenkopf	bar	1
pInit_Bottom	Initialisierung Druck Kolonnensumpf	bar	1.1
LiqLevTray	Flüssigkeitshöhe auf Kolonnenboden	mm	50
Split_Max	Maximaler Trennfaktor	–	0.9999
InitLevelBottom	Initialisierung Füllstand Kolonnensumpf	%	50
kvsLiq	Widerstandsbeiwert Flüssige Anschlüsse	m³/h	1000
kvsVap	Widerstandsbeiwert Gasförmige Anschlüsse	m³/h	100000
kv0	Widerstandsbeiwert Anschlüsse entgegen der beabsichtigten Strömungsrichtung (Rückschlagklappe)	m³/h	0.000001

Bedienfenster

Im Bedienfenster werden die wichtigsten Kenngrößen der Kolonne angezeigt. Es werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Kopfdruck	pKopf	bar
Kopftemperatur	TKopf	°C
Rücklaufmassenstrom	mflR	kg/s
Konzentration Rücklauf	xR	kg/kg
Kopfproduktstrom	mflD	kg/s
Konzentration Kopfprodukt	xD	kg/kg
Rücklaufverhältnis	RR	–
Zulaufmassenstrom	mflF	kg/s
Zulaufkonzentration	xF	kg/kg
Rückdampfverhältnis	BR	–
Rückdampfmassenstrom	mflV	kg/s
Konzentration Rückdampf	xV	kg/kg
Sumpftemperatur	T_Sumpf	°C
Sumpfdruck	p_sumpf	bar
Sumpfproduktstrom	mflB	kg/s
Konzentration Sumpfprodukt	xB	kg/kg
Druckverlust Kolonne	DeltaP	mbar

Filter

Symbol

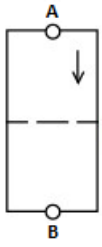


Bild 8-53 Symbol des Komponententyps Filter

Funktion

Die Komponente *Filter* bildet das Verhalten eines Filters in einer Rohrleitung in Form eines Druckabfalls nach.

Dieser Druckabfall ist abhängig von:

- Durchflusskoeffizient des unverschmutzten Filters (k_{VS}-Wert)
- Durchströmung (Volumenstrom, Medium)
- Verschmutzung des Filters (*Clogging*)

Der Druckabfall über dem Filter wird gemäß folgender Formel berechnet:

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2\ (1-clogging)^2\ \rho\frac{\text{kg}}{\text{m}^3}}\ 12960\left(\frac{\text{sec}}{\text{h}}\right)^2$$

Dabei ist

$\Delta p = p_B - p_A$	der Druckabfall über dem Filter in bar
$\dot{m}$	der Durchfluss bzw. Massenstrom in kg/s
ρ	die Dichte des Mediums in kg/m³
k_{VS}	der Durchflusskoeffizient des Filters in m³/h
clogging	der Verschmutzungsgrad des Filters [0..1]

Für den Medienfluss $\dot{m}$ ist die Bezugsrichtung vom Anschluss *A* zum Anschluss *B* definiert, d. h. für einen Medienfluss in Bezugsrichtung gilt $\dot{m} > 0$. Der Druckabfall hat dann somit einen negativen Wert: $\Delta p < 0$.

Dies wird auch durch einen Pfeil im Symbol dargestellt.

Über den Parameter *CloggingRate* können Sie eine Verschmutzungsgeschwindigkeit parametrieren. Dieser Parameter gibt die Verschmutzungsgeschwindigkeit in Prozent je durchströmter Masse an. Somit können Sie einen Filter modellieren, dessen Verschmutzungsgrad und damit dessen Druckverlust mit einer bestimmten Geschwindigkeit zunehmen.

Vereinfachend wird vorausgesetzt, dass zu jedem Zeitpunkt der einströmendende Massenstrom dem ausströmenden Massenstrom entspricht. Der aus dem Hauptstrom herausgefilterte Teilmassenstrom wird somit dem Hauptmassenstrom weder entnommen noch eingespeichert oder abgeführt.

Über den Button "Clogging Reset" ist ein Reinigen des Filters möglich. Diese Funktion kann auch von außen über den Eingang *Reset* verschaltet werden (z. B. nach Ablauf einer Sequenz zum Filterspülen).

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Durchflusskoeffizient des unverschmutzten Filters k_{VS} mit $k_{VS} \geq 10^{-6}$ m³/h	m³/h	10.0
CloggingRate	Verschmutzungsgradänderung je durchströmter Menge (angegeben in kg) in %	%/kg	0.01

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckverlust	Δp	bar
Massenstrom	$\dot{m}$	kg/s
Verschmutzung[1]	–	%

[1] Mit "Reset" zurücksetzbar

PressureStrainer – Drucknutsche

Symbol

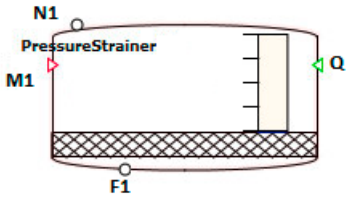


Bild 8-54 Symbol des Komponententyps PressureStrainer

Funktion

Der Komponententyp *PressureStrainer* bildet die Funktionalität einer Drucknutsche nach.

Die Drucknutsche ist ein Filter, in dem eine Suspension mechanisch getrennt wird. Hierzu wird die Drucknutsche mit einer Suspension befüllt. Danach wird die Drucknutsche mit einem Überdruck beaufschlagt, der das Filtrat durch einen Filterboden presst und damit die Trennung vollzieht.

Zwischen Befüllen der Nutsche und Druckbeaufschlagung können noch weitere Schritte erfolgen, wie z. B.

- Sedimentation des Feststoffs am Boden
- Glätten der am Boden durch Sedimentation entstandenen Schicht mit Hilfe eines Rührwerks

Die Komponente Drucknutsche wird als ideal durchmischt modelliert. Damit lassen sich die oben genannten Schritte nicht abbilden. Entsprechende Vorgänge können aber in der Aktor-Sensor-Ebene simuliert werden, wie folgende:

- Wartezeiten des Rührwerks
- Aktivitäten des Rührwerks (Linkslauf / Rechtslauf, Geschwindigkeiten, Rührzeiten usw.)

Effekte dieser Vorgänge sind in der Regel in der Drucknutsche nicht direkt messbar.

Wenn Sie einzelne Effekte aus diesen Aktionen benötigen, so müssen Sie diese zusätzlich über Logik-Bausteine modellieren.

Durch die Druckbeaufschlagung der Nutsche wird das Filtrat langsam durch den Filterboden gepresst. Sobald eine parametrierbare Trockenheit *DryingRate* im Filterkuchen erreicht ist, erfolgt ein "Durchschlagen" des Gases. Dabei entweicht das Gas durch Trocknungsrisse im Filterkuchen aus dem Nutschenraum. Der damit verbundene Druckabfall stellt für die Komponente das Ende der Filtration dar. Eine weiteres Trocknen ist nun nur noch über eine Beheizung der Drucknutsche möglich.

Der Austrag des Filterkuchens erfolgt als flüssiges Medium. Entsprechende Geräte zum Austrag (Schaber, Förderschnecken o. ä.) müssen somit im Modell durch Ventile und Pumpen ersetzt werden.

Zum Trocknen bietet der Komponententyp *PressureStrainer* die Möglichkeit, einen Heiz- bzw. Kühlmantel (Komponente *HeatingJacket*, *ElectricalHeatingJacket*) anzuschließen. Ein Wärmeeintrag, z. B. über einen Rührer, kann über den Eingang *Q* erfolgen.

Innerhalb des Komponententyps *PressureStrainer* erfolgt die Trennung vereinfacht durch Entnahme der verschiedenen Fraktionen (Filtrat und Filterkuchen) aus Teilmassenbilanzen. Der Aufbau des Filterkuchens und der Filterboden selbst werden nicht modelliert.

Beschaltung

Die Komponente *PressureStrainer* besitzt zwei Typen von Anschlüssen für Stoffströme und somit zur Verschaltung von Rohrleitungen:

- **N1 ... Nn – allgemeine Anschlussstutzen**
  Diese Anschlusspunkte dienen zum Befüllen und Entleeren der Drucknutsche und als Gasanschlüsse.
  Alle Anschlüsse des Vektors *N1..Nn* sind zunächst gleichwertig und können für alle Anschlusstypen verwendet werden (Wasser/Dampf, ideales Gas, Liquid)
- **F1 … Fn – Filtratablauf**
  Diese Anschlusspunkte sind ein Sonderfall. Über diese Anschlüsse erfolgt ausschließlich der Austritt des abgeschiedenen Filtrats. Alle anderen Anschlüsse erfolgen über den bereits beschriebenen Vektor *N1..Nn*.

Die Anzahl der Anschlusspunkte hängt vom zu simulierenden System ab und wird über die Parameter *NbrOfConnectors* für *N1..Nn* und *NbrOfFiltrateConnectors* für *F1..Fn* vorgegeben.

Das Stoffmedium in den angrenzenden Flussnetzen wird über die Komponente *NetParam* definiert.

Die Mindestverschaltung ist:

- Suspensionseintritt (*Nn*)
- Filterkuchenaustritt (Gemischanteil mit niederer Dichte) (*Nn*)
- Filtrataustritt (Gemischanteil mit höherer Dichte) (*Fn*)
- Gasanschluss (*Nn*)

Zusätzliche Verschaltungen können beispielsweise sein:

- Ein-/Austritt für Reinigungsmedien (*Nn*)
- Ein-/Austritt für Waschlösungen (*Nn*)
- Zusätzliche Gasein- und Gasaustritte (*Nn*)

Beim Anschlusstyp Wasser/Dampf wird das einströmende Medium intern in der Drucknutsche mit einem Liquid-Medium vermischt. Die einströmende Masse und Enthalpie wird berücksichtigt, es bildet sich im Modell jedoch kein einsprechendes Phasengleichgewicht aus. Einströmender Dampf wird in der Massenbilanz der Flüssigphase zugerechnet und trägt in der Gasphase nicht zum Druckaufbau bei. Wasser wird im Gemisch wie ein Liquid-Medium behandelt.

Wenn Wasser/Dampf als Austrittsmedium verwendet wird, so wird am entsprechenden Stoffausgang der Zentrifuge die Gemischenthalpie der Flüssigphase vorgegeben. Alle weiteren Berechnungen innerhalb dieses Flussnetzes erfolgen dann unter Nutzung der Wasser-/Dampf-Stoffeigenschaften (z. B. Wärmekapazität).

Aufgrund der fehlenden Dampfeigenschaften ist derzeit somit z. B. eine Dampfsterilisation auch im Stillstand nicht möglich.

Parameter ConnectorHeight

Zu jedem der Anschlusspunkte *N1..Nn* und *F1..Fn* wird die Einbauhöhe des Stutzens an der Drucknutsche benötigt (Parameter *ConnectorHeight* für Anschlüsse *N1..Nn* bzw. *ConnectorHeightF* für den Anschluss *F1..Fn).* Die Angabe ist in Metern. Die Voreinstellung ist jeweils 0 m und somit am Boden der Drucknutsche. Negative Werte sind für die Nachbildung von tiefer liegenden Anschlüssen zulässig.

Eingangsvektor SolidFraction

Der Feststoffanteil der Suspension wird entsprechend am Eintritt der Anschlüsse *N1..Nn* vorgegeben und in der Drucknutsche bilanziert. Hierbei erfolgt die Bilanzierung und Parametrierung der Komponente wie beim *StorageTank*.

Anschlüsse ohne Abscheideanteile (z. B. Waschmittel) werden mit dem Wert "0" (Voreinstellung) belegt. Gasanschlüsse erhalten ebenfalls den Wert "0".

Beispiele:

- Produktsuspension – z. B. 90 %
- Lösungsmittel: 0 %
- Spülmittel: 0 %

Die hier vorgegebenen Werte gelten ausschließlich für einströmende Stoffströme. Alle ausströmenden Stoffströme strömen mit der in der Zentrifuge aktuell gespeicherten Feststoffkonzentration aus. Diese steht am Ausgang *Rate* dem Anwender für weitere Berechnungen zur Verfügung. In diesem Fall wird eine ggf. eingetragene *SolidFraction* nicht ausgewertet.

Eine Ausnahme bildet hierbei wiederum die Filtrataustritte *F1..Fn*. Diese Austritte erfolgen immer als reines Filtrat. Zu den Filtrataustritten *F1..Fn* gibt es somit keine Anschlüsse *SolidFraction*. Wenn diese Anschlüsse als Eingänge benutzt werden, erfolgt die Bilanzierung mit der *SolidFraction* = 0.0 (ohne abscheidbare Bestandteile).

Der Eingangsvektor *SolidFraction* ist als default nicht sichtbar.

Alternativ können Sie den Parameter *SolidFraction* der **befüllten Drucknutsche** nach dem Befüllvorgang auch manuell über das Bedienfenster setzen.

Bilanzierung der Flüssigphase

Alle definierten Eingänge im Eingangsvektor *N1 ..Nn* und *F1..Fn* müssen verschaltet werden. Während der Simulation darf keiner dieser Eingänge offen bleiben, da sonst offene Flussnetze angelegt werden. Dies führt bei der Codegenerierung zu einer Fehlermeldung.

Die Flüssigphase wird bezüglich der ein- und austretenden Massen und Enthalpien bilanziert. Zusätzlich erfolgt eine Massenbilanz des abscheidbaren Gemischbestandteils.

$$\frac{dm_{Liquid}}{dt} = \sum_{i=1}^{n} \dot{m}_{Liquid,i}$$

$$\frac{dm_{Screening}}{dt} = \sum_{i}^{n} \dot{m}_{Screening,i}$$

$$\frac{dh_{Liquid}}{dt} = \frac{1}{m_{Liquid}} \left(\sum_{i}^{n} \dot{m}_{Liquid,i}\, h_{Liquid,i} + \sum \dot{Q} \right)$$

$$T_{Liquid} = \frac{h_{Liquid}}{c_P}$$

Gasbilanzierung

Sobald mindestens ein Gasanschluss angeschlossen ist (charakterisiert durch eine entsprechende *NetParam*-Komponente), wird auch ein entsprechender Gasdruck berechnet. Durch Ein- oder Ausspeichern von Gas und/oder Veränderung des Gasvolumens (z. B. durch Füllen des Tanks) wird der Gasdruck verändert. Der berechnete Gasdruck wirkt auf den entstehenden Filterkuchen und drückt aus diesem das Filtrat bis zu einer parametrierbaren Trockenheit *DryingRate* heraus.

Sobald die im Tank gespeicherte aktuelle Feststoffkonzentration *Rate* der parametrierten Solltrockenheit *DryingRate* entspricht, erfolgt ein Druckdurchschlag durch den Filterkuchen. Der Gasdruck in der Drucknutsche sinkt schlagartig ab. Dieser Druckabfall wird im Symbol durch einen Farbumschlag der kreuzgestrichelten Filterzone nach rot gekennzeichnet.

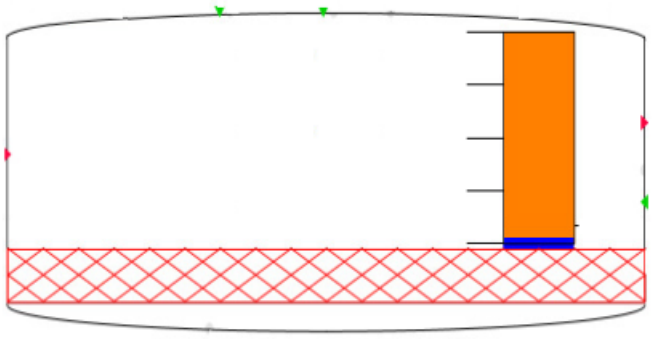


Bild 8-55 Symbol des PressureStrainer bei Erreichen des parametrierten Trocknungsgrades DryingRate (Druckdurchschlag)

Die Gasbilanzierung bezieht sich jedoch ausschließlich auf von außen zugegebene (Inert-)Gase. Dampfdrücke von Lösemitteln werden nicht berücksichtigt.

Innerhalb des Gasraums werden die ein- und austretenden Massen und Enthalpien bilanziert:

$$\frac{dm_{Gas}}{dt} = \sum_{i}^{n} \dot{m}_{Gas,i}$$

$$\frac{dh_{Gas}}{dt} = \frac{1}{m_{Gas}} \left(\sum_{i}^{n} \dot{m}_{Gas,i}\, h_{Gas,i} + \sum \dot{Q} \right)$$

Der Gasdruck wird dann entsprechend der Zustandsgleichung idealer Gase berechnet:

$$p_{Gas} = \frac{m_{Gas}\, R_S\, T_{Gas}}{V_{Gas}}$$

Die Gastemperatur T_{Gas} wird aus der bilanzierten Gasenthalpie h_{Gas} berechnet. Bei unterschiedlicher Gastemperatur und Flüssigtemperatur erfolgt während der Simulation ein entsprechender Temperaturausgleich.

Der absolute Wert des Filtratmassenstroms kann über k_V-Werte im Filtratablauf eingestellt werden (z. B. Ventil oder Nozzle). Hierzu ist eine zusätzliche Komponente notwendig.

Überschreiben von Zustandsgrößen

Über das Bedienfenster der Komponente können Sie während der Simulation folgende Zustandsgrößen auf beliebige (zulässige) Werte setzen:

- Gespeicherte Masse / Füllstand
- Temperatur
- Aktuelle Feststoffkonzentration

Hierzu müssen Sie zunächst den Sollwert in die entsprechende Digitaleingabe *SetValues* eingeben und mit dem Button "Set" übernehmen. Den Zustand der gespeicherten Masse können Sie hierbei sowohl über den Wert "Masse" als auch über den Füllstand (*Level*) vorgeben.

Über diese Beeinflussung von Zustandsgrößen können Sie z. B. Aufwärmvorgänge oder Füllvorgänge beschleunigen. Ebenso können Sie als Bediener die Drucknutsche entleeren, um beispielsweise in einer Schrittkette zurückzuspringen.

Austrag des Feststoffs

Ein Austrag des Feststoffs aus der Drucknutsche über Förderschnecken, Schaber oder ähnliches ist nicht möglich. Dies liegt daran, dass der Feststoffanteil zwar als Feststoff bilanziert werden kann, intern im Modell aber als Flüssigkeit behandelt wird. Dies hat zur Folge, dass für den Austrag entsprechend Ventile und Pumpen benutzt werden müssen, welche durch die Ansteuersignale der eigentlichen Austraggeräte angesteuert werden.

Grenzfall "Drucknutsche leer"

Zuflüsse in die Drucknutsche und Abflüsse aus der Drucknutsche werden generell am Stutzen der Nutsche gedrosselt. Im normalen Betrieb ist diese Drosselung für den Anwender nicht direkt erkennbar, da diese Drosselung nur wenig Einfluss auf die Simulation haben soll. Für eine leere Nutsche wird der Abfluss stark gedrosselt. Der Durchflusskoeffizient wird dazu für alle Anschlüsse, über die ein Medium abfließt, auf den Wert für maximale Drosselung gesetzt.

Diese Durchflusskoeffizienten werden intern abhängig vom Behältervolumen berechnet und können durch den Modellierer nicht beeinflusst werden.

Die Nutsche wird als leer angenommen, wenn die Flüssigkeitsmenge M_W/ρ kleiner als eine gegebene minimale Füllung *RemainderLiquid* wird:

$m_L < RemainderLiquid \cdot \rho$

Die Bilanzierung der Zustände wird für eine leere Nutsche gestoppt, d. h. Änderungen der Masse, Teilmassen und spezifischen Enthalpie werden verworfen. Der Zustand "leer" wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Flüssigkeitsmenge vorliegt, d. h. die Differenz aus einströmender und abfließender Menge positiv ist.

Grenzfall "Drucknutsche voll"

Die Nutsche wird als voll angenommen, wenn ihre Flüssigkeitsmenge den maximal möglichen Wert erreicht hat:

$m_L < (V - RemainderGas) \cdot \rho$

Die Bilanzierung der Zustände wird für eine volle Nutsche gestoppt, d. h. Änderungen der gespeicherten Masse, Teilmasse und spezifischen Enthalpie werden verworfen. Der Zustand "voll" wird erst wieder verlassen, wenn eine genügend große Abnahme der Flüssigkeitsmenge vorliegt, d. h. die Differenz aus einströmender und abfließender Menge negativ ist.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Behälters; online änderbar	m^3	1.0
Height	Höhe des Behälters; online änderbar	m	1.0
NbrOfInletsOutlets	Anzahl *N* der Anschlüsse allgemein	–	3
NbrOfFiltrateOutlets	Anzahl *N* der Filtratanschlüsse	–	1
NbrOfMeasurements	Anzahl *N* der Messstellen	–	1
MeasurementHeight [Nr]	Höhe der Messstellen über dem Behälterboden	m	0.0
ConnectorHeight [Nr]	Höhe der allgemeinen Rohranschlüsse über dem Behälterboden	m	0.0
ConnectorHeightF [Nr]	Höhe der Filtratanschlüsse über dem Behälterboden	m	0.0

Parametername	Beschreibung	Einheit	Standardwert
Boiling	Dampfdruckkurve des Leichtsieders: Druck in bar als Funktion der Temperatur (°C)	–	
Vaporization	Verdampfungsenthalpie des Leichtsieders in kJ/kg als Funktion der Temperatur (°C)	–	

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureOutside	Umgebungsdruck	bar	1.0
LevelInit	Startfüllstand	%	3.0
TemperatureInit	Starttemperatur	°C	20.0
PressureInit	Startdruck	bar	1.0
InitSolidPart	Initialmassenanteil des Feststoffs in der Suspension	%	0.0
Density	Gemischdichte im Behälter	kg/m^3	997.337
DryingRate	Maximal erreichbare Feststoffkonzentration im Filterkuchen, nach welcher der Druckdurchschlag erfolgt	kg/kg	0.95
RemainderGas	Restvolumen, das beim Füllen der Nutsche nicht mit Flüssigkeit gefüllt werden kann	m^3	0.05
RemainderLiquid	Restvolumen, das beim Entleeren der Nutsche in der Zentrifuge verbleibt	m^3	0.005
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Länge	L	m	Ja[1]
Temperatur (Flüssigkeit)	T_L	°C	Ja[1]
Temperatur (Gas)	T_G	°C	Ja[1]
Gewicht	M	kg	Ja[1]
Druck	p_B	bar	Nein
Gasdruck	p_G	bar	Nein
Anteil Feststoff in der Suspension	w	%	Ja[1]

[1] Eingabe mit "Set" aktivieren

Beispielverschaltung

In der folgenden Abbildung ist beispielhaft eine Verschaltung einer Drucknutsche unter Nutzung von Komponenten der Bibliothek CHEM-BASIC dargestellt.

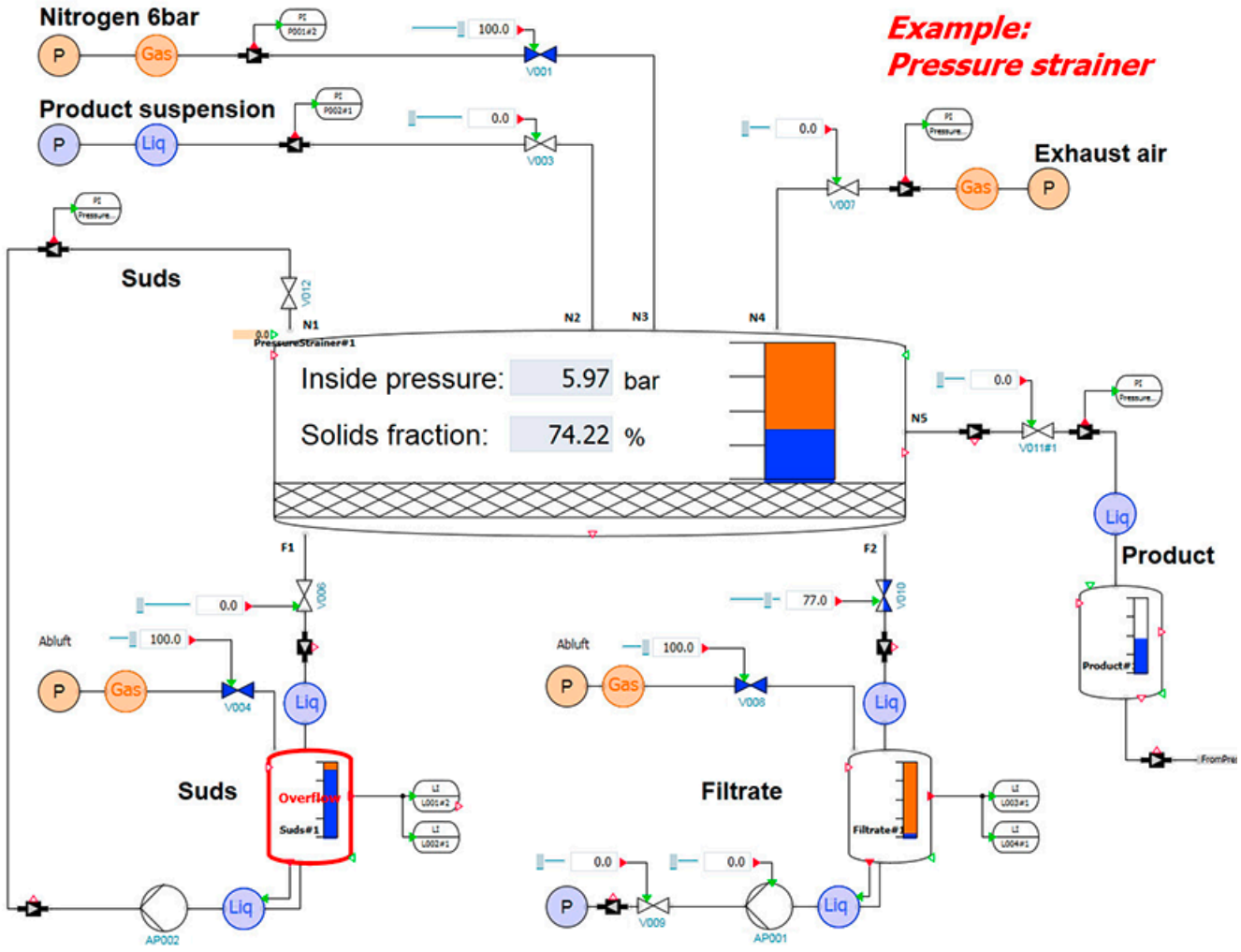


Bild 8-56 Beispielverschaltung des Komponententyps PressureStrainer

Für diese Beispielverschaltung wird die folgende Parametrierung der Komponente *PressureStrainer* verwendet:

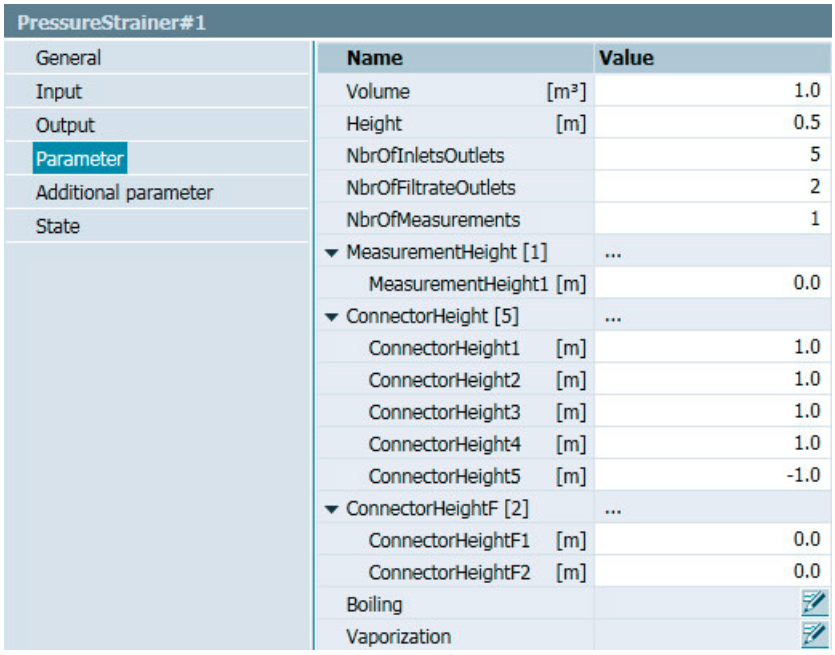


Bild 8-57 Parametrierung der Komponente PressureStrainer zur Beispielverschaltung

Siehe auch

Geodätische Höhe (Seite 612)

ScreeningDevice – Siebeinrichtung

Symbol

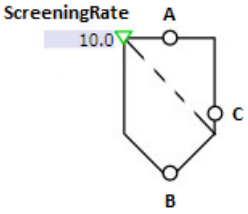


Bild 8-58 Symbol des Komponententyps ScreeningDevice

Funktion

Der Komponententyp *ScreeningDevice* dient zum Abtrennen eines Teilmassenstroms aus einem Gesamtmassenstrom.

Beispiel:

- Abtrennung einer bestimmten Korngröße aus einem Gesamtgemenge
- Abtrennung eines Feststoffanteils aus einer Suspension

Diese Komponente verfügt über eine Siebstufe.

Der abtrennbare Anteil des Gesamtmassenstroms wird über den Eingang *RateA* vorgegeben.

Das Flussnetz des abgetrennten Teilmassenstroms *C* ist topologisch nicht mit dem Flussnetz des Hauptmassenstroms *AB* verbunden. Es muss daher über einen eigenen Baustein *NetParam* beschrieben werden.

Eine Abtrennung kann nur erfolgen, wenn das an den Anschluss *C* angeschlossene Flussnetz geöffnet ist.

Im Betrieb muss immer mindestens der abtrennbare Anteil aus dem Gesamtmassenstrom abgetrennt werden. Falls dieser Anteil über das an den Anschluss *C* angeschlossene Flussnetz nicht abgeführt werden kann, wird der Hauptstrom reduziert. Das Sieb wird blockiert.

Ebenso ist ein entsprechender Durchflusskoeffizient des Siebs (k_{VS}-Wert) für den Hauptmassenstrom notwendig. Wenn dieser Durchflusskoeffizient zu klein ist, kann gegebenenfalls der geforderte Massenstrom nicht erreicht werden.

Der Ausgang *Rate* zeigt den Anteil des abtrennbaren Gemischbestandteils am Ausgang *B* an. In der Komponente *Separator* wird von einer vollständigen Trennung ausgegangen, so dass dieser Anteil immer bei 0 % liegt.

Eine Rückströmung über den Anschluss *C* erfolgt nicht. Ebenso ist eine Rückströmung von *B* nach *A* nicht möglich.

Für den Hauptmassenstrom wird in der Komponente ein Druckabfall berechnet. Dieser Druckabfall ist abhängig von:

- Durchflusskoeffizient des Siebs (k_{VS}-Wert)
- Durchströmung (Volumenstrom, Medium)

Der Druckabfall über dem Sieb wird gemäß folgender Formel berechnet:

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \; \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Dabei ist

$\Delta p = p_B - p_A$	der Druckabfall über dem Sieb in bar
$\dot{m}$	der Durchfluss bzw. Massenstrom in kg/s
ρ	die Dichte des Mediums in kg/m³
k_{VS}	der Durchflusskoeffizient des Siebs in m³/h

ACHTUNG

Das Flussnetz des ausgetragenen Massenstroms am Anschluss C beginnt in der Komponente mit einer Massenstromquelle.

Aufgrund dieser Massenstromquelle ist der sich am Rand des Flussnetzes einstellende Druck abhängig vom abgeschiedenen Massenstrom und dem Druckabfall im folgenden Zweig. Zu hohe Druckabfälle im folgenden Zweig (d. h. beispielsweise zu geringe k_{VS}-Werte in den Ventilen) können zu physikalisch nicht möglichen Zuständen führen (z. B. Druck im Austrag ist höher als im Hauptstrom).

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Durchflusskoeffizient k_{VS} mit $k_{VS} \geq 10^{-6}$ m^3/h	m^3/h	10.0
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen für folgende Massenströme angezeigt:

- Eintretender Gesamtmassenstrom
- Austretender Gesamtmassenstrom
- Abgetrennter Teilmassenstrom

Größe	Formelzeichen	Einheit
Massenanteil	w	%
Massenstrom	$\dot{m}$	kg/s

Siehe auch

Geodätische Höhe (Seite 612)

Separator – Abscheider

Symbol

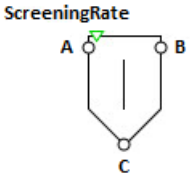


Bild 8-59 Symbol des Komponententyps Separator

Funktion

Der Separator dient zur teilweisen oder vollständigen Abtrennung eines Teilmassenstroms *C* (z. B. eines Feststoffmassenanteils) aus dem Hauptstrom *AB*.

Da in SIMIT das Flussnetzwerk prinzipiell nur den Transport von Gasen und Flüssigkeiten zulässt, werden Feststoffe in SIMIT wie Flüssigkeiten behandelt. Somit ist für einen Stofftransport ein Druckunterschied notwendig. Fördereinrichtungen werden durch Ersatzverschaltungen ersetzt, die aus Pumpen, Ventilen, Quellen und Senken bestehen.

Das Flussnetz des abgetrennten Teilmassenstroms *C* ist topologisch nicht mit dem Flussnetz des Hauptmassenstroms *AB* verbunden. Das Flussnetz des Teilmassenstroms muss daher über einen eigenen Baustein *NetParam* beschrieben werden.

Eine Abtrennung kann nur erfolgen, wenn das an den Anschluss *C* angeschlossene Flussnetz geöffnet ist. Wenn dieses Flussnetz z. B. durch ein geschlossenes Ventil blockiert ist, so erfolgt durch diesen Separator keine Abtrennung. In diesem Fall wird der entsprechende Massenstrom an den Ausgang *B* weitergeleitet.

Ebenso ist ein entsprechender Druckabfall vom Hauptmassenstrom zum Anschlusspunkt *C* notwendig.

Eine Rückströmung über den Anschluss *C* ist nicht möglich.

Über die Eingänge *RateA* oder *RateB* wird je nach Strömungsrichtung der Massenanteil des abzuscheidenden Teilmassenstroms vorgegeben.

In der Komponente *Separator* ist ein Knoten enthalten. Über die Dynamik dieses Knotens können Sie ein Speichervolumen nachbilden. Knoteneinstellungen müssen Sie über die entsprechende Netzparametrierungskomponente (*NetParam*) separat für Hauptstromnetz und Teilstromnetz vornehmen.

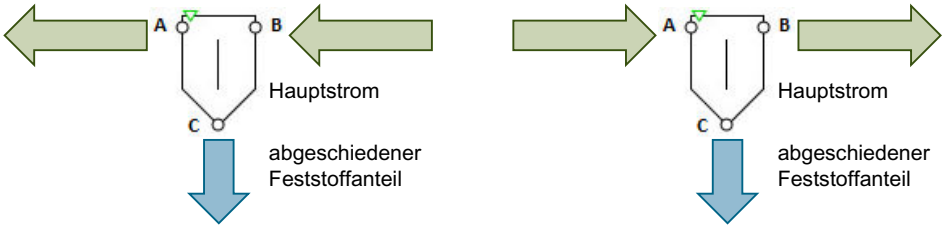


Bild 8-60 Strömungsmöglichkeiten des Komponententyps Separator

Die Hauptstromanschlüsse *A* und *B* sind gleichwertig ausgelegt. Der abzuscheidende Teilmassenstrom wird immer über den Anschluss *C* ausgetragen.

Der Massenanteil des an *A* oder *B* austretenden Hauptstroms wird berechnet und am Ausgang *Rate* angezeigt.

Für den Hauptmassenstrom wird ein Druckabfall berechnet. Dieser Druckabfall ist abhängig von:

- Druckverlustwert des Separators (k_{VS}-Wert)
- Durchströmung (Volumenstrom, Medium)

Der Druckabfall über dem Separator wird gemäß folgender Formel berechnet.

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \; \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Dabei ist

$\Delta p = p_B - p_A$	der Druckabfall über dem Separator in bar
$\dot{m}$	der Durchfluss bzw. Massenstrom in kg/s
ρ	die Dichte des Mediums in kg/m³
k_{VS}	der Durchflusskoeffizient des Separators in m³/h

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Durchflusskoeffizient k_{VS} mit $k_{VS} \geq 10^{-6}$ m^3/h	m^3/h	10.0
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen für folgende Massenströme angezeigt:

- Eintretender Hauptmassenstrom
- Austretender Hauptmassenstrom
- Abgetrennter Teilmassenstrom

Größe	Formelzeichen	Einheit
Massenanteil	w	%
Massenstrom	$\dot{m}$	kg/s

Siehe auch

Geodätische Höhe (Seite 612)

8.2.5.9 System

BranchParam – Zweigparametrierung

Symbol



Bild 8-61 Symbol des Komponententyps BranchParam

Funktion

Der Komponententyp *BranchParam* dient zur Parametrierung eines Zweigs im Flussnetz. Dazu wird die Komponente an beliebiger Stelle in den zu parametrierenden Zweig eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0

Mnode – Massenstromvorgabe

Symbol

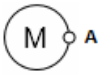


Bild 8-62 Symbol des Komponententyps Mnode

Funktion

Der Komponententyp *Mnode* gibt Werte für Massenstrom $\dot{m}$ und spezifische Enthalpie h an seinem Anschluss A vor. Eine Komponente dieses Typs bildet damit einen Rand für das Flussnetz. In der Betrachtung des Flussnetzes als Graph entspricht das einer Ein- oder Ausströmung in einen (internen) Knoten oder Zweig. Dazu wird für jede Komponente dieses Typs ein interner Knoten mit der definierten Ein- oder Ausströmung in das Flussnetz eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Massenstrom (Vorbelegung: 0)	$\dot{m}$	kg/s	Ja
spezifische Enthalpie (Vorbelegung: 100)	h	kJ/kg	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

NetParam – Netzparametrierung

Symbol



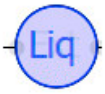
Bild 8-63 Symbol des Komponententyps NetParam

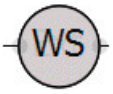
Funktion

Der Komponententyp *NetParam* dient zur Parametrierung eines Flussnetzes. Dazu wird die Komponente an beliebiger Stelle in einen beliebigen Zweig des Flussnetzes eingefügt.

Darstellung der Komponente *NetParam* in Abhängigkeit vom eingestellten Medium:







Medium	Gas	Liquid	Water/Steam

Die in dieser Komponente vorgegebenen Parameter gelten flussnetzweit. Sie können von anderen im Flussnetz enthaltenen Komponenten gelesen und weiterverarbeitet werden.

Die Komponente *NetParam* hat folgende Eingänge:

- *MediumExt*
  Externe Umschaltung des im Flussnetz eingestellten Mediums zur Laufzeit durch Ereignisse aus dem Modell. Definierte Eingänge sind:
  - −9999 (Voreinstellung): Dieser Eingang wird nicht verwendet; das im Parameter Medium eingestellte Medium ist aktuell
  - 0: Umschaltung auf Medium "Water/Steam"
  - 1: Umschaltung auf Medium "Ideal Gas"
  - 2: Umschaltung auf Medium "Liquid"
- *CpExt*
  Externer Eingang für die Wärmekapazität im Flussnetz (kann z. B. vom Ausgang aus einem vorgeschalteten Tank verschaltet werden). Als Eingang werden nur Werte größer null akzeptiert, bei Eingang null (Voreinstellung) gelten die Parameter *sHeatCapGas* bzw. *sHeatCapLiquid*.

Beide Eingänge haben die Voreinstellung "nicht sichtbar" und müssen zur Verwendung zunächst sichtbar geschaltet werden.

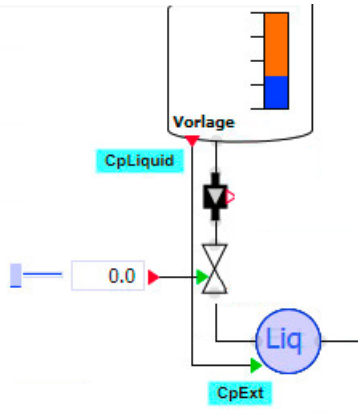


Bild 8-64 Beispiel für die Verschaltung des Eingangs CpExt

ACHTUNG
Die Komponente ermöglicht das Umschalten des Mediums während der Laufzeit. Schalten Sie jedoch nur in Ausnahmefällen um, weil das Modell hierdurch je nach Verschaltung der Komponenten in einen undefinierten Zustand kommen kann. Modellabstürze oder das Erreichen von nicht definierten Zuständen sind u. U. möglich.

Wenn in einem Flussnetz keine Komponente zum Parametrieren eines Flussnetzes verwendet wird, wird mit den Voreinstellungen für Wasser/Dampf gerechnet.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Medium	Sie können als Medium im Flussnetz "Water/Steam" für Wasser/Dampf, "Liquid" für Flüssigkeit oder "Ideal Gas" für ideales Gas auswählen.	–	Liquid
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0
sCompressionGas	Spezifischer Kompressionsmodul für Medium "Water/Steam" mit Dichten ρ < 500 kg/m³ oder Medium "Ideal Gas"	bar/kg	10.0
sCompressionLiquid	Spezifischer Kompressionsmodul für Medium "Water/Steam" mit Dichten ρ > 500 kg/m³ oder Medium "Liquid"	bar/kg	100.0
FactorThermalGas	Faktor für die Enthalpiebilanzierung Medium "Water/Steam" mit Dichten ρ < 500 kg/m³ oder Medium "Ideal Gas"	1/kg	100.0
FactorThermalLiquid	Faktor für die Enthalpiebilanzierung Medium "Water/Steam" mit Dichten ρ > 500 kg/m³ oder Medium "Liquid"	1/kg	0.1
DensityLiquid	Mediendichte; nur gültig für Medium "Liquid"	kg/m³	997.337
sHeatCapGas	Spezifische Wärmekapazität für Gas; nur gültig für Medium "Ideal Gas"	kJ/kgK	1.0
sHeatCapLiquid	Spezifische Wärmekapazität für Flüssigkeiten; nur gültig für Medium "Liquid"	kJ/kgK	4.18
GasConstant	Spezifische Gaskonstante des Mediums; nur gültig für Medium "Ideal Gas"	kJ/kgK	0.287

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck in den internen Knoten des Flussnetzes	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie in den internen Knoten des Flussnetzes	kJ/kg	100.0
SmoothTransition	Wenn dieser Zusatzparameter auf True gesetzt ist, werden die Größen *sCompression* und *FactorThermal* mit einer dichteabhängigen linearen Übergangsfunktion angesetzt	–	False
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums in den internen Knoten mit der Umgebung; bei einem Wert null erfolgt kein Wärmeaustausch	kW/kgK	0.0

PHnode – Druck- und Enthalpievorgabe

Symbol



Bild 8-65 Symbol des Komponententyps PHnode

Funktion

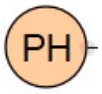
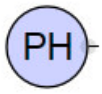
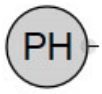
Der Komponententyp *PHnode* stellt eine Möglichkeit des Abschlusses eines Flussnetzes nach außen dar. In der Komponente werden ein fester Druck *p* und die zugehörige Enthalpie *h* als Randbedingung für das Flussnetz vorgegeben.

In der Betrachtung des Flussnetzes als Graph entspricht diese Komponente einem (externen) Knoten, für den die eingegebenen Bedingungen fest vorgegeben sind.

Darstellung

Während der Simulation wird im Symbol der Komponente das ausgewählte Medium farbig dargestellt. Die hierfür notwendige Information bekommt die Komponente aus der zugehörigen Flussnetzparametrierung (Komponente *NetParam*).

Darstellung der Komponente *PHnode* in Abhängigkeit vom eingestellten Medium:

Medium	Gas	Liquid	Water/Steam
			

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Druck (Vorbelegung: 1)	p	bar	Ja
spezifische Enthalpie (Vorbelegung: 100)	h	kJ/kg	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom[1]	$\dot{m}$	kg/s
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

[1] Für Ausströmung ist $\dot{m} > 0$, für Einströmung ist $\dot{m} < 0$.

PTnode – Druck- und Temperaturvorgabe

Symbol



Bild 8-66 Symbol des Komponententyps PTnode

Funktion

Der Komponententyp *PTnode* stellt eine Möglichkeit des Abschlusses eines Flussnetzes nach außen dar. In der Komponente werden ein fester Druck *p* und die zugehörige Temperatur *T* als Randbedingung für das Flussnetz vorgegeben.

In der Betrachtung des Flussnetzes als Graph entspricht diese Komponente einem (externen) Knoten, für den die eingegebenen Bedingungen fest vorgegeben sind.

Hinweis

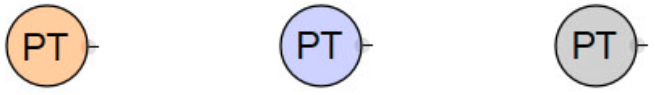
Beim Medium "Water/Steam" müssen Sie beachten, dass die Eingabe eines Zustandes im Zwei-Phasen-Gebiet nicht möglich ist. In Abhängigkeit von Temperatur und Druck ist der Zustand entweder vollständig flüssig oder vollständig dampfförmig. Wird die Siedetemperatur bei dem entsprechenden Duck eingegeben, ist der Zustand vollständig gasförmig.

Wenn ein Zustand im Zwei-Phasen-Gebiet festgelegt werden soll, muss der Komponententyp *PHNode* verwendet werden.

Darstellung

Während der Simulation wird im Symbol der Komponente das ausgewählte Medium farbig dargestellt. Die hierfür notwendige Information bekommt die Komponente aus der zugehörigen Flussnetzparametrierung (Komponente *NetParam*).

Darstellung der Komponente *PTnode* in Abhängigkeit vom eingestellten Medium:



Medium Gas Liquid Water/Steam

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

8.2.5.10 Tanks

Agitator – Rührer

Symbol



Bild 8-67 Symbol des Komponententyps Agitator

Funktion

Der Komponententyp *Agitator* hat die Funktion, das Vorhandensein und den Betrieb eines Rührwerks darzustellen. Das Rührwerk wird über den Anschluss *Speed* eingeschaltet. Bei einer Drehzahl *Speed* > 0 wird die Drehung des Rührwerks grafisch dargestellt.

Auf die Durchmischung in einem Rührkessel hat diese Komponente jedoch keinen Einfluss, da bei Rührkesseln immer von ideal durchmischten Systemen ausgegangen wird.

Auch auf den Wärmeübergang von einem Heizelement in einen Rührkessel hat diese Komponente keinen Einfluss. Zur Modellierung des Wärmeübergangs müssen Sie die Rührerdrehzahl mit der Komponente *HeatingJacket* verknüpfen.

DrumWS – Speichertank für Wasser/Dampf

Symbol

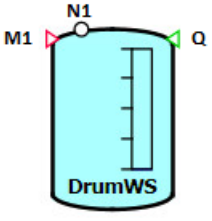


Bild 8-68 Symbol des Komponententyps DrumWS

Funktion

Der Komponententyp *DrumWS* stellt die Simulation eines gegen die Umgebung abgeschlossenen Behälters für Wasser/Dampf zur Verfügung.

Der geschlossene Behälter wird als liegender oder stehender zylindrischer Behälter angenommen, dessen Zu- und Abflüsse über Anschlüsse *N1* .. *Nn* erfolgen. Für jeden Anschluss wird eine Drosselwirkung angenommen gemäß der Formel:

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Die Einbauhöhe der einzelnen Anschlüsse wird über den Parametervektor *ConnectorHeight* jeweils einzeln vorgegeben. Die Art jedes Anschlusses (Eintritt oder Austritt, Wasser oder Dampf) ergibt sich dynamisch während der Simulation und hängt von den thermodynamischen Zuständen im Tank, der parametrierten Geometrie (*ConnectorHeight*, *Orientation*) und den jeweils anliegenden Randbedingungen ab.

Sie können die Anschlüsse bei gedrückter <Alt>-Taste mit der Maus beliebig auf dem Umriss des Komponentensymbols verschieben.

Im Behälter wird eine unmittelbare Trennung des Mediums in zwei homogene gesättigte Phasen angenommen: die gesättigte flüssige Phase und die gesättigte dampfförmige Phase. Es werden ausschließlich Anschlüsse an Flussnetze des Typs Wasser/Dampf bilanziert. Anschlüsse an andere Flussnetztypen (Gas, Liquid oder benutzerspezifische Medien) werden nicht bilanziert.

Man erhält so die über die Massenströme bilanzierte Masse von Wasser und Dampf im Behälter gemäß

$$\frac{dm}{dt} = \sum_{i=1}^{n} \dot{m}_i$$

bzw. wegen

$m = \rho V$

die Änderung der mittleren Dichte zu

$$V \frac{d\rho}{dt} = \sum_{i=1}^{n} \dot{m}_i$$

Weiter wird die Energie bilanziert gemäß

$$\frac{d(hm)}{dt} = \sum_{i=1}^{n} h_i \dot{m}_i$$

Dabei ist h die spezifische Enthalpie (mittlere Enthalpie) von Wasser/Dampf im Behälter und h_i die spezifische Enthalpie des Zu- bzw. Abflusses am i-ten Anschluss. Für Abflüsse ist die spezifische Enthalpie des Mediums im Behälter anzusetzen, also h' für Abflüsse von gesättigtem Wasser und h'' für Abflüsse von gesättigtem Dampf. Die Bilanzierung ergibt sich damit gemäß

$$m \frac{dh}{dt} = \sum_{i=1, i \in Z}^{n} (h_i - h) \dot{m}_i + (h' - h) \sum_{i=1, i \in A'}^{n} \dot{m}_i + (h'' - h) \sum_{i=1, i \in A''}^{n} \dot{m}_i$$

Dabei wird über die Zuflüsse Z, Abflüsse von gesättigtem Wasser A' bzw. Abflüsse von gesättigtem Dampf A'' summiert.

Um einen Wärmeaustausch mit dem ideal isolierten Behälter zu modellieren, wird die Enthalpiebilanz ergänzt gemäß

$$m\frac{dh}{dt} = \sum_{i=1, i\in Z}^{n}(h_i - h)\dot{m}_i + (h' - h)\sum_{i=1, i\in A'}^{n}\dot{m}_i + (h'' - h)\sum_{i=1, i\in A''}^{n}\dot{m}_i + A\alpha(T_T - T_S)$$

T_T ist die Temperatur der Behälterwand, T_S die Sättigungstemperatur von Wasser/Dampf.

Die Wärmeeinspeicherung in der Behälterwand wird mit der Wärmebilanz beschrieben:

$$\frac{dT_T}{dt} = \frac{1}{m_T c_T} A\alpha(T_S - T_T)$$

m_T Masse der Behälterwand [kg]

c_T Wärmekapazität der Behälterwand [kJ/(kg·K)]

Am Anschluss *Measure* werden das Niveau des Wassers *l*, die Sättigungstemperatur T_S, der Sättigungsdruck p_S und die Masse *m* von Wasser/Dampf ausgegeben.

Die Anschlüsse *N1* .. *Nn* werden entsprechend der parametrierten Höhe angenommen. Für unterhalb des Füllstandes liegende Anschlüsse setzt sich der Druck an diesen Anschlüssen damit aus dem Sättigungsdruck und dem Schweredruck des Wassers in Abhängigkeit von der Differenz aus Anschlusshöhe (l_A) und Füllstand (*l*) zusammen:

$p = \rho' g (l - l_A) + p_S$

l_A Höhe des Anschlussstutzens [m]

Grenzfall "Behälter leer"

Für einen leeren Behälter wird der Abfluss stark gedrosselt. Der Ventilkennwert wird dazu für alle Anschlüsse über die Wasser oder Dampf abfließt, auf den Wert k_{V0} für maximale Drosselung gesetzt.

Der Behälter wird als leer angenommen, wenn die Wassermenge m_W/ρ kleiner als eine gegebene minimale Füllung V_{min} wird:

$m_W < V_{min}\rho$

Die Bilanzierung der Zustände wird für einen leeren Behälter gestoppt, d. h. Änderungen der Dichte und der spezifischen Enthalpie werden verworfen. Der Zustand "leer" wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Wassermenge vorliegt. In jedem Zeitschritt wird dazu geprüft, ob gilt

$$m_W \geq V_{min}\,\rho\left(1 + \frac{H_{min}}{100\%}\right)$$

Der erforderliche Zuwachs kann über die Hysterese H_{min} gesetzt werden.

Im Symbol einer Komponente wird für einen leeren Behälter ein entsprechender Hinweis eingeblendet (siehe Abbildung).

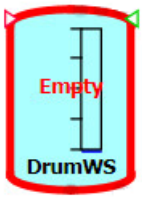


Bild 8-69 Anzeige eines leeren Behälters im Symbol

Grenzfall "Behälter voll"

Der Behälter wird als voll angenommen, wenn seine Wassermenge den maximal möglichen Wert erreicht hat:

$m_W > (V - V_{min})\,\rho$

Die Bilanzierung der Zustände wird für einen vollen Behälter gestoppt, d. h. Änderungen der Dichte und der spezifischen Enthalpie werden verworfen. Der Zustand "voll" wird erst wieder verlassen, wenn eine genügend große Abnahme der Wassermenge vorliegt. In jedem Zeitschritt wird dazu geprüft, ob gilt

$$m_W \leq \rho V - \rho V_{min}\left(1 + \frac{H_{min}}{100\%}\right)$$

Die erforderliche Abnahme kann über die Hysterese H_{min} gesetzt werden.

Im Symbol einer Komponente wird für einen vollen Behälter ein entsprechender Hinweis eingeblendet (siehe Abbildung).

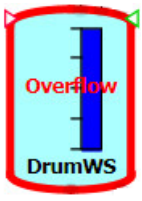


Bild 8-70 Anzeige eines vollen Behälters im Symbol

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Behälters; online änderbar	m^3	1.0
HeightOrLength	Höhe bzw. Länge des Behälters; online änderbar	m	1.0
NbrOfConnectors	Anzahl der Anschlüsse N	–	1
NbrOfMeasurements	Anzahl N der Messstellen	–	1
MeasurementHeight [Nr]	Höhe der Messstellen über dem Behälterboden	m	0.0

Parametername	Beschreibung	Einheit	Standardwert
ConnectorHeight [Nr]	Höhe der Rohranschlüsse über dem Behälterboden	m	0.0
Position	Lage des Behälters: *Vertically* oder *Horizontally* Im Fall der Auswahl *Horizontally* wird für den Füllstand ein liegender Zylinder angenommen	–	*Vertically*

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
LevelInit	Initialisierungswert für den Füllstand (Wasservolumen) in %	m^3	50.0
TemperatureInit	Initialisierungswert für die Temperatur von Wasser/Dampf (Sättigungstemperatur)	m	20.0
Kvs	für alle Anschlüsse einheitlicher Ventilkennwert k_V	m^3/h	360.0
Kv0	Ventilkennwert k_{V0} für starke Drosselung im Behälteranschluss	m^3/h	0.000001
MinVolume	Minimales Volumen V_{min} des Behälters; online änderbar	m^3	0.01
MinVolumeHys	Hysterese H_{min}; online änderbar	%	50.0
MassDrum	Masse m_T des Behälters; online änderbar	kg	5000.0
SurfaceDrum	Innere Oberfläche A des Behälters; online änderbar	m^2	12.5
sHeatCapDrum	Spezifische Wärmekapazität c_T des Behälters; online änderbar	kJ/kgK	0.5
HeatTransCoe	Wärmeübergangskoeffizient α für Wasser/Dampf des Behälters; online änderbar	kW/m^2K	0.0
geoHeight	Geodätische Höhe; online änderbar	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Wasserniveau	L	m	Ja[1]
Sättigungstemperatur	T	°C	Ja[1]
Gewicht von Wasser/ Dampf	M	kg	Ja[1]
Sättigungsdruck	p_B	bar	Nein
Wasserdruck	p_G	bar	Nein
Behältertemperatur	T_{Drum}	°C	Nein

[1] Eingabe mit "Set" aktivieren

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

- Mix: mittlere Größen
- Water: Größen für gesättigtes Wasser
- Steam: Größen für gesättigten Dampf

Größe	Formelzeichen	Einheit
Dichte	r	kg/m³
spezifische Enthalpie	h	kJ/kg

Siehe auch

Geodätische Höhe (Seite 612)

ElectricalHeatingJacket – elektrischer Heizmantel

Symbol

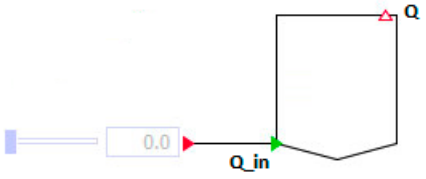


Bild 8-71 Symbol des Komponententyps ElektricalHeatingJacket

Funktion

Der elektrische Heizmantel dient als AddOn zum *StorageTankLiquid*. Die am Eingang Q_In anliegende Heizleistung wird als Wärmestrom Q dem Tank zugeführt. Die Vorgabe des Wärmestroms ist in kW.

Die Komponente dient zur Visualisierung des Heizmantels im SIMIT-Diagramm. Simulationstechnisch kann die Heizleistung auch direkt an einen *StorageTankLiquid* angeschlossen werden.

Im Gegensatz zur Komponente *HeatingJacket* wird hier der Einfluss der Rührerdrehzahl oder des Füllstandes im Tank nicht berücksichtigt.

Diese Komponente besitzt keine Parameter und kein Bedienfenster.

Flash – 2-Phasen-Abscheider

Symbol

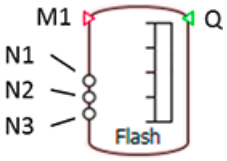


Bild 8-72 Symbol des Komponententyps Flash

Funktion

Die Komponente *Flash* beschreibt einen 2-Phasen-Abscheider und dient zur Aufteilung des Inhalts in eine flüssige und eine gasförmige Phase. In Abhängigkeit von Temperatur und Druck stellt sich dabei das Phasengleichgewicht zwischen dem Leichtsieder (LB = Low Boiling) und dem Schwersieder (HB = High Boiling) ein. Hierbei werden die ein- und austretenden Stoffströme automatisch bezüglich Masse und Enthalpie bilanziert. Eine Temperierung der im Tank gespeicherten Masse ist über einen Direkteingang oder verschiedene Zusatzkomponenten möglich.

Parametriert wird das Phasengleichgewicht über die Normal-Siedetemperatur (bei 1 bar) des Leicht- und Schwersieders (*TS_LB*, *TS_HB*) sowie über deren Verdampfungsenthalpien (*hv_LB*, *hv_HB*). Die Größe der Siedelinse kann über den Parameter *S* variiert werden. Dabei bedeutet:

- $S = 0$, maximale Größe der Siedelinse (siehe Abbildung)
- $S = T_{HB} - T_{LB}$, keine Siedelinse (physikalisch unsinnig)

Es wird angenommen, dass für beide Stoffe eine gleiche Druckabhängigkeit der Siedepunkte besteht. Parametriert wird dies mit der Siedetemperatur des Schwersieders (*T2_HB*) für einen anderen (vorzugsweise höheren) Druck (*p2_HB*).

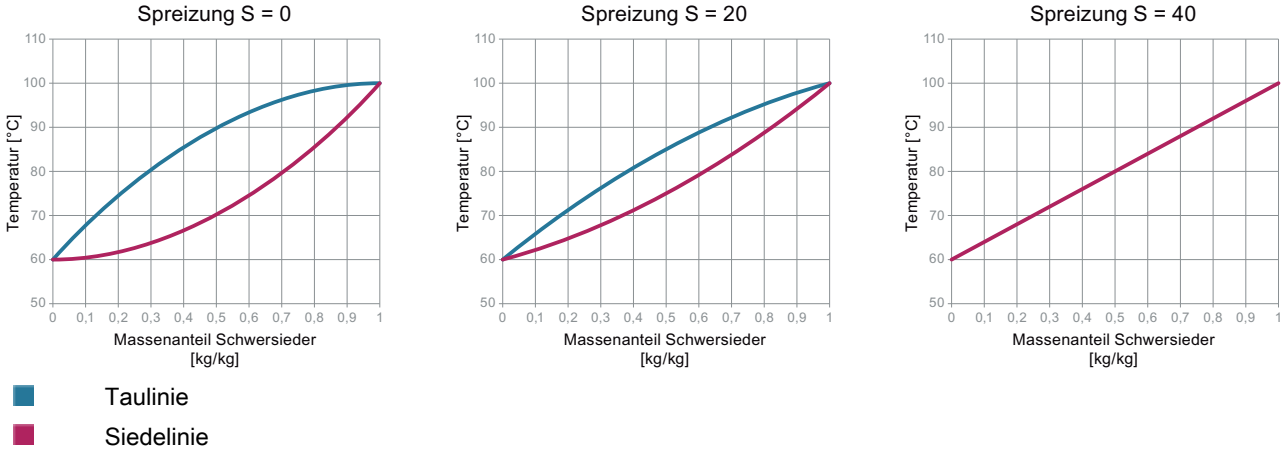


Bild 8-73 Siedelinse in Abhängigkeit des Parameters *S* (Spreizung)

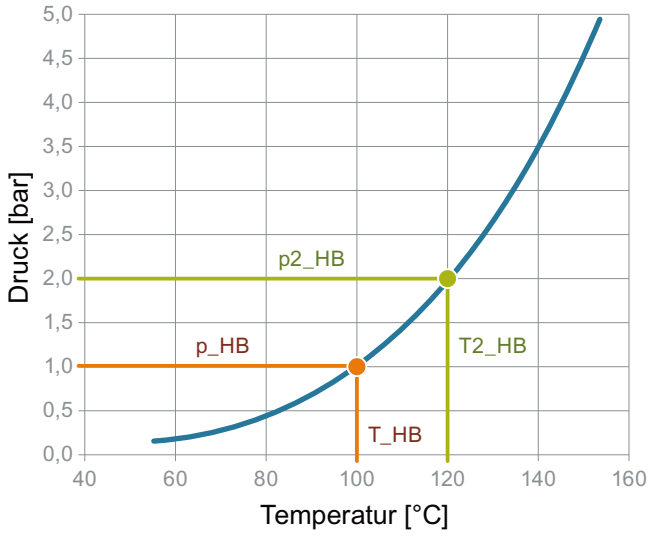


Bild 8-74 Abhängigkeit der Siedetemperatur vom Druck

Beschaltung

Die Anschlüsse *N1 ..Nn* dienen als Anschlusspunkte für die Stoffströme, d. h. zur Verschaltung von Rohrleitungen. Die Anzahl dieser Anschlusspunkte hängt vom zu simulierenden System ab und wird über den Parameter *NbrOfConnectors* vorgegeben. Alle Anschlüsse des Eingangsvektors *N1..Nn* sind zunächst gleichwertig und können für alle Anschlusstypen (Wasser/Dampf, ideales Gas, Liquid) verwendet werden.

Das Stoffmedium in den angrenzenden Flussnetzen wird über die Komponente *NetParam* definiert. Es müssen alle definierten Eingänge im Eingangsvektor *N1 ..Nn* verschaltet werden. Während der Simulation darf keiner dieser Eingänge offenbleiben, da damit sonst offene Flussnetze angelegt würden. Dies würde dann bei der Codegenerierung zu einer Fehlermeldung führen.

Beim Anschlusstyp Wasser/Dampf wird das einströmende Medium intern im Tank mit einem Liquid-Medium vermischt. Die einströmende Masse und Enthalpie werden berücksichtigt.

Typischerweise hat ein Flash drei Anschlüsse. Über den ersten Anschluss erfolgt der Zulauf. Am zweiten Anschluss am Kopf des Behälters wird der Gasstrom abgezogen. Am dritten Anschluss am Boden des Behälters wird der flüssige Ablauf angeschlossen.

An dem Flash muss mindestens ein Gasstrom angeschlossen sein.

Eingangsvektor *z_in* und Ausgangsvektor *z_aus*

Über den Eingangsvektor *z_in* kann für jeden der Anschlüsse *N1..Nn* der Massenanteil des Leichtsieders im einströmenden Medium vorgeben werden. Anschlüsse ohne Leichtsieder-Bestandteil werden mit dem Wert "0" belegt. Die vorgegebenen Werte gelten ausschließlich für eintretende Stoffströme.

Für alle austretenden Stoffströme wird der Massenanteil des Leichtsieders an den Ausgangsvektor *z_aus* übergeben. Für gasförmige Anschlüsse wird die Konzentration der Gas-Phase ausgegeben. Für alle anderen Anschlüsse wird die Konzentration in der flüssigen Phase ausgegeben. Im Fall von austretenden Massenströmen wird ein gegebenenfalls eingetragener Parameter in *z_in* nicht ausgewertet.

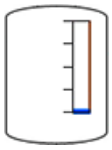
Parameter *ConnectorHeight*

Zu jedem der Anschlusspunkte *N1..Nn* wird die Einbauhöhe des Stutzens am Flash benötigt (Parameter *ConnectorHeight*). Die Angabe ist in Metern. Die Voreinstellung ist jeweils 0 m, also am Boden des Behälters. Negative Werte sind für die Nachbildung von tieferliegenden Anschlüssen zulässig. Liegt ein Gasanschluss unterhalb des Füllstandes des Behälters, so kann weder Flüssigkeit noch Gas durch diesen Anschluss aus dem Behälter hinausströmen. Ein Einströmen von Gas ist allerdings möglich.

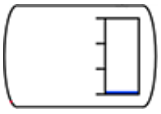
Ausrichtung

Die Komponente *Flash* kann entweder als stehender oder liegender Behälter betrieben werden. Die Ausrichtung kann über den Parameter *Position* eingestellt werden. Dabei wird der Behälter jeweils als stehender (vertikal) oder liegender (horizontal) Zylinder simuliert. Im Fall des stehenden Flashs gibt der Parameter *HeightOrLength* die Höhe des Tanks an, für einen liegenden Tank die Länge.

Stehender Flash



Liegender Flash



Beheizung / Kühlung des Flash-Behälters

Der Flash kann über folgende Möglichkeiten beheizt oder gekühlt werden:

- Kombination mit Komponente *HeatingJacket*
- Kombination mit Komponente *ElectricalHeatingJacket*
- Anschluss eines *Sliders* am Eingang Q

Überschreiben von Zustandsgrößen

Über das Bedienfenster der Komponente kann der Bediener folgende Zustandsgrößen während der Simulation auf beliebige zulässige Werte setzen:

- Gesamtmasse m
- Temperatur T
- Gesamtmassenanteil z
- Füllstand L

Hierzu wird zunächst den Sollwert in die entsprechende Digitaleingabe *SetValues* eingegeben und durch Anklicken des Buttons "Set" übernommen. Der Zustand "Gesamtmasse" kann hierbei sowohl über den Wert der Masse als auch über den Füllstand (Level) vorgegeben werden. Über diese Beeinflussung von Zustandsgrößen können z. B. Aufwärmvorgänge oder Füllvorgänge beschleunigt werden. Ebenso kann der *Flash* entleert werden, beispielsweise um in einer Schrittkette zurückzuspringen.

Grenzfall "Flash leer"

Zuflüsse und Abflüsse aus dem Flash werden generell am Stutzen des Tanks gedrosselt. Im normalen Betrieb ist diese Drosselung für den Anwender nicht direkt erkennbar, da die Drosselung nur wenig Einfluss auf die Simulation haben soll. Für einen leeren Tank wird der Abfluss stark gedrosselt. Der Durchflusskoeffizient wird dazu für alle Anschlüsse, über die Wasser oder Dampf abfließt, auf den Wert für maximale Drosselung gesetzt. Diese Durchflusskoeffizienten werden intern in Abhängigkeit vom Tankvolumen berechnet und können durch den Modellierer nicht beeinflusst werden. Der Tank wird als leer angenommen, wenn die Flüssigkeitsmenge *mL/ρ* kleiner als eine gegebene minimale prozentuale Füllung *MinLiquidVolumePercent* wird:

$m_L < MinLiquidVolumePercent \cdot Volume \cdot \rho$

Die Bilanzierung der Zustände wird für einen leeren Flash gestoppt, d. h. Änderungen der Gesamtmasse, Komponentenmasse und spezifischen Enthalpie werden verworfen. Der Zustand "leer" wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Flüssigkeitsmenge vorliegt. Im Symbol einer Komponente wird für einen leeren Tank ein entsprechender Hinweis eingeblendet (siehe Abbildung).

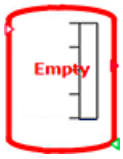


Bild 8-75 Anzeige eines leeren Flashs im Symbol

Grenzfall "Flash voll"

Der Flash wird als voll angenommen, wenn seine Flüssigkeitsmenge den maximal möglichen prozentualen Wert erreicht hat:

$m_L > (V - MinVaporVolumePercent \cdot Volume) \cdot \rho$

Die Bilanzierung der Zustände wird für einen vollen Flash gestoppt, d. h. Änderungen der Gesamtmasse, Komponentenmasse und spezifischen Enthalpie werden verworfen. Der Zustand "voll" wird erst wieder verlassen, wenn eine genügend große Abnahme der Flüssigkeitsmenge vorliegt. Im Symbol einer Komponente wird für einen vollen Flash ein entsprechender Hinweis eingeblendet (siehe Abbildung).

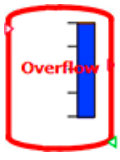


Bild 8-76 Anzeige des vollen Flashs im Symbol

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Tanks; online änderbar	m^3	1.0
HeightOrLength	Höhe des stehenden bzw. Länge des liegenden Tanks; online änderbar	m	1.0

Parametername	Beschreibung	Einheit	Standardwert
NbrOfConnectors	Anzahl N der Anschlüsse	–	3
NbrOfMeasurements	Anzahl N der Messstellen	–	1
MeasurementHeight	Höhe der Messstellen über dem Tankboden	m	0.0
ConnectorHeight	Höhe der Rohranschlüsse über dem Tankboden	m	0.0
Position	Ausrichtung des Tanks (stehend oder liegend)	–	vertically

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
InitLevelPercent	Initialisierungsfüllstand	%	50.0
InitTemperature	Initialisierungstemperatur	°C	80.0
Initz	Initialisierungszusammensetzung Leichtsieder	kg/kg	0.5
InitEvaporationPart	Initialisierungsmassenanteil des leichtsiedenden Mediums	%	0.0
InitcpLiquid	Wärmekapazität der Flüssigkeit für Initialisierung	kJ/kg/K	4.18
InitcpVapor	Wärmekapazität der Gasphase für Initialisierung	kJ/kg/K	1.0
InitDensityLiquid	Flüssig-Dichte im Tank für Initialisierung	kg/m³	997.337
InitRS	Spezifische Gaskonstante für Initialisierung	kJ/kg/K	0.287
geoHeight	Geodätische Höhe des Tanks	m	0.0
T_LB	Siedetemperatur Leichtsieder	°C	60
T_HB	Siedetemperatur Schwersieder	°C	100
S	Spreizung Siedelinse	–	10
hv_HB	Verdampfungsenthalpie Schwersieder	kJ/kg/k	2300
hc_LB	Verdampfungsenthalpie Leichtsieder	kJ/kg/K	1800
T2_HB	Siedetemperatur Schwersieder bei *p2_HB*	°C	120
P2_HB	Druck für Siedetemperatur *T2_HB*	Bar	2.0
MinVaporVolumePercent	Prozentualer Anteil des Tankvolumens, das beim Füllen des Tanks nicht mit Flüssigkeit gefüllt werden kann	%	5
MinLiquidVolumePercent	Prozentualer Anteil des Tankvolumens, das beim Entleeren des Tanks im Tank verbleibt	%	0.5

Bedienfenster

Größe	Formelzeichen	Einheit	Größe änderbar?
Füllstand	L	m	Ja
Temperatur	T	°C	Ja
Gesamtmasse	m	kg	Ja
Druck	p	bar	Nein
Massenanteil Leichtsieder	z	%	Ja
Massenanteil Leichtsieder Flüssigphase	x	%	Nein
Massenanteil Leichtsieder Gasphase	y	%	Nein
Dampfanteil	vf	%	Nein

HeatingJacket – Heizmantel

Symbol

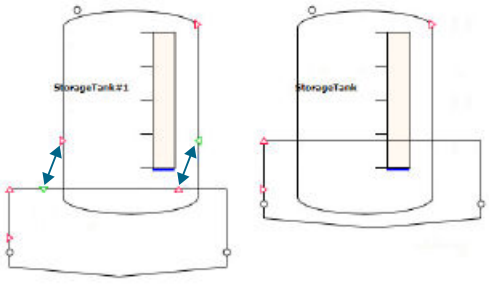


Bild 8-77 Symbol des Komponententyps HeatingJacket in Verbindung mit der Komponente StorageTankLiquid, Darstellung der zu verbindenden Anschlüsse

Funktion

Die Komponente *HeatingJacket* ist für den Einsatz in Verbindung mit einem *StorageTankLiquid* optimiert. Sie stellt eine AddOn-Komponte zum *StorageTankLiquid* dar, kann aber auch in Verbindung mit der Drucknutsche oder einzeln eingesetzt werden.

Verwendung in Verbindung mit einem StorageTankLiquid

Achten Sie bei Verwendung der Komponente *HeatingJacket* in Verbindung mit einem *StorageTankLiquid* darauf, dass der Signalaustausch zwischen beiden Komponenten über zwei Anschlüsse erfolgt. Diese sind:

- Wärmestrom Q (Ausgang aus dem *HeatingJacket*, Eingang für den *StorageTankLiquid*)
- Messwerteanschluss M (Ausgang aus dem *StorageTankLiquid*, Eingang für das *HeatingJacket*)
  Der hier verwendete Messwertanschluss muss am Behälter zwingend auf der Behälterhöhe "null" bzw. unterhalb des zu erwartenden Flüssigkeitsniveaus liegen (Parameter MeasurementHeight)

Die Größe beider Komponenten und die Position der genannten Anschlüsse sind so voreingestellt, dass beide Komponenten wie in der obigen Abbildung links dargestellt übereinandergelegt werden können, so dass die entsprechenden Verbindungen automatisch geschlossen werden. In diesem Fall verschwinden die Anschlüsse, wie in der obigen Abbildung rechts dargestellt. Alternativ können Sie die Verbindungen auch über Verbindungslinien ziehen.

Die Komponente *HeatingJacket* muss als Zweigelement in ein Flussnetz eingebunden werden.

Wegen des sich einstellenden Temperaturunterschieds im *HeatingJacket* und im *StorageTankLiquid* entsteht ein (Ausgleichs-)Wärmestrom Q.

Dieser Wärmestrom wird vereinfacht über folgende Beziehung berechnet:

$$\dot{Q} = k\ A\ \Delta T$$

k	Wärmedurchgangskoeffizient
A	Wärmeübergangsfläche m^2
ΔT	Temperaturdifferenz zwischen Heiz-Kühlmedium und Behältertemperatur

Einfluss eines Rührwerks auf den Wärmeübergang

Sie können den Einfluss eines Rührwerks auf den Wärmeübergang über den Parameter *AgitatorInfluence* darstellen. Mit diesem Parameter wird der berechnete Wärmeübergang Q in Abhängigkeit von der Rührerdrehzahl skaliert. Dabei entspricht 100 % dem vollen Wärmeübergang und 0 % keinem Wärmeübergang.

Die Voreinstellung für den Parameter *AgitatorInfluence* ist über den gesamten Drehzahlbereich des Rührers 100 %, also immer voller Wärmeübergang, kein Einfluss des Rührers auf den Wärmeübergang.

Über den Eingang *AgitatorSpeed* können Sie die Drehzahl des Rührwerks an die Komponente *HeatingJacket* koppeln.

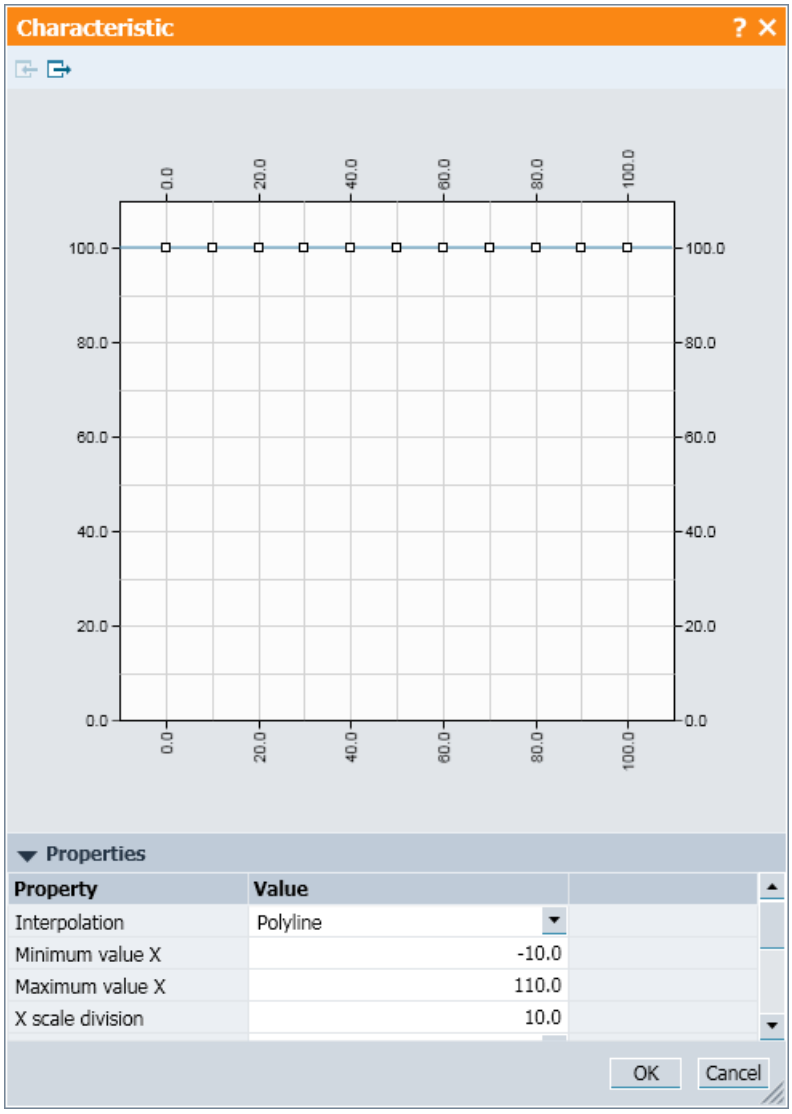


Bild 8-78 Parameter AgitatorInfluence des Komponententyps HeatingJacket

Einfluss des Füllstandes auf den Wärmeübergang

Den Einfluss des Tankfüllstandes auf den Wärmeübergang können Sie über den Parameter *LevelInfluence* darstellen. Mit diesem Parameter wird der berechnete Wärmeübergang in Abhängigkeit vom Tankfüllstand skaliert. Dabei entspricht 100 % dem vollen Wärmeübergang und 0 % keinem Wärmeübergang.

Die Voreinstellung für diesen Parameter ist über den gesamten Füllstandsbereich des Tanks 100 %, also kein Einfluss des Füllstandes auf den Wärmeübergang.

Über den Eingang *TankLevel* können Sie den Tankfüllstand an die Komponente *HeatingJacket* koppeln. Hierfür verwenden Sie den prozentualen Tankfüllstand.

Die Einflüsse des Rührwerks und des Füllstandes werden multiplikativ miteinander verknüpft.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Ventilkennwert k_{VS} mit $k_{VS} \geq k_{V0} + 10^{-6}$ m^3/h	m^3/h	10.0
NbrOfMeasurements	Anzahl N der Messstellen	–	1
AgitatorInfluence	Einfluss eines Rührwerks auf den Wärmeübergang in Prozent in Abhängigkeit von der Drehzahl des Rührwerks	–	
LevelInfluence	Einfluss des Tankfüllstandes auf den Wärmeübergang in Prozent in Abhängigkeit vom Tankfüllstand	–	

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen des Heizmantels	m^3	0.1
Area	Wärmeübergangsfläche	m^2	10.0
k	Wärmeübergangskoeffizient [$kW/(m^2 \cdot K)$]	–	1.0
InitialTemperature	Initialisierungstemperatur	°C	20.0
MinEnthalpy	Kleinste Enthalpie	kJ/kg	−300.0
MaxEnthalpy	Größte Enthalpie	kJ/kg	3000.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Temperatur	T	°C
Aktuelle Durchströmung	–	kg/s

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Volumen	–	m^3
kvs	h	m^3/h
Wärmeübergangsfläche	–	m^2

StorageTankLiquid – Speichertank für flüssige Medien

Symbol

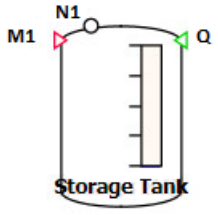


Bild 8-79 Symbol des Komponententyps StorageTankLiquid

Funktion

Der Komponententyp *StorageTankLiquid* dient zur Mischung und Speicherung von flüssigen Medien. Hierbei werden die ein- und austretenden Stoffströme automatisch bezüglich Masse und Enthalpie bilanziert.

Eine Temperierung des im Tank gespeicherten Stoffs ist über einen Direkteingang oder verschiedene Zusatzkomponenten möglich.

Vorhandene Gasanschlüsse werden separat bilanziert und bilden ein Druckpolster oberhalb der Flüssigkeit im Tank.

Die im Tank bilanzierte Masse ist ein Gemisch aus maximal zwei flüssigen Komponenten. Die Konzentration dieser Komponenten in den Eingangsströmen kann an jedem Eingang vorgegeben werden.

Während der Simulation ist eine thermische Trennung der beiden Komponenten möglich. Im Folgenden werden diese Komponenten daher als Leichtsieder und Schwersieder bezeichnet.

Ebenso sind natürlich auch andere physikalische Vorgänge möglich – z. B. die Abmischung von Konzentrationen.

Beschaltung

Die Anschlüsse *N1..Nn* dienen als Anschlusspunkte für die Stoffströme, d. h. zur Verschaltung von Rohrleitungen.

Die Anzahl dieser Anschlusspunkte hängt vom zu simulierenden System ab und wird über den Parameter *NbrOfConnectors* vorgegeben. Alle Anschlüsse des Eingangsvektors *N1..Nn* sind zunächst gleichwertig und können für alle Anschlusstypen (Wasser/Dampf, ideales Gas, Liquid) verwendet werden.

Anschlusstypen können sein:

- Flüssigzuläufe von verschiedenen chemischen Stoffen
- Abläufe der im Tank gespeicherten Stoffe
- Ein-/Austritt für Reinigungsmedien
- Ein-/Austritt für Waschlösungen

- Gasein- und Gasaustritte
- Entlüftung

Das Stoffmedium in den angrenzenden Flussnetzen wird über die Komponente *NetParam* definiert.

Sie müssen alle definierten Eingänge im Eingangsvektor *N1 ..Nn* verschalten. Während der Simulation darf keiner dieser Eingänge offenbleiben, da damit sonst offene Flussnetze angelegt würden. Dies würde dann bei der Codegenerierung zu einer Fehlermeldung führen.

Beim Anschlusstyp Wasser/Dampf wird das einströmende Medium intern im Tank mit einem Liquid-Medium vermischt. Die einströmende Masse und Enthalpie werden berücksichtigt, im Modell bildet sich jedoch kein einsprechendes Phasengleichgewicht aus. Einströmender Dampf trägt in der Gasphase nicht zum Druckaufbau bei. Wasser wird im Gemisch wie ein Liquid-Medium behandelt.

Aufgrund der fehlenden Wasserdampfeigenschaften ist derzeit somit eine Dampfsterilisation mit einem Flussnetz-Anschluss Wasser/Dampf nicht möglich.

Parameter ConnectorHeight

Zu jedem der Anschlusspunkte *N1..Nn* wird die Einbauhöhe des Stutzens am Tank benötigt (Parameter *ConnectorHeight*). Die Angabe ist in Metern. Die Voreinstellung ist jeweils 0 m, also am Boden des Tanks. Negative Werte sind für die Nachbildung von tieferliegenden Anschlüssen zulässig. Liegt ein Gasanschluss unterhalb des Füllstandes des Tanks, so kann weder Flüssigkeit noch Gas durch diesen Anschluss aus dem Behälter hinausströmen.

Eingangsvektor EvapFraction

Über den Eingangsvektor *EvapFraction* können Sie für jeden der Anschlüsse *N1..Nn* den Anteil des leichtsiedenden Gemischbestandteils im einströmenden Medium in Prozent vorgeben. Anschlüsse ohne leichtsiedenden Gemischbestandteil (z. B. Waschmittel) werden mit dem Wert "0" (Voreinstellung) belegt. Gasanschlüsse erhalten ebenfalls den Wert "0".

Die vorgegebenen Werte gelten ausschließlich für eintretende Stoffströme. Alle austretenden Stoffströme strömen mit dem im Tank aktuell gespeicherten leichtsiedenden Gemischbestandteil an der Gesamtmasse *Rate* aus. Dieser Anteil wird basierend auf den ein- und austretenden Massenströmen und ihren Parametern *EvapFraction* laufend berechnet.

Im Fall von austretenden Massenströmen wird ein gegebenenfalls eingetragener Parameter *EvapFraction* nicht ausgewertet.

Bilanzierung der Flüssigphase

Die Flüssigphase wird bezüglich der ein- und austretenden Massen und Enthalpien bilanziert. Zusätzlich erfolgt eine Massenbilanzierung des leichtsiedenden Gemischbestandteils.

$$\frac{dm_{Liquid}}{dt} = \sum_{i=1}^{n} \dot{m}_{Liquid,i} - \dot{m}_{Evap}$$

$$\frac{dm_{EvapFraction}}{dt} = \sum_{i=1}^{n} \dot{m}_{EvapFraction,i} - \dot{m}_{Evap}$$

$$\frac{dh_{Liquid}}{dt} = \frac{1}{m_{Liquid}} \left(\sum_{i}^{n} \dot{m}_{Liquid,i}\, h_{Liquid,i} + \sum \dot{Q} - \dot{m}_{Evap} \left(h' + \Delta H_V \right) \right)$$

$$T_{Liquid} = \frac{h_{Liquid}}{c_P}$$

Gasbilanzierung

Für den Fall, dass mindestens ein Gasanschluss angeschlossen ist, wird auch ein entsprechender Gasdruck berechnet. In diesem Fall wird von einem geschlossenen, also druckdichten Speichertank ausgegangen. Durch Ein- oder Ausspeichern von Gas und/oder Veränderung des Gasvolumens (z. B. durch Füllen des Tanks) wird der Gasdruck verändert. Der berechnete Gasdruck wirkt ebenfalls auf die Flüssigkeitsanschlüsse. Der Zustand der Gasbilanzierung wird im Symbol durch eine orangefarbene Hinterlegung der Füllstandsanzeige erreicht.

Innerhalb des Gasraums werden die ein- und austretenden Massen und Enthalpien bilanziert:

$$\frac{dm_{Gas}}{dt} = \sum_{i}^{n} \dot{m}_{Gas,i} - \dot{m}_{Evap}$$

$$\frac{dh_{Gas}}{dt} = \frac{1}{m_{Gas}} \left(\sum_{i}^{n} \dot{m}_{Gas,i}\, h_{Gas,i} + \dot{m}_{Evap} \left(h' + \Delta H_V \right) \right)$$

Der Gasdruck wird dann entsprechend der Zustandsgleichung idealer Gase berechnet:

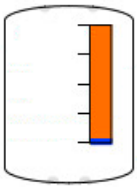
$$p_{Gas} = \frac{m_{Gas}\, R_S\, T_{Gas}}{V_{Gas}}$$

Die Gastemperatur T_{Gas} wird aus der bilanzierten Gasenthalpie h_{Gas} berechnet. Bei unterschiedlicher Gastemperatur und Flüssigtemperatur erfolgt während der Simulation ein entsprechender Temperaturausgleich.

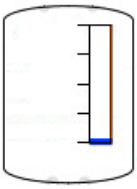
Falls kein Gasanschluss verbunden ist, entfällt die Gasbilanzierung. In diesem Fall wird der Gasdruck durch den Umgebungsdruck ersetzt. In diesem Fall wird von einem gegen die Umgebung offenen Tank ausgegangen.

Darstellung der Bilanzierung im Speichertank mit und ohne Gasanteil:

Bilanzierung mit Gasanteil



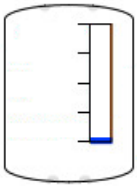
Bilanzierung als offener Tank



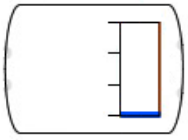
Ausrichtung

Die Komponente *StorageTankLiquid* kann entweder als stehender oder liegender Tank betrieben werden. Die Ausrichtung kann über den Parameter *Position* eingestellt werden. Dabei wird der Tank jeweils als stehender (vertikal) oder liegender (horizontal) Zylinder simuliert. Der Parameter *HeightOrLength* gibt für einen stehenden Tank die Höhe des Tanks an, für einen liegenden Tank dessen Länge.

Stehender Tank



Liegender Tank



Verdampfung

Wenn in der gespeicherten Flüssigkeit ein leichtsiedender Bestandteil enthalten ist, ist es möglich, diesen zu verdampfen. Hierzu muss das Flüssigkeitsgemisch auf eine Temperatur erwärmt werden, die über der dem aktuellen Druck entsprechenden Siedetemperatur liegt. Die hierfür notwendige Dampfdruckkurve wird über den Parameter *Boiling* parametriert (Dampfdruckkurve des Leichtsieders $p_S = f(T_S)$ in [bar]).

Des Weiteren muss über den Parameter *Vaporization* die Verdampfungsenthalpie des Leichtsieders als Kennlinie vorgegeben werden ($\Delta H_V = f(T_S)$ in [kJ/kg]).

Voreingestellt sind diese Kennlinien für Wasser. Sie können innerhalb der Diagramme editiert oder aus einem Textfile eingelesen werden.

Diese Kennlinien sind beispielhaft in den folgenden Abbildungen dargestellt:

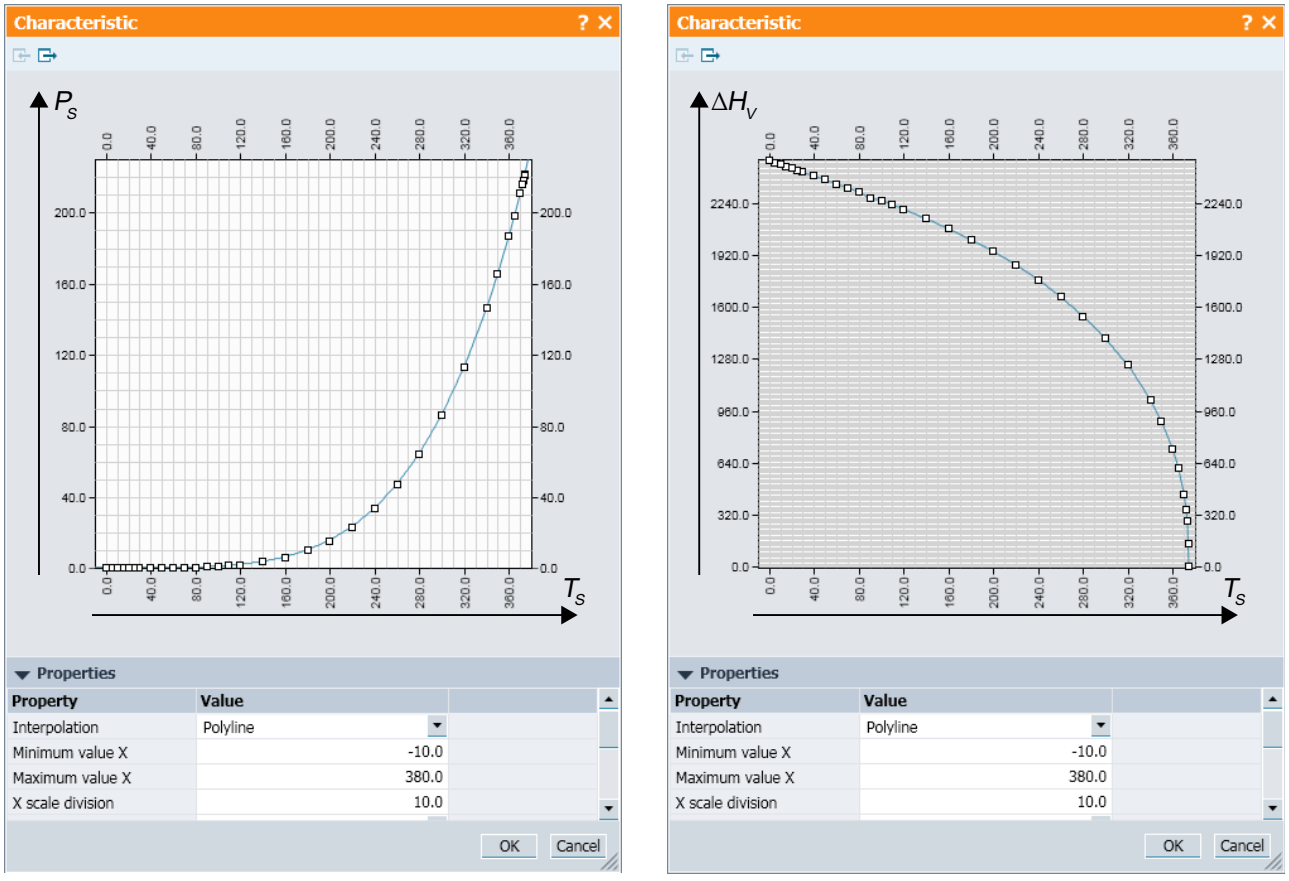


Bild 8-80 Beispiel für Verdampfungsenthalpie und Dampfdruckkurve

Wenn die Verdampfungsbedingung $T > T_S(p_S)$ erreicht ist, geht der Leichtsieder schrittweise in die gasförmige Phase über und wird dem Gasbilanzraum zugeführt. Eine Kondensation im Behälter ist nicht vorgesehen.

Beheizung / Kühlung des Speichertanks

Der *StorageTankLiquid* kann über folgende Möglichkeiten beheizt oder gekühlt werden:

- Kombination mit Komponente *HeatingJacket*
- Kombination mit Komponente *ElectricalHeatingJacket*
- Anschluss eines Sliders am Eingang *Q*

Der zugeführte Wärmestrom wird vereinfachend ausschließlich zur Erwärmung der Flüssigphase verwendet.

Überschreiben von Zustandsgrößen

Über das Bedienfenster der Komponente können Sie diese beiden Zustandsgrößen während der Simulation vom Bediener auf beliebige zulässige Werte setzen:

- Gespeicherte Masse
- Temperatur

Hierzu geben Sie zunächst den Sollwert in die entsprechende Digitaleingabe *SetValues* ein und übernehmen mit dem Button "Set". Der Zustand "gespeicherte Masse" kann hierbei sowohl über den Wert der Masse als auch über den Füllstand (*Level*) vorgegeben werden.

Über diese Beeinflussung von Zustandsgrößen können Sie z. B. Aufwärmvorgänge oder Füllvorgänge beschleunigen. Ebenso können Sie den Tank entleeren, beispielsweise um in einer Schrittkette zurückzuspringen.

Grenzfall "Tank leer"

Zuflüsse in den Tank und Abflüsse aus dem Tank werden generell am Stutzen des Tanks gedrosselt. Im normalen Betrieb ist diese Drosselung für den Anwender nicht direkt erkennbar, da die Drosselung nur wenig Einfluss auf die Simulation haben soll.

Für einen leeren Tank wird der Abfluss stark gedrosselt. Der Durchflusskoeffizient wird dazu für alle Anschlüsse, über die Wasser oder Dampf abfließt, auf den Wert für maximale Drosselung gesetzt.

Diese Durchflusskoeffizienten werden intern in Abhängigkeit vom Tankvolumen berechnet und können durch den Modellierer nicht beeinflusst werden.

Der Tank wird als leer angenommen, wenn die Flüssigkeitsmenge m_L/ρ kleiner als eine gegebene minimale prozentuale Füllung *RemainderLiquidPercent* wird:

$m_L < RemainderLiquidPercent \cdot Volume \cdot \rho$

Die Bilanzierung der Zustände wird für einen leeren Tank gestoppt, d. h. Änderungen der Masse, Teilmassen und spezifischen Enthalpie werden verworfen. Der Zustand "leer" wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Flüssigkeitsmenge vorliegt, d. h. die Differenz aus einströmender und abfließender Menge positiv ist.

Im Symbol einer Komponente wird für einen leeren Tank ein entsprechender Hinweis eingeblendet (siehe Abbildung).

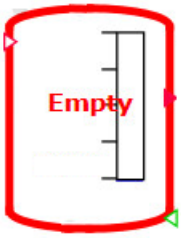


Bild 8-81 Anzeige eines leeren Tanks im Symbol

Grenzfall "Tank voll"

Der Tank wird als voll angenommen, wenn seine Flüssigkeitsmenge den maximal möglichen prozentualen Wert erreicht hat:

$m_L < (V - RemainderGasPercent \cdot Volume) \cdot \rho$

Die Bilanzierung der Zustände wird für einen vollen Tank gestoppt, d. h. Änderungen der gespeicherten Masse, Teilmasse und spezifischen Enthalpie werden verworfen. Der Zustand "voll" wird erst wieder verlassen, wenn eine genügend große Abnahme der Flüssigkeitsmenge vorliegt, d. h. die Differenz aus einströmender und abfließender Menge negativ ist.

Im Symbol einer Komponente wird für einen vollen Tank ein entsprechender Hinweis eingeblendet (siehe Abbildung).

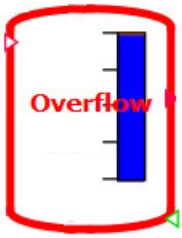


Bild 8-82 Anzeige des vollen Tanks im Symbol

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Tanks; online änderbar	m^3	1.0
HeightOrLength	Höhe des stehenden bzw. Länge des liegenden Tanks; online änderbar	m	1.0
NbrOfConnectors	Anzahl N der Anschlüsse	–	1
NbrOfMeasurements	Anzahl N der Messstellen	–	1
MeasurementHeight [Nr]	Höhe der Messstellen über dem Tankboden	m	0.0
ConnectorHeight [Nr]	Höhe der Rohranschlüsse über dem Tankboden	m	0.0
Boiling	Dampfdruckkurve des Leichtsieders: Druck in bar als Funktion der Temperatur (°C)	–	
Vaporization	Verdampfungsenthalpie des Leichtsieders in kJ/kg als Funktion der Temperatur (°C)	–	
Position	Ausrichtung des Tanks (stehend oder liegend)	–	vertically

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureOutside	Umgebungsdruck	bar	1.0
LevelInit	Initialisierungsfüllstand	%	3.0
TemperatureInit	Initialisierungstemperatur	°C	20.0
PressureInit	Initialisierungsdruck	bar	1.0
InitEvaporationPart	Initialisierungsmassenanteil des leichtsiedenden Mediums	%	0.0
Density	Gemischdichte im Tank	kg/m^3	997.337
RemainderGasPercent	Prozentualer Anteil des Tankvolumens, das beim Füllen des Tanks nicht mit Flüssigkeit gefüllt werden kann	%	5

Parametername	Beschreibung	Einheit	Standardwert
RemainderLiquidPercent	Prozentualer Anteil des Tankvolumens, das beim Entleeren des Tanks im Tank verbleibt	%	0.5
geoHeight	Geodätische Höhe des Tanks	m	0.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Füllstand	L	m	Ja
Temperatur Flüssigkeit	T_L	°C	Ja
Temperatur Gas	T_G	°C	Nein
Masse	m	kg	Ja
Druck am Tankboden	p_B	bar	Nein
Gasdruck	p_G	bar	Nein
Massenanteil Leichtsieder	w	%	Ja

Siehe auch

Geodätische Höhe (Seite 612)

8.2.5.11 Valves

ControlValve / Valve / AngleValve – Regelventil / Ventil / Eckventil

Symbol

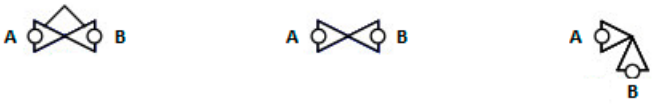


Bild 8-83 Symbole der Komponententypen ControlValve / Valve / AngleValve

Funktion

Die Komponententypen *ControlValve*, *Valve* und *AngleValve* dienen zur Simulation eines Stellventils. Sie können aber auch für Klappen, Hähne oder Schieber eingesetzt werden.

Die genannten Komponenten unterscheiden sich ausschließlich in ihren Symbolen. Das Verhalten der Komponenten ist identisch.

Abhängig von der Ventilstellung, der parametrierten Kennlinie und der Ventilgröße (k_{VS}-Wert) wird der Druckabfall über dem Ventil gemäß folgender Formel berechnet.

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Dabei ist

$\Delta p = p_B - p_A$	der Druckabfall über dem Ventil in bar
$\dot{m}$	der Durchfluss bzw. Massenstrom in kg/s
ρ	die Dichte des Mediums in kg/m³
k_{VS}	der Ventilkennwert in m³/h

Die Bezugsrichtung für den Medienfluss $\dot{m}$ geht vom Anschluss *A* zum Anschluss *B*. Für einen Medienfluss in Bezugsrichtung gilt also $\dot{m} > 0$. Der Druckabfall hat dann einen negativen Wert: $\Delta p < 0$. Hierbei handelt es sich jedoch nur um reine Zählrichtung, d. h. prinzipiell ist die Einbaurichtung des Ventils egal. Ebenso kann auch während der Simulation eine Strömungsumkehr erfolgen.

Die Ansteuerung dieser Komponenten durch die Leittechnik ist nicht in den Komponententypen enthalten. Diese werden in der Aktor-Sensor-Ebene abgebildet. Die Schnittstelle zwischen der Antriebsebene eines Ventils und den Komponententypen *ControlValve,Valve* und *AngleValve* in der Modellebene ist die Ventilstellung.

Dieser Stellungswert wird begrenzt auf den Bereich 0 ≤ Position ≤ 100 %. Die Ventilstellung wird über die Ventilkennlinie auf den Ventilkennwert k_l abgebildet.

Sie können folgende Ventilkennlinien auswählen:

- Lineare Kennlinie
- Quadratische Kennlinie
- Gleichprozentige Kennlinie
- Grafisch projektierte Kennlinie (Polygonzug)

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Characteristic	Auswahl der Ventilcharakteristik	–	*Linear*
Kvs	Ventilkennwert k_{VS} mit $k_{VS} \geq k_{V0} + 10^{-6}$ m³/h	m³/h	10.0
OpeningCharacteristic	Vorgabe der Ventilcharakteristik als Kennlinie. Dieser Parameter wird nur verwendet, wenn für den Parameter *Characteristic* der Wert *Polygon* ausgewählt wurde. Hierbei wird die prozentuale Öffnung in Abhängigkeit vom prozentualen Hub angegeben.		

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m^3/h	m^3/h	0.000001

Bedienfenster

Im Bedienfenster können Sie zwischen automatisierter Fahrweise (Steuerung über die Aktor-Sensor-Ebene) und manueller Fahrweise umschalten. Voreingestellt ist immer die automatisierte Fahrweise.

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckunterschied	Δp	bar
Massenstrom	$\dot{m}$	kg/s

RuptureDisc – Berstscheibe

Symbol



Bild 8-84 Symbol des Komponententyps RuptureDisc

Funktion

Der Komponententyp *RuptureDisc* bildet die Funktion einer Berstscheibe nach. Abhängig von einem parametrierbaren Ansprechdruck wird eine Membran dauerhaft zerstört bzw. aus Simulationssicht ein Ventil irreversibel geöffnet.

Über den Button "Reset" des Bedienfensters können Sie im Simulationsmodell jedoch die Berstscheibe nach dem Ansprechen wieder manuell verschließen. Diese Funktion wird benötigt, um das Ansprechen der Berstscheibe mehrmals nacheinander ohne Rücksetzen der Simulation zu testen.

Diese Funktion können Sie nur manuell über das Bedienfenster nutzen. Es besteht keine Möglichkeit zur Verschaltung mit anderen Komponenten oder der Leittechnik.

Simulationstechnisch wird die Berstscheibe vereinfacht als geschlossenes Ventil betrachtet, das bei Überschreiten des Ansprechdrucks dauerhaft geöffnet wird. Daher ist die Parametrierung wie bei einem Ventil.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Ventilkennwert k_{VS} mit $k_{VS} \geq k_{V0} + 10^{-6}$ m^3/h	m^3/h	100.0
OpeningPressure	Ansprechdruck der Berstscheibe	bar	10.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m^3/h	m^3/h	0.000001

Bedienfenster

Im Bedienfenster können Sie die Berstscheibe zurücksetzen.

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckunterschied	Δp	bar
Massenstrom	$\dot{m}$	kg/s

Während der Simulation werden die Betriebszustände der Komponente *RuptureDisc* im Symbol visualisiert:

Geschlossene Berstscheibe



Zerstörte Berstscheibe



Beispielverschaltung:

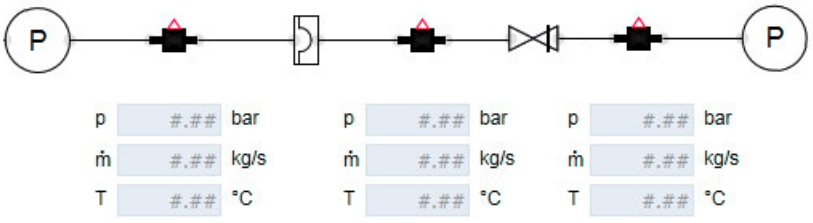


Bild 8-85 Verschaltungsmöglichkeit des Komponententyps RuptureDisc

In dieser Beispielverschaltung wird eine Berstscheibe einem Sicherheitsventil vorgeschaltet. Hierbei müssen Sie den k_{V0}-Wert des nachgeschalteten Sicherheitsventils deutlich größer wählen als den k_{V0}-Wert der Berstscheibe.

Beispiel: $k_{V0Berstscheibe}$ = 0,000001
$k_{V0Sicherheitsventil}$ = 0,01

Wenn diese beiden Werte gleich eingestellt würden, würde sich bei geschlossenem Ventil und Berstscheibe, bedingt durch den Lösungsalgorithmus des Flussnetzes, zwischen der Berstscheibe und dem Sicherheitsventil ein mittlerer Druck aus dem Druckwert vor der Berstscheibe und dem Druck nach dem Sicherheitsventil einstellen.

Ebenso müssen Sie den k_{VS}-Wert der Berstscheibe deutlich größer wählen als den k_{VS}-Wert des Sicherheitsventils.

StopValve – Rückschlagklappe

Symbol

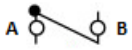


Bild 8-86 Symbol des Komponententyps StopValve

Funktion

Der Komponententyp *StopValve* dient zur Simulation eines Rückschlagventils. Abhängig von der Richtung des Durchflusses wird der Druckabfall über dem Rückschlagventil berechnet. Die Bezugsrichtung für den Medienfluss $\dot{m}$ geht vom Anschluss *A* zum Anschluss *B*. Für einen Medienfluss in Bezugsrichtung gilt also $\dot{m} > 0$.

Dabei wird das Ventil bei einem positiven Medienfluss als voll geöffnetes Ventil nachgebildet. Bei negativem Medienstrom stellt der Komponententyp ein geschlossenes Ventil dar. In diesem Fall wird die Leckage aktiv; der Ventilkennwert k_V entspricht also der Mindestöffnung k_{V0}.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m^3/h	m^3/h	10.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m^3/h	m^3/h	0.000001

Zur Realisierung der Rückschlagfunktion wählen Sie den Ventilkennwert k_{V0} genügend klein.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckunterschied	Δp	bar
Massenstrom	$\dot{m}$	kg/s

SafetyValve / AngleSafetyValve – Sicherheits-Ventil / Sicherheits-Eckventil

Symbol

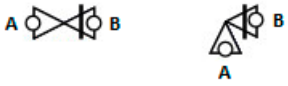


Bild 8-87 Symbole der Komponententypen SafetyValve / AngleSafetyValve

Funktion

Die Komponenten *SafetyValve* / *AngleSafetyValve* dienen als Sicherheitsventil. Bei einem parametrierbaren Druckunterschied zwischen den Anschlüssen *A* und *B* wird das Ventil in einer ebenfalls parametrierbaren Zeit geöffnet. Solange der Druckunterschied ansteht, bleibt das Ventil geöffnet. Das Ventil wird automatisch geschlossen, sobald der Druckunterschied unter den parametrierten Ansprechdruck abzüglich der parametrierten Hysterese gefallen ist.

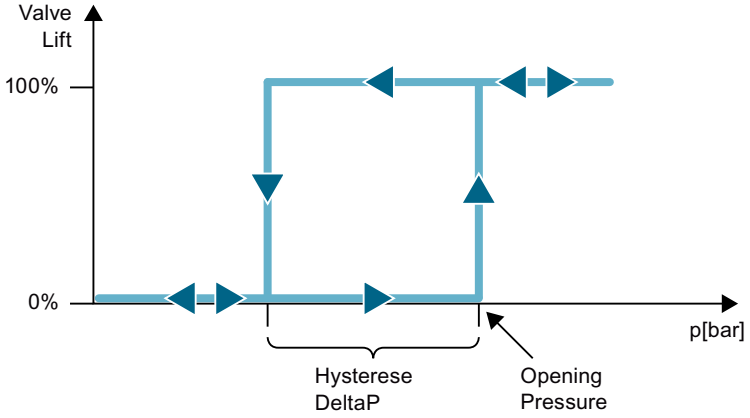


Bild 8-88 Öffnungs- und Schließverhalten des Sicherheitsventils

Das Modell des Überdruckventils ist verpolungssicher ausgeführt; es kann in beiden Richtungen durchströmt werden. Für die Berechnung des Druckunterschiedes ist unerheblich, an welcher Seite der höhere Druck anliegt. Damit wird erreicht, dass es im Modell nicht versehentlich (z. B. durch eine automatische Generierung) in verkehrter Richtung eingebaut wird.

Beachten Sie jedoch, dass innerhalb des Ventils eine Vorzugsrichtung der Strömung von *A* nach *B* definiert ist. Wenn die Strömung in dieser Richtung läuft, wird im Bedienfenster ein positiver Massenstrom angezeigt. Wenn sie in entgegengesetzter Richtung läuft (wenn das

Ventil beispielsweise gespiegelt ist), ist dieser Massenstrom negativ. Der Druckabfall Δp ist ebenfalls entsprechend negativ oder positiv.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Characteristic	Auswahl der Ventilcharakteristik Die Auswahl der Ventilcharakteristik ist wie bei Stellventilen. Die Hysterese wird durch den Parameter *HystereseDeltaP* erreicht	–	*Linear*
Kvs	Ventilkennwert k_{VS} mit $k_{VS} \geq k_{V0} + 10^{-6}$ m³/h	m³/h	10.0
OpeningPressure	Ansprechdruck (Druckdifferenz über das Ventil)	bar	10.0
OpeningTime	Zeit bis zur vollständigen Öffnung des Ventils	s	0.0
OpeningCharacteristic	Vorgabe der Ventilcharakteristik als Kennlinie. Dieser Parameter wird nur verwendet, wenn für den Parameter *Characteristic* der Wert *Polygon* ausgewählt wurde. Hierbei wird die prozentuale Öffnung in Abhängigkeit vom prozentualen Hub angegeben.	–	
HystereseDeltaP	Differenz zwischen Ansprechdruck und Abfalldruck	bar	1.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m³/h	m³/h	0.000001

Während der Simulation werden die Betriebszustände der Komponente im Symbol visualisiert.

Hierzu wird die Ventilstellung wie in der folgenden Abbildung dargestellt angezeigt:

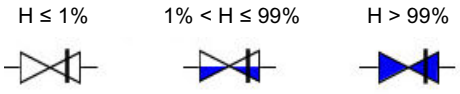


Bild 8-89 Anzeige der Ventilstellung im SafetyValve-Symbol

Alle Parameter sind online änderbar.

Bedienfenster

Im Bedienfenster können Sie über die Funktion "local control" das Sicherheitsventil auch manuell öffnen oder blockieren. Hiermit haben Sie die Möglichkeit zur einfachen Simulation eines Fehlverhaltens des Sicherheitsventils.

Bei manueller Fahrweise können Sie die Ventilstellung über einen Slider vorgeben. In diesem Fall entspricht die Funktionsweise dem Verhalten eines Handventils. Es besteht keine Abhängigkeit mehr vom Druckabfall.

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckunterschied	Δp	bar
Massenstrom	$\dot{m}$	kg/s

ThreeWayValve – Dreiwegeventil

Symbol

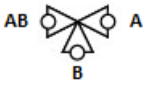


Bild 8-90 Symbol des Komponententyps ThreeWayValve

Funktion

Der Komponententyp *ThreeWayValve* dient zur Simulation eines Dreiwegeventils. Abhängig von der Ventilstellung können die in der Abbildung dargestellten Wege freigeschaltet werden.

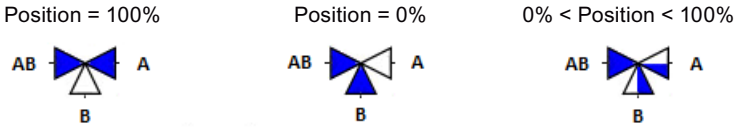


Bild 8-91 Stellungen des ThreeWayValve abhängig vom Eingang *Position*

Der Komponententyp besitzt zwei Strömungszweige. Der Anschluss *AB* wird immer durchströmt. Er ist der beiden Ventilzweigen gemeinsame Anschluss. Je nach Stellung des Ventils kann der Anschluss *A* oder der Anschluss *B* geöffnet sein (Zwischenstellungen sind möglich). Von *A* nach *B* oder umgekehrt ist keine Strömung möglich (analog zum realen Dreiwegeventil).

Je nach Einbau kann die Komponente als Mischventil oder als Verteilventil benutzt werden. Bei der Verschaltung muss die erwartete Strömung im Ventil (Mischventil oder Verteilventil) bereits bekannt sein. **Bauen Sie das Ventil entsprechend gerichtet ein**.

Die Strömungsrichtung ergibt sich während der Simulation aus den Druckverhältnissen.

Die Stellung des Ventils wird am Eingang *Position* vorgegeben. Dieser Wert bezieht sich auf den Ventilzweig *AB–A*. Die Öffnung des Ventilzweiges *AB–B* ergibt sich aus der Differenz der Öffnung des Ventilzweiges *AB–A* zu 100 %.

Der Druckabfall über dem Ventil wird gemäß folgender Formel abhängig von der Ventilstellung, der parametrierten Kennlinie und der Ventilgröße (k_{VS}-Wert) berechnet:

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_{VS}^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

Diese Berechnung wird für beide Ventilzweige durchgeführt. Dabei gilt beispielsweise für den Zweig *AB–A*:

$\Delta p = p_{AB} - p_A$	der Druckabfall über dem Ventil in bar (*AB–A*)
$\dot{m}_{AB–A}$	der Durchfluss bzw. Massenstrom von *AB* nach *A* in kg/s
ρ	die Dichte des Mediums in kg/m³
$k_{V\,AB–A}$	der Ventilkennwert in m³/h

Die Bezugsrichtung für den Medienfluss $\dot{m}$ ist vom Anschluss *AB* zum Anschluss *A* bzw. *B*. Für einen Medienfluss in Bezugsrichtung gilt somit $\dot{m} > 0$ (**Verteilventil**). Die entsprechenden Druckabfälle haben damit dann einen negativen Wert: $\Delta p < 0$.

Wenn die Strömungsrichtung umgekehrt wird, ergibt sich ein **Mischventil**. Hierbei wird der Massenstrom entgegen der Bezugsrichtung berechnet: $\dot{m} > 0$ (von den Anschlüssen *A* und *B* nach *AB*). Die entsprechenden Druckabfälle haben dann einen positiven Wert: $\Delta p > 0$ (Zählrichtung beachten!).

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Characteristic	Auswahl der Ventilcharakteristik	–	*Linear*
Kvs	Ventilkennwert k_{VS} mit $k_{VS} \geq k_{V0} + 10^{-6}$ m³/h	m³/h	10.0
OpeningCharacteristic	Vorgabe der Ventilcharakteristik als Kennlinie. Dieser Parameter wird nur verwendet, wenn für den Parameter *Characteristic* der Wert *Polygon* ausgewählt wurde. Hierbei wird die prozentuale Öffnung in Abhängigkeit vom prozentualen Hub angegeben.	–	

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m³/h Simulationstechnisch stellt das Dreiwegeventil im Flussnetz einen internen Knoten dar. Da sich die Größe dieses Knotens von der Größe anderer Komponenten deutlich unterscheiden kann, wird hier die Möglichkeit zur Parametrierung dieses Knotens gegeben.	m³/h	0.000001
sCompressionGas	Spezifischer Kompressionsmodul für Medien niedriger Dichte (Gase/Dämpfe, $\rho < 500$ kg/m³)	bar/kg	10.0

Parametername	Beschreibung	Einheit	Standardwert
sCompressionLiquid	Spezifischer Kompressionsmodul für Medien hoher Dichte (Flüssigkeiten, ρ > 500 kg/m³)	bar/kg	100.0
FactorThermalGas	Faktor für die Enthalpiebilanzierung für Medien niedriger Dichte (Gase/Dämpfe, ρ < 500 kg/m³)	1/kg	1000.0
FactorThermalLiquid	Faktor für die Enthalpiebilanzierung für Medien hoher Dichte (Flüssigkeiten, ρ > 500 kg/m³)	1/kg	10.0
PressureInit	Initialisierungswert für den Druck im Knoten	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie im Knoten	kJ/(kgK)	100.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums im Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

Bedienfenster

Im Bedienfenster können Sie zwischen automatisierter Fahrweise (Steuerung über die Aktor-Sensor-Ebene) und manueller Fahrweise umschalten. Voreingestellt ist immer die automatisierte Fahrweise.

Bei manueller Fahrweise können Sie die Ventilstellung über einen Slider vorgeben. Der Wert des Sliders bezieht sich hierbei auf den Zweig *AB–A*.

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

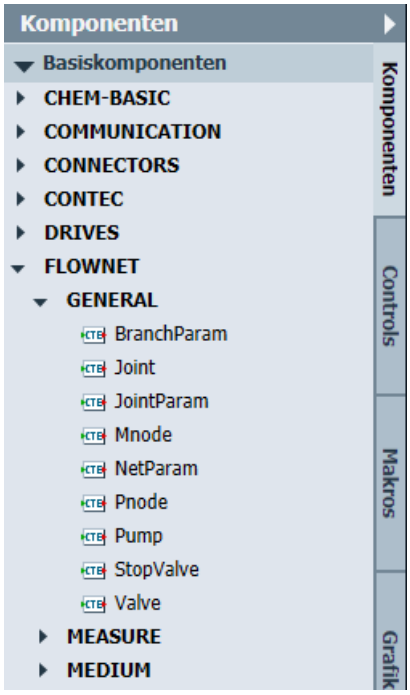
Größe	Formelzeichen	Einheit
Druckunterschied	Δp_1	bar
Druckunterschied	Δp_2	bar
Massenstrom	$\dot{m}$	kg/s

8.2.6 Komponenten der Bibliothek FLOWNET

8.2.6.1 Allgemeine Komponenten

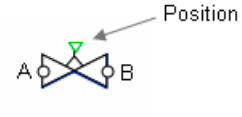
Auswahl der allgemeinen Komponenten

Im Verzeichnis *GENERAL* der Bibliothek FLOWNET stehen Komponententypen zur Verfügung, die in Flussnetzen mit beliebigen Medien einsetzbar sind.



Valve – Ventil

Symbol



Funktion

Der Komponententyp *Valve* dient zur Simulation eines Stellventils. Abhängig von der Ventilstellung wird der Druckabfall über dem Ventil gemäß

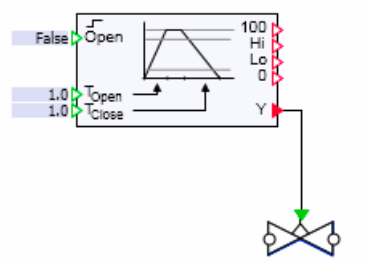
$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

berechnet. Dabei gilt Folgendes:

- $\Delta p = p_B - p_A$ der Druckabfall über dem Ventil in bar
- $\dot{m}$ der Durchfluss bzw. Massenstrom in kg/s
- ρ die Dichte des Mediums in kg/m³
- k_V der Ventilkennwert in m³/h

Für den Medienfluss $\dot{m}$ ist die Bezugsrichtung vom Anschluss *A* zum Anschluss *B* definiert, d. h. für einen Medienfluss in Bezugsrichtung gilt $\dot{m} > 0$. Der Druckabfall hat damit dann einen negativen Wert: $\Delta p < 0$.

Am Anschluss *Position* wird die Stellung *H* des Ventilantriebs als prozentualer Wert vorgegeben, wozu beispielsweise die Antriebskomponententypen der Basisbibliothek von SIMIT eingesetzt werden können.



Der Stellungswert wird begrenzt auf den Bereich $0 \le H \le 100\ \%$. Die Ventilstellung wird über die Ventilkennlinie auf den Ventilkennwert k_V abgebildet. Dabei gilt

$$\frac{k_V}{k_{V100}} = \frac{1}{S_V} + \left(1 - \frac{1}{S_V}\right) \frac{H}{100\%}$$

für eine lineare Kennlinie,

$$\frac{k_V}{k_{V100}} = \frac{1}{S_V} + \left(1 - \frac{1}{S_V}\right) \left(\frac{H}{100\%}\right)^2$$

für eine quadratische Kennlinie,

$$\frac{k_V}{k_{V100}} = S_V^{\frac{H}{100\%} - 1}$$

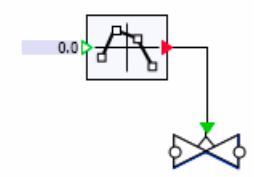
für eine gleichprozentige Kennlinie.

Das Stellverhältnis

$$S_V = \frac{k_{V100}}{k_{V0}}$$

ist hier als Quotient aus dem Ventilkennwert *kV100* bei vollständig geöffnetem Ventil (*H = 100 %*) und dem Ventilkennwert k_{V0} bei vollständig geschlossenem Ventil (*H = 0*) angesetzt.

Beliebige Ventilkennlinien lassen sich beispielsweise mit dem Komponententyp *Characteristic* aus der Basisbibliothek von SIMIT realisieren. Die Kennlinienkomponente wird dazu einfach dem Eingang *Position* des Ventils vorgeschaltet.



Am Ventil wird dann die lineare Kennlinie eingestellt, so dass die mit der Kennlinienkomponente vorgegebene Kennlinie lediglich mit dem Faktor $(k_{V100} - k_{V0}) / k_{V0}$ skaliert und um k_{V0} verschoben wird.

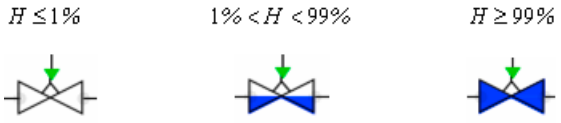
Parameter

Parametername	Beschreibung	Einheit	Standardwert
Characteristic	Eingestellte Ventilkennlinie: *Linear*, *Quadratic*, *EqualPercentage*; online änderbar	–	*Linear*
Kvs	Ventilkennwert k_{V100} mit $k_{V100} \geq k_{V0} + 10^{-6}$ m³/h; online änderbar	m³/h	360.0
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}$ m³/h	m³/h	0.000001

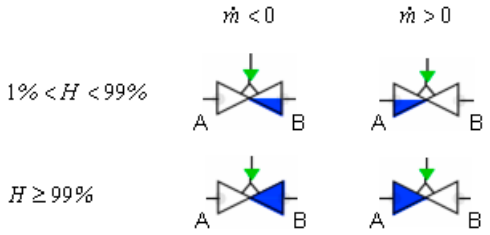
Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
ShowFlow	Visualisierung der Ventilstellung im Symbol	–	*False*
ShowFlowDirection	Visualisierung der Richtung des Durchflusses im Symbol	–	*False*

Wenn *ShowFlow* auf *True* gesetzt ist, wird die Ventilstellung angezeigt wie in der folgenden Abbildung dargestellt.



Ist zusätzlich *ShowFlowDirection* auf *True* gesetzt, werden die Ventilstellung und die Richtung des Durchflusses angezeigt wie in der folgenden Abbildung dargestellt.



Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckabfall	Δp	bar
Massenstrom	$\dot{m}$	kg/s

StopValve – Rückschlagventil

Symbol



Funktion

Der Komponententyp *StopValve* dient zur Simulation eines Rückschlagventils. Abhängig von der Richtung des Durchflusses wird der Druckabfall über dem Rückschlagventil gemäß

$$\frac{\Delta p}{\text{bar}} = \begin{cases} \dfrac{-\dot{m}^2}{k_{V100}^2\,\rho\dfrac{\text{kg}}{\text{m}^3}}\,12960\left(\dfrac{\text{sec}}{\text{h}}\right)^2 & \text{für} \quad \dot{m} > 0 \\ \dfrac{-\dot{m}^2}{k_{V0}^2\,\rho\dfrac{\text{kg}}{\text{m}^3}}\,12960\left(\dfrac{\text{sec}}{\text{h}}\right)^2 & \text{für} \quad \dot{m} < 0 \end{cases}$$

berechnet. Dabei gilt folgendes:

- $\Delta p = p_B - p_A$ der Druckabfall über dem Rückschlagventil in bar,
- $\dot{m}$ der Durchfluss bzw. Massenstrom in kg/s,
- ρ die Dichte des Mediums in kg/m³ und
- k_V der Ventilkennwert in m³/h.

Für den Medienfluss $\dot{m}$ ist die Bezugsrichtung vom Anschluss *A* zum Anschluss *B* definiert, d. h. für einen Medienfluss in Bezugsrichtung gilt $\dot{m} > 0$.

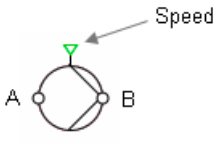
Parameter

Parametername	Beschreibung	Einheit	Standardwert
Kvs	Ventilkennwert k_{V100} mit $k_{V100} \geq k_{V0} + 10^{-6}\ m^3/h$	m^3/h	360.0
Kv0	Ventilkennwert k_{V0} mit $k_{V0} \geq 10^{-6}\ m^3/h$	m^3/h	0.000001

Zur Realisierung der Rückschlagfunktion ist der Ventilkennwert k_{V0} genügend klein zu wählen.

Pump – Pumpe

Symbol



Funktion

Der Komponententyp *Pump* berechnet die Druckerhöhung abhängig vom Förderstrom und der Drehzahl gemäß

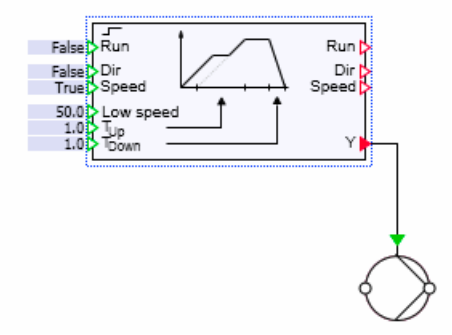
$$\Delta p = n^2 \Delta p_0 + \left(\Delta p^* - \Delta p_0\right) \frac{\dot{m}^2}{\left(\dot{m}^*\right)^2} \quad \text{für} \quad \dot{m} > 0$$

Dabei ist

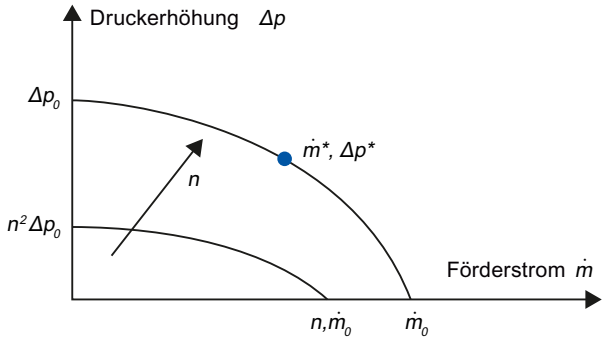
$\Delta p = p_B - p_A$ die Druckerhöhung in bar,
$\dot{m}$ der Förderstrom, Massenstrom in kg/s,
Δp_0 die Nullförderhöhe in bar,
Δp^* die Nenndruckerhöhung in bar,
$\dot{m}^*$ der Nennförderstrom in kg/s und
n der dimensionslose Drehzahlwert.

Für den Förderstrom $\dot{m}$ ist die Bezugsrichtung vom Anschluss *A* zum Anschluss *B* definiert, d. h. für einen Förderstrom in Bezugsrichtung gilt $\dot{m} > 0$.

Die Drehzahl *N* wird als prozentualer Wert am Eingang *Speed* vorgegeben. Der Drehzahlwert wird begrenzt auf den Bereich $0 \leq N \leq 100$ %. Folglich gilt $n = N / 100$ % und damit $0 \leq n \leq 1$. Verwendet werden können hier beispielsweise die Antriebe aus der Basisbibliothek von SIMIT wie in der folgenden Abbildung.



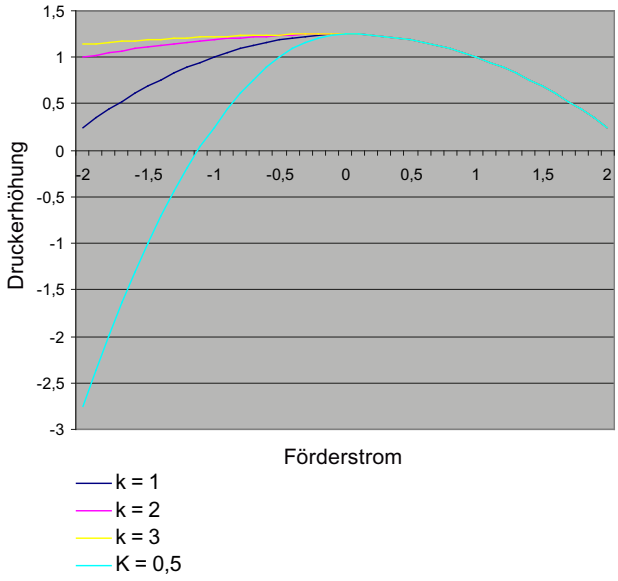
Der oben definierte quadratische Zusammenhang zwischen Druckerhöhung und Förderstrom ist für den Betrieb der Pumpe im Normalbereich, d. h. für $\dot{m} > 0$, $\Delta p > 0$ in der folgenden Abbildung veranschaulicht.



Die Pumpenkennlinie wird gemäß

$$\Delta p = n^2 \Delta p_0 + \left(\Delta p^* - \Delta p_0\right) \frac{\dot{m}^2}{\left(k\dot{m}^*\right)^2} \quad \text{für} \quad \dot{m} < 0$$

für den Fall von Flussumkehr ($\dot{m} < 0$) stetig erweitert. Über den Durchflusskennwert k kann die Drosselwirkung beeinflusst werden. Folgende Abbildung zeigt die erweiterte Kennlinie qualitativ für verschiedene Kennwerte k:



Offensichtlich wird die Drosselwirkung für kleinere Durchflusskennwerte *k* größer.

Zur Stabilisierung einer Simulation, in der eine Komponente dieses Typs enthalten ist, wird mit den oben beschriebenen Gleichungen auch im Falle von Druckumkehr ($\Delta p < 0$) gerechnet. Für die ausgeschaltete Pumpe, d. h. für Drehzahl null, ergibt sich dann eine reine Drosselwirkung für einen Fluss sowohl in als auch gegen die Pumprichtung:

$$\Delta p = \begin{cases} \left(\Delta p^* - \Delta p_0\right)\dfrac{\dot{m}^2}{\left(k\dot{m}^*\right)^2} & \text{für} \quad \dot{m} < 0 \\ \left(\Delta p^* - \Delta p_0\right)\dfrac{\dot{m}^2}{\left(\dot{m}^*\right)^2} & \text{für} \quad \dot{m} > 0 \end{cases}$$

Parameter

Parametername	Beschreibung	Einheit	Standardwert
ZeroFlowHead	Nullförderhöhe Δp_0, $\Delta p_0 > \Delta p^*$; online änderbar	bar	10.0
NominalPressure	Nenndruck Δp^*, $\Delta p^* > 0$; online änderbar	bar	8.0
NominalMassflow	Nennmassenstrom $\dot{m}^*$, $\dot{m}^* > 0$; online änderbar	kg/s	10.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
Throttling	positiver Durchflusskennwert *k (k > 0)*	–	1.0
ShowFlow	Visualisierung des Betriebszustands im Symbol	–	*False*

Falls der Zusatzparameter *Showflow* auf True gesetzt ist, wird der Betriebszustand im Symbol der Pumpe angezeigt.



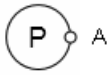
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druckerhöhung	Δp	bar
Förderstrom	$\dot{m}$	kg/s

Pnode – Druckvorgabe

Symbol



Funktion

Der Komponententyp *Pnode* gibt Werte für Druck *p* und spezifische Enthalpie *h* an seinem Anschluss A vor. Eine Komponente dieses Typs bildet einen Rand für das Flussnetz. In der Betrachtung des Flussnetzes als Graph entspricht das einem (externen) Knoten, für den der Druck und die spezifische Enthalpie fest vorgegeben sind.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Druck (Vorbelegung: 0)	p	bar	Ja
spezifische Enthalpie (Vorbelegung: 100)	h	kJ/kg	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s
Druckabfall des Kondensats	Δp	bar
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

Für Ausströmung ist $\dot{m} > 0$, für Einströmung ist $\dot{m} < 0$.

Mnode – Massenstromvorgabe

Symbol



Funktion

Der Komponententyp *Mnode* gibt Werte für Massenstrom $\dot{m}$ und spezifische Enthalpie *h* an seinem Anschluss A vor. Eine Komponente dieses Typs bildet damit einen Rand für das Flussnetz. In der Betrachtung des Flussnetzes als Graph entspricht das einer Ein- oder Ausströmung in einen (internen) Knoten oder Zweig. Dazu wird für jede Komponente dieses Typs ein interner Knoten mit der definierten Ein- oder Ausströmung in das Flussnetz eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Massenstrom (Vorbelegung: 0)	$\dot{m}$	kg/s	Ja
spezifische Enthalpie (Vorbelegung: 100)	h	kJ/kg	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

NetParam – Netzparametrierung

Symbol



Funktion

Der Komponententyp *NetParam* dient zur Parametrierung eines Flussnetzes. Dazu wird die Komponente an beliebiger Stelle in einen beliebigen Zweig des Flussnetzes eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Medium	Auswählt werden kann "*Water/Steam*" für Wasser/Dampf, "*Liquid*" für Flüssigkeit oder "*Ideal Gas*" für ideales Gas als Medium im Flussnetz	–	*Water/Steam*
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0
sCompressionGas	spezifisches Kompressionsmodul für Medium "*Water/Steam*" mit Dichten ρ < 500 kg/m³ oder Medium "*Ideal Gas*"	bar/kg	10.0
sCompressionLiquid	spezifisches Kompressionsmodul für Medium "*Water/Steam*" mit Dichten ρ > 500 kg/m³ oder Medium "*Liquid*"	bar/kg	100.0
FactorThermalGas	Faktor für die Enthalpiebilanzierung Medium "*Water/Steam*" mit Dichten ρ < 500 kg/m³ oder Medium "*Ideal Gas*"	1/kg	100.0
FactorThermalLiquid	Faktor für die Enthalpiebilanzierung Medium "*Water/Steam*" mit Dichten ρ > 500 kg/m³ oder Medium "*Liquid*"	1/kg	0.1
DensityLiquid	Mediendichte; nur gültig für Medium "*Liquid*"	kg/m³	997.337
sHeatCapGas	spezifische Wärmekapazität für Gas; nur gültig für Medium "*Ideal Gas*"	kJ/kgK	1.0
sHeatCapLiquid	spezifische Wärmekapazität für Flüssigkeiten; nur gültig für Medium "*Liquid*"	kJ/kgK	4.18
GasConstant	spezifische Gaskonstante des Mediums; nur gültig für Medium "*Ideal Gas*"	kJ/kgK	0.287

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck in den internen Knoten des Flussnetzes	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie in den internen Knoten des Flussnetzes	kJ/kg	100.0
SmoothTransition	Wenn dieser Zusatzparameter auf True gesetzt ist, werden die Größen *sCompression* und *FactorThermal* mit einer dichteabhängigen linearen Übergangsfunktion angesetzt	–	False
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums in den internen Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

BranchParam – Zweigparametrierung

Symbol



Funktion

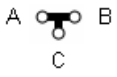
Der Komponententyp *BranchParam* dient zur Parametrierung eines Zweiges im Flussnetze. Dazu wird die Komponente an beliebiger Stelle im zu parametrierenden Zweig eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0

Joint – Verzweigung

Symbol



Funktion

Über den Komponententyp *Joint* können drei an seinen Anschlüssen *A*, *B* und *C* angeschlossene Zweige in einem Knoten zusammengeführt werden. Mit der Komponente *Joint* wird ein interner Knoten in das Flussnetz eingefügt.

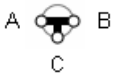
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³
Temperatur	T	°C

JointParam – parametrierbare Verzweigung

Symbol



Funktion

Über den Komponententyp *JointParam* können drei an seinen Anschlüssen *A*, *B* und *C* angeschlossene Zweige in einem Knoten zusammengeführt werden. Mit der Komponente *JointParam* wird ein interner Knoten in das Flussnetz eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
sCompressionGas	spezifisches Kompressionsmodul für Medien niedriger Dichte (Gase/Dämpfe, ρ < 500 kg/m³)	bar/kg	10.0
sCompressionLiquid	spezifisches Kompressionsmodul für Medien hoher Dichte (Flüssigkeiten, ρ > 500 kg/m³)	bar/kg	100.0
FactorThermalGas	Faktor für die Enthalpiebilanzierung für Medien niedriger Dichte (Gase/Dämpfe, ρ < 500 kg/m³)	1/kg	100.0
FactorThermalLiquid	Faktor für die Enthalpiebilanzierung für Medien hoher Dichte (Flüssigkeiten, ρ > 500 kg/m³)	1/kg	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck im Knoten	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie im Knoten	kJ/kg	100.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums im Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

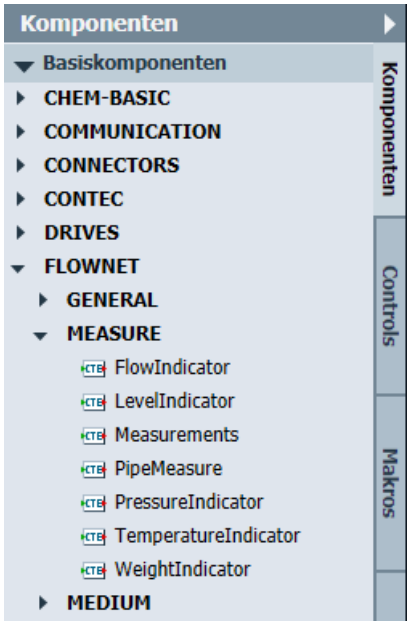
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³
Temperatur	T	°C

8.2.6.2 Messkomponenten

Komponenten zur Simulation von Messungen in Rohrleitungsnetzen werden im Verzeichnis *MEASURE* zur Verfügung gestellt.



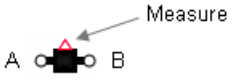
In den Messkomponenten werden Anschlüsse vom Verbindungstyp *Measure* eingesetzt. Die Bedeutung der einzelnen Signale dieses Typs ist in der folgenden Tabelle erläutert.

Tabelle 8-43 Signale des Verbindungstyps Measure

Signal	Bedeutung	Einheit
Temperature	Temperatur der Messung	°C
Pressure	Druck der Messung	bar
Level	gemessenes Niveau	m
Weight	gemessenes Gewicht	kg
Flow	gemessener Durchfluss	kg/s

PipeMeasure – Rohrmessstelle

Symbol



Funktion

Der Komponententyp *PipeMeasure* bildet eine Mess-Stelle im Rohr. Es wird mit seinen Anschlüssen *A* und *B* an der gewünschten Messstelle im Flussnetz eingefügt. Die Messverfahren für die verschiedenen Größen werden hierbei nicht mit geeigneten Modellen simuliert, es werden die im Flussnetzlöser berechneten Größen ausgegeben.

Über den Anschluss *Measure* werden die Messgrößen ausgegeben:

- Absolutwert $|\dot{m}|$ des Durchflusses,
- Druck p_A am Anschluss *A* und
- Temperatur *T*.

Alle anderen Signale des Anschlusses *Measure* werden nicht gesetzt.

Die Richtung des Medienflusses wird bei laufender Simulation durch einen Pfeil im Symbol angezeigt.



Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

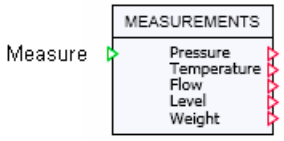
Größe	Formelzeichen	Einheit
Druck	p	bar
Massenstrom	$\dot{m}$	kg/s
Temperatur	T	°C

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³

Measurements – Messanzeigen

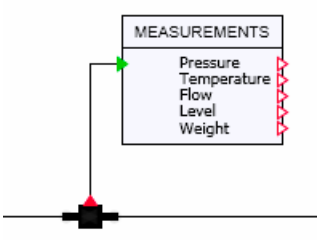
Symbol



Funktion

Der Komponententyp *MeasureAll* stellt an seinen Ausgängen die über den Messeingang *Measure* gebündelt aufgenommen Messwerte als Einzelsignale zur Verfügung: Druck (Pressure), Temperatur (Temperature), Durchfluss (Flow), Niveau (Level) und Gewicht (Weight).

Eine Komponente dieses Typs kann beispielsweise an die Rohrmessstelle angeschlossen werden und so deren Messgrößen als Einzelsignale an seinen Ausgängen zur Verfügung stellen.



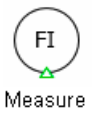
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C
Massenstrom	$\dot{m}$	kg/s
Niveau	l	m
Gewicht	M	kg

FlowIndicator – Messanzeige für Durchfluss

Symbol



Funktion

Der Komponententyp *FlowIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Durchflusses.

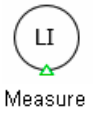
Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s

LevelIndicator – Niveaumessanzeige

Symbol



Funktion

Der Komponententyp *LevelIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Niveaus.

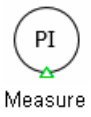
Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Niveau	l	m

PressureIndicator – Druckmessanzeige

Symbol



Funktion

Der Komponententyp *PressureIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Drucks.

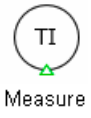
Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar

TemperatureIndicator – Temperaturmessanzeige

Symbol



Funktion

Der Komponententyp *TemperatureIndicator* dient zur Anzeige der über seinen Eingang *Measure* vorgegebenen Temperatur.

Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Temperatur	T	°C

WeightIndicator – Gewichtsmessanzeige

Symbol



Funktion

Der Komponententyp *WeightIndicator* dient zur Anzeige des über seinen Eingang *Measure* vorgegebenen Gewichtswerts.

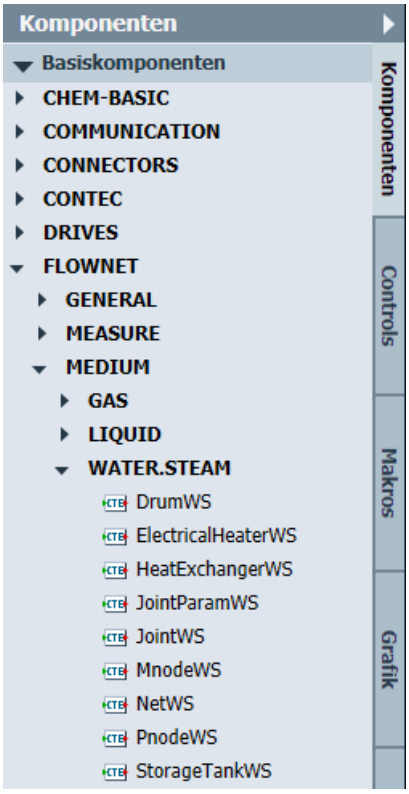
Bedienfenster

Im Bedienfenster wird folgende Größe angezeigt:

Größe	Formelzeichen	Einheit
Gewicht	M	kg

8.2.6.3 Komponententypen für Medium "Wasser/Dampf"

Im Ordner *MEDIUM\WATER.STEAM* der Bibliothek FLOWNET sind Komponententypen abgelegt, die in Flussnetzen mit dem Medium Wasser/Dampf eingesetzt werden können. Der Parameter Medium für ein Flussnetz, das diese Komponenten enthält ist auf den Wert "*Water/Steam*" zu setzen.



NetWS – Netzparametrierung

Symbol



Funktion

Der Komponententyp *NetWS* dient zur Parametrierung eines Netzes für das Medium Wasser/Dampf. Dazu wird die Komponente an beliebiger Stelle in einen beliebigen Zweig des Flussnetzes eingefügt.

Parameter

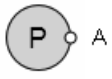
Parametername	Beschreibung	Einheit	Standardwert
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0
sCompressionSteam	spezifisches Kompressionsmodul für Dichten ρ < 500 kg/m³ (Dampf)	bar/kg	10.0
sCompressionWater	spezifisches Kompressionsmodul für Dichten ρ > 500 kg/m³ (Wasser)	bar/kg	100.0
FactorThermalSteam	Faktor für die Enthalpiebilanzierung für Dichten ρ < 500 kg/m³ (Dampf)	1/kg	100.0
FactorThermalWater	Faktor für die Enthalpiebilanzierung für Dichten ρ > 500 kg/m³ (Wasser)	1/kg	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck in den internen Knoten des Flussnetzes	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie in den internen Knoten des Flussnetzes	kJ/kg	100.0
SmoothTransition	Wenn dieser Zusatzparameter auf True gesetzt ist, werden die Größen *sCompression* und *FactorThermal* mit einer dichteabhängigen linearen Übergangsfunktion angesetzt	–	False
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums in den internen Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

PnodeWS – Druckvorgabe

Symbol



Funktion

Der Komponententyp *PnodeWS* gibt Werte für Druck *p* und spezifische Enthalpie *h* an seinem Anschluss A vor. Eine Komponente dieses Typs bildet einen Rand für das Flussnetz. In der Betrachtung des Flussnetzes als Graph entspricht das einem (externen) Knoten, für den der Druck und die spezifische Enthalpie fest vorgegeben sind.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Druck (Vorbelegung: 1)	p	bar	Ja
spezifische Enthalpie (Vorbelegung: 100)	h	kJ/kg	Ja

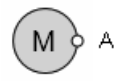
Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

Für Ausströmung ist $\dot{m} > 0$, für Einströmung ist $\dot{m} < 0$.

MnodeWS – Massenstromvorgabe

Symbol



Funktion

Der Komponententyp *MnodeWS* gibt Werte für Massenstrom $\dot{m}$ und spezifische Enthalpie *h* an seinem Anschluss A vor. Eine Komponente dieses Typs bildet damit einen Rand für das Flussnetz. In der Betrachtung des Flussnetzes als Graph entspricht das einer Ein- oder Ausströmung in einen (internen) Knoten oder Zweig. Dazu wird für jede Komponente dieses Typs ein interner Knoten mit der definierten Ein- oder Ausströmung in das Flussnetz eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Massenstrom (Vorbelegung: 0)	$\dot{m}$	kg/s	Ja
spezifische Enthalpie (Vorbelegung: 100)	h	kJ/kg	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Dichte	r	kg/m³
Temperatur	T	°C
spezifische Enthalpie	h	kJ/kg

JointWS – Verzweigung

Symbol



Funktion

Über den Komponententyp *JointWS* können drei an seinen Anschlüssen *A*, *B* und *C* angeschlossene Zweige in einem Knoten zusammengeführt werden. Mit der Komponente *JointWS* wird ein interner Knoten in das Flussnetz eingefügt.

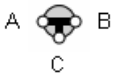
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³
Temperatur	T	°C

JointParamWS – parametrierbare Verzweigung

Symbol



Funktion

Über den Komponententyp *JointParamWS* können drei an seinen Anschlüssen *A*, *B* und *C* angeschlossene Zweige in einem Knoten zusammengeführt werden. Mit der Komponente *JointParamWS* wird ein interner Knoten in das Flussnetz eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
sCompressionSteam	spezifisches Kompressionsmodul für Dichten ρ < 500 kg/m³ (Dampf)	bar/kg	10.0
sCompressionWater	spezifisches Kompressionsmodul für Dichten ρ > 500 kg/m³ (Wasser)	bar/kg	100.0
FactorThermalSteam	Faktor für die Enthalpiebilanzierung für Dichten ρ < 500 kg/m³ (Dampf)	1/kg	100.0
FactorThermalWater	Faktor für die Enthalpiebilanzierung für Dichten ρ > 500 kg/m³ (Wasser)	1/kg	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck im Knoten	bar	1.0
sEnthalpyInit	Initialisierungswert für die spezifische Enthalpie im Knoten	kJ/kg	100.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums im Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

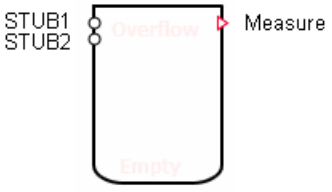
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
spezifische Enthalpie	h	kJ/kg
Dichte	r	kg/m³
Temperatur	T	°C

StorageTankWS – Speicherbehälter

Symbol



Funktion

Der Komponententyp *StorageTankWS* stellt die Simulation eines offenen, d. h. nicht gegen die Umgebung abgeschlossenen Behälters für Wasser zur Verfügung.

Zu- und Abflüsse erfolgen über Anschlüsse *STUBx* am Behälter. Für jeden der *N* Anschlüsse wird eine Drosselwirkung gemäß

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

angenommen. Der Behälter kann minimal einen, maximal 16 Anschlüsse haben. Die Anschlüsse können mit der Maus bei gedrückter ALT-Taste beliebig auf dem Umriss des Komponentensymbols verschoben werden.

Es wird eine unmittelbare vollständige Durchmischung des Wassers im Behälter angenommen, d. h. ein homogenes Medium im Behälter mit überall gleicher Dichte und Enthalpie.

Die Zu- und Abflüsse des Wassers werden über die *N* Anschlüsse des Behälters bilanziert. Man erhält so die über die *N* Massenströme $\dot{m}_i$ bilanzierte Masse *M* des Mediums im Behälter gemäß

$$\frac{dM}{dt} = \sum_{i=1}^{N} \dot{m}_i$$

Aus der Bilanzierung des Wasservolumens über die Volumenströme resultiert die Änderung seiner Dichte ρ gemäß

$$\frac{d\rho}{dt} = \frac{\rho}{M} \left(\sum_{i=1, i \in Z}^{N} \dot{m}_i - \rho \sum_{i=1, i \in Z}^{N} \frac{1}{\rho_i} \dot{m}_i \right)$$

wobei über die Zuflüsse ($i \in Z$) summiert werden muss.

Die spezifische Enthalpie des Wassers wird aus der Enthalpiebilanz gemäß

$$\frac{dh}{dt} = \frac{1}{M} \left(\sum_{i=1, i \in Z}^{N} h_i \dot{m}_i - h \sum_{i=1, i \in Z}^{N} \frac{1}{\rho_i} \dot{m}_i \right)$$

ermittelt. Auch hier muss lediglich über die Zuflüsse ($i \in Z$) summiert werden.

Mit diesen Bilanzen für Masse *M*, Dichte ρ und spezifische Enthalpie *h* ist das dynamische Verhalten des Wassers im Behälter beschrieben.

Am Anschluss *Measure* werden die berechneten Größen für das Niveau *l* des Wassers im Behälter, seine Masse *M* und Temperatur *T* sowie der Druck p_0 am Behälterboden ausgegeben. Die Temperatur *T* wird über die Zustandsgleichung

$T = T(\rho, h)$

aus der Dichte und spezifischen Enthalpie des Wassers berechnet. Der Druck *p* am Behälterboden bildet sich aus dem Gewichtsdruck ρ*gl* des Wassers und dem Umgebungsdruck p_U zu

$p = p_U + \rho gl.$

Grenzfall "Behälter leer"

Für einen leeren Behälter wird der Abfluss stark gedrosselt. Der Ventilkennwert wird dazu für alle Anschlüsse über die Wasser abfließt, auf den Wert k_{V0} für maximale Drosselung gesetzt.

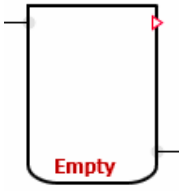
Der Behälter wird als leer angenommen, wenn die Füllung V kleiner als eine gegebene minimale Füllung V_{min} wird:

$V < V_{min}$

Die Bilanzierung der Zustände wird für einen leeren Behälter gestoppt, d. h. Änderungen der drei Zustandsgrößen Masse, Dichte und spezifische Enthalpie werden verworfen. Der Zustand leer wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Füllung vorliegt. In jedem Zeitschritt wird dazu geprüft, ob

$$M \geq V_{min}\,\rho\left(1+\frac{H_{min}}{100\%}\right)$$

gilt. Der erforderliche Zuwachs kann über die Hysterese H_{min} gesetzt werden. Im Symbol einer Komponente wird für einen leeren Behälter ein entsprechender Hinweis eingeblendet.

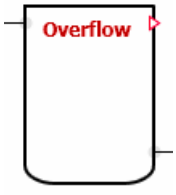


Grenzfall "Behälter voll"

Für einen vollen Behälter wird in der Berechnung der Zustandsgrößen die bilanzierte Füllung begrenzt:

$M = V\rho$

Es wird ein entsprechender Hinweis im Symbol eingeblendet.



Hinweis

Es wird nicht überprüft, ob die Zustandsgrößen (Druck, Enthalpie) der Zuflüsse wie auch des Behälterinhalts immer Werte für den Zustand Wasser aufweisen.

Durch entsprechend gesetzte Zuflüsse können in der Simulation auch (physikalisch nicht sinnvolle) Behälterfüllungen auftreten, deren Zustandswerte ein Wasser-Dampf-Gemisch oder reinen Dampf beschreiben.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Behälters; online änderbar	m^3	10.0
Height	Höhe des Behälters; online änderbar	m	5.0
NbrOfStubs	Anzahl *N* der Anschlüsse	–	1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureOutside	Umgebungsdruck p_U; online änderbar	bar	1.0
LevelInit	Initialisierungswert für das Niveau	%	50.0
EnthalpyInit	Initialisierungswert für die spezifische Enthalpie der Füllung	kJ/kg	100.0
DensityInit	Initialisierungswert für die Dichte der Füllung	kg/m^3	997.337
Kvs	für alle Anschlüsse einheitlicher Ventilkennwert k_V	m^3/h	360.0
Kv0	Ventilkennwert k_{V0} für starke Drosselung im Behälteranschluss	m^3/h	0.000001
MinVolume	*Minimales V*olumen V_{min} des Behälters; online änderbar	m^3	0.001
MinVolumeHys	Hysterese H_{min}; online änderbar	%	50.0

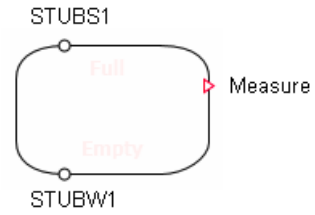
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Niveau	l	m
Druck	p	bar
Temperatur	T	°C
Gewicht	M	kg

DrumWS – Trommel

Symbol



Funktion

Der Komponententyp *DrumWS* stellt die Simulation einer Trommel, d. h. eines geschlossenen, gegen die Umgebung abgeschlossenen Behälters für Wasser/Dampf zur Verfügung.

Die Trommel wird als zylindrischer, liegender oder stehender Behälter angenommen, dessen Zu- und Abflüsse über Anschlüsse *STUBWx* bzw. *STUBSx* erfolgen. Für jeden der Anschlüsse wird eine Drosselwirkung gemäß

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12960 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

angenommen. An den N_W Anschlüssen *STUBWx* kann Medium mit den Zustandsgrößen des gesättigten Wassers entnommen werden, über die N_S Anschlüsse *STUBSx* Medium mit den Zustandsgrößen des gesättigten Dampfes. Die Trommel kann minimal einen, maximal 8 Anschlüsse eines jeden Typs haben. Die Anschlüsse können mit der Maus bei gedrückter ALT-Taste beliebig auf dem Umriss des Komponentensymbols verschoben werden.

In der Trommel wird eine unmittelbare Trennung des Mediums in zwei homogene gesättigte Phasen angenommen: die gesättigte flüssige Phase und die gesättigte dampfförmige Phase.

Die Zu- und Abflüsse von Wasser/Dampf werden über die $N = N_W + N_S$ Anschlüsse der Trommel bilanziert. Man erhält so die über die Massenströme bilanzierte Masse von Wasser und Dampf in der Trommel gemäß

$$\frac{dM}{dt} = \sum_{i=1}^{N} \dot{m}_i$$

bzw. wegen

$M = \rho V$

die Änderung der mittleren Dichte zu

$$V \frac{d\rho}{dt} = \sum_{i=1}^{N} \dot{m}_i$$

Weiter wird die Energie bilanziert gemäß

$$\frac{d(hM)}{dt} = \sum_{i=1}^{N} h_i \dot{m}_i$$

Dabei ist h die spezifische Enthalpie (mittlere Enthalpie) von Wasser/Dampf in der Trommel und h_i ist die spezifische Enthalpie des Zu- bzw. Abflusses am i-ten Anschluss. Für Abflüsse ist die spezifische Enthalpie des Mediums in der Trommel anzusetzen, also h' für Abflüsse von gesättigtem Wasser, h'' für Abflüsse von gesättigtem Dampf. Die Bilanzierung ergibt sich damit gemäß

$$M \frac{dh}{dt} = \sum_{i=1, i\in Z}^{N} (h_i - h)\dot{m}_i + (h' - h) \sum_{i=1, i\in A'}^{N} \dot{m}_i + (h'' - h) \sum_{i=1, i\in A''}^{N} \dot{m}_i$$

Dabei wird über die Zuflüsse Z, Abflüsse von gesättigtem Wasser A' bzw. Abflüsse von gesättigtem Dampf A'' summiert.

Um einen Wärmeaustausch mit der ideal isolierten Trommel zu modellieren, wird die Enthapiebilanz ergänzt gemäß

$$M \frac{dh}{dt} = \sum_{i=1, i\in Z}^{N} (h_i - h)\dot{m}_i + (h' - h) \sum_{i=1, i\in A'}^{N} \dot{m}_i + (h'' - h) \sum_{i=1, i\in A''}^{N} \dot{m}_i + A\alpha (T_T - T_S)$$

T_T ist die Temperatur der Trommelwand, T_S die Sättigungstemperatur von Wasser/Dampf.

Die Wärmeeinspeicherung in der Trommelwand wird mit der Wärmebilanz

$$\frac{dT_T}{dt} = \frac{1}{M_T c_T} A\alpha (T_S - T_T)$$

beschrieben.

Am Anschluss *Measure* werden das Niveau des Wassers l, Sättigungstemperatur T_S und Sättigungsdruck p_S und die Masse M von Wasser/Dampf ausgegeben.

Die Anschlüsse STUBWx werden als unten an der Trommel liegend angenommen. Der Druck an diesen Anschlüssen setzt sich damit gemäß

$p = p'gl + p_S$

aus dem Sättigungsdruck und dem Schweredruck des Wassers zusammen.

Grenzfall "Trommel leer"

Für eine leere Trommel wird der Abfluss stark gedrosselt. Der Ventilkennwert wird dazu für alle Anschlüsse über die Wasser oder Dampf abfließt, auf den Wert k_{V0} für maximale Drosselung gesetzt.

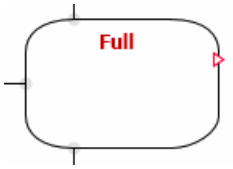
Die Trommel wird als leer angenommen, wenn die Wassermenge $M_{W/\rho}$ kleiner als eine gegebene minimale Füllung V_{min} wird:

$M_W < V_{min}\rho$.

Die Bilanzierung der Zustände wird für eine leere Trommel gestoppt, d. h. Änderungen der Dichte und spezifischen Enthalpie werden verworfen. Der Zustand leer wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Wassermenge vorliegt. In jedem Zeitschritt wird dazu geprüft, ob

$$M_W \geq V_{min}\,\rho\left(1+\frac{H_{min}}{100\%}\right)$$

gilt. Der erforderliche Zuwachs kann über die Hysterese H_{min} gesetzt werden. Im Symbol einer Komponente wird für eine leere Trommel ein entsprechender Hinweis eingeblendet



Grenzfall "Trommel voll"

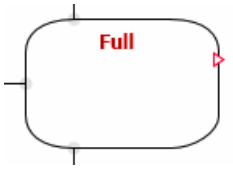
Die Trommel wird als voll angenommen, wenn ihre Wassermenge den maximal möglichen Wert erreicht hat:

$M_W < (V - V_{min})\,\rho$.

Die Bilanzierung der Zustände wird für eine volle Trommel gestoppt, d. h. Änderungen der Dichte und spezifischen Enthalpie werden verworfen. Der Zustand voll wird erst wieder verlassen, wenn eine genügend große Abnahme der Wassermenge vorliegt. In jedem Zeitschritt wird dazu geprüft, ob

$$M_W \leq \rho V - \rho V_{min}\left(1+\frac{H_{min}}{100\%}\right)$$

gilt. Die erforderliche Abnahme kann über die Hysterese H_{min} gesetzt werden. Im Symbol einer Komponente wird für eine volle Trommel ein entsprechender Hinweis eingeblendet.



Parameter

Parametername	Beschreibung	Einheit	Standardwert
NbrOfStubsW	Anzahl N_W der Anschlüsse *STUBWx*	–	1
NbrOfStubsS	Anzahl N_S der Anschlüsse *STUBSx*	–	1
Position	Lage der Trommel: *Vertically* oder *Horizontally*	–	*Vertically*
VolumeDrum	Volumen *V* der Trommel; online änderbar	m^3	10.0
HeightOrLength	Höhe bzw. Länge der Trommel; online änderbar	m	5.0
MassDrum	Masse M_T der Trommel; online änderbar	kg	5000.0
SurfaceDrum	Innere Oberfläche *A* der Trommel; online änderbar	m^2	12.5

Parametername	Beschreibung	Einheit	Standardwert
sHeatCapDrum	spezifische Wärmekapazität c_T der Trommel; online änderbar	kJ/kgK	0.5
HeatTransCoe	Wärmeübergangskoeffizient α für Wasser/Dampf der Trommel; online änderbar	kW/m²K	4.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
VolumeInit	Initialisierungswert für das Wasservolumen	%	50.0
TemperatureInit	Initialisierungswert für die Temperatur von Wasser/Dampf (Sättigungstemperatur)	°C	20.0
Kvs	für alle Anschlüsse einheitlicher Ventilkennwert k_V	m³/h	360.0
Kv0	Ventilkennwert k_{V0} für starke Drosselung im Trommelanschluss	m³/h	0.00001
MinVolume	*Minimales V*olumen V_{min} der Trommel; online änderbar	m³	0.01
MinVolumeHys	Hysterese H_{min}; online änderbar	%	50.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Wasserniveau	L	m	Ja[1]
Sättigungstemperatur	T	°C	Ja[1]
Gewicht von Wasser/ Dampf	M	kg	Ja[1]
Sättigungsdruck	p_B	bar	Nein
Wasserdruck	p_G	bar	Nein
Behältertemperatur	T_{Drum}	°C	Nein

[1] Eingabe mit "Set" aktivieren

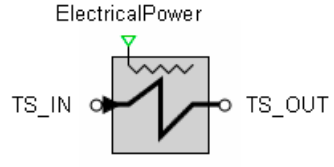
Im erweiterten Bedienfenster werden folgende Größen angezeigt:

- Mix: mittlere Größen
- Water: Größen für gesättigtes Wasser
- Steam: Größen für gesättigten Dampf

Größe	Formelzeichen	Einheit
Dichte	r	kg/m³
spezifische Enthalpie	h	kJ/kg

ElectricalHeaterWS – Elektrisch beheizter Wärmetauscher

Symbol



Funktion

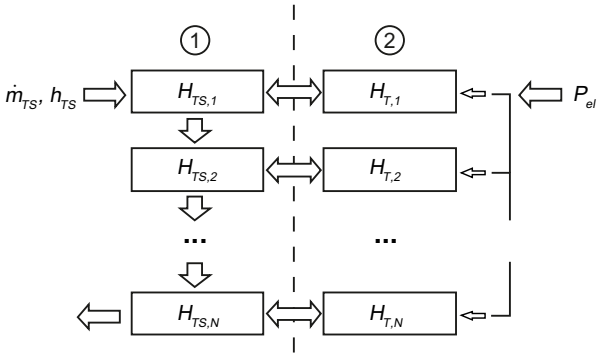
Der Komponententyp *ElectricalHeaterWS* dient zur Simulation eines elektrisch beheizten Wärmetauschers. Über den Anschluss *ElectricalPower* wird die elektrische Heizleistung P_{el} in kW vorgegeben.

Wasser/Dampf als beheiztes Medium wird über die Anschlüsse *TS_IN* und *TS_OUT* geführt. Der Durchfluss $\dot{m}$ wird gedrosselt gemäß der Beziehung

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho_{TS} \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

mit dem Drosselkennwert k_V. Die Bezugsrichtung für den Durchfluss ist von *TS_IN* nach *TS_OUT* gewählt.

Für die Wärmeübertragung wird ein einfaches Einrohrmodell mit Segmentierung angesetzt. Die Anzahl *N* der Segmente kann auf einen Wert von 4 bis 16 gesetzt werden.



① Rohrseitig (TubeSide)

② Rohr (Tube)

Es wird angenommen, dass die zugeführte elektrische Energie vollständig in Wärme umgewandelt wird. Die Wärmebilanz für ein Segment *i* des Ersatzrohrs mit einem stark vereinfachten Wärmeübergang lautet dann

$$\frac{dT_{T,i}}{dt} = a_T P_{el} + b_T \left(T_{TS,i} - T_{T,i} \right)$$

mit

$$a_T = \frac{1}{M_T c_T}, \quad b_T = \frac{A_{TS}\alpha_{TS}}{M_T c_T}$$

Weiter ist die Wasser/Dampf-seitige Enthalpiebilanzierung für ein Segment *i* beschrieben durch

$$\frac{dh_{TS,i}}{dt} = a_{TS}\left(h_{TS,i-1} - h_{TS,i}\right) + b_{TS}\left(T_{T,i} - T_{TS,i}\right)$$

mit

$$a_{TS} = \frac{N\dot{m}}{\rho_{TS} V_{TS}}, \quad b_{TS} = \frac{A_{TS}\alpha_{TS}}{\rho_{TS} V_{TS}}$$

$T_{T,i}$ und $T_{TS,i}$ sind die Temperaturen des Rohrsegments und des Mediums im Segment, $h_{T,i}$ und $h_{TS,i}$ die zugehörigen spezifischen Enthalpien.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
NbrOfSegments	Anzahl *N* der Segmente: 4 ≤ *N* ≤ 16	–	4
Kvs	Drosselkennwert k_V	m^3/h	360.0
Volume	Volumen V_{TS} des Mediums (Rohrinnenvolumen)	m^3	2.25
Surface	Oberfläche A_{TS} des Rohrs auf der Wasser/Dampf-Seite	m^2	550.0
HeatTransCoef	Wärmeübergangskoeffizient α_{TS} vom Rohr auf Wasser/Dampf	kW/m^2K	4.0
sHeatCapTube	spezifische Wärmekapazität c_T des Rohrs	kJ/kgK	1.3
MassTube	Masse M_T des Rohrs	kg	10000.0

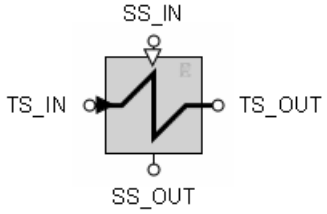
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Durchfluss	$\dot{m}$	kg/s
Druckabfall	Δp	bar
Enthalpiedifferenz	Δh	kJ/kg
Temperaturen für erstes und letztes Segment, getrennt nach Rohr und des beheizten Mediums	T_1 / T_N	°C
Elektrische Heizleistung	P_{el}	kW

HeatExchangerWS – Wärmetauscher Wasser/Dampf auf Wasser/Dampf

Symbol



Funktion

Der Komponententyp *HeatExchangerWS* dient zur Simulation eines Wärmetauschers für die Medien Wasser/Dampf auf der Rohr- wie auch Mantelseite. Implementiert ist die Simulation der drei Bauarten:

- Gleichstromwärmetauscher
- Gegenstromwärmetauscher
- Kreuzstromwärmetauscher

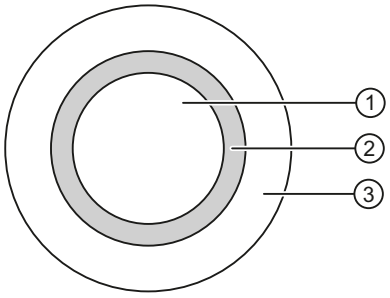
Die beiden Medien werden rohrseitig über die Anschlüsse *TS_IN* und *TS_OUT* geführt, mantelseitig über die Anschlüsse *SS_IN* und *SS_OUT*.

Der Durchfluss $\dot{m}$ wird rohr- und mantelseitig gedrosselt gemäß der Beziehung

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

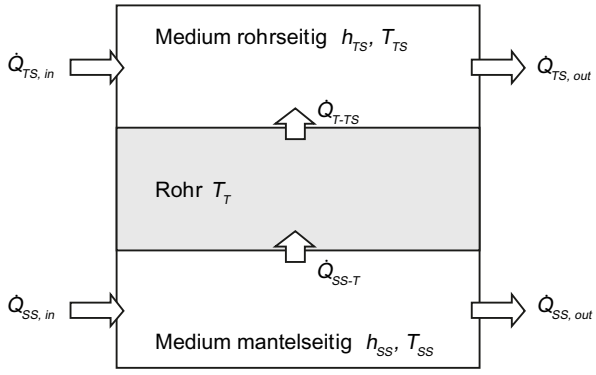
mit dem jeweiligen Drosselkennwert k_v. Die Bezugsrichtung für den Durchfluss ist rohr- wie mantelseitig vom Anschluss *_IN* nach *_OUT* gewählt.

Für die Wärmeübertragung wird ein einfaches Einrohrmodell mit Segmentierung angesetzt. Die Anzahl *N* der Segmente kann auf einen Wert von 4 bis 16 gesetzt werden.



① Rohrseitiger Raum
② Rohr
③ Mantelseitiger Raum

Bilanziert werden die Wärmen für Wasser/Dampf rohr- und mantelseitig und im Rohr selbst.



Die beiden Wärmeübergänge vom mantelseitigen Medium auf das Rohr und vom Rohr auf das rohrseitige Medium werden gemäß

$$\dot{Q}_{SS-T} = A_{SS}\dot{q}_{SS-T} = A_{SS}\alpha_{SS}\left(T_{SS} - T_T\right)$$

$$\dot{Q}_{T-TS} = A_{TS}\dot{q}_{T-TS} = A_{TS}\alpha_{TS}\left(T_T - T_{TS}\right)$$

stark vereinfacht angesetzt. Für jedes Segment *i* gelten dann die Wärmebilanzen

$$\frac{dh_{TS,i}}{dt} = a_{TS}\left(h_{TS,i-1} - h_{TS,i}\right) + b_{TS}\left(T_{T,i} - T_{TS,i}\right)$$

$$\frac{dT_{T,i}}{dt} = a_T\left(T_{SS,i} - T_{T,i}\right) + b_T\left(T_{TS,i} - T_{T,i}\right)$$

$$\frac{dh_{SS,i}}{dt} = a_{SS}\left(h_{SS,i-1} - h_{SS,i}\right) + b_{SS}\left(T_{T,i} - T_{SS,i}\right)$$

mit den für alle Segmente gleichen Koeffizienten

$$a_{TS} = \frac{N\dot{m}_{TS}}{\rho_{TS}V_{TS}},\quad b_{TS} = \frac{A_{TS}\alpha_{TS}}{\rho_{TS}V_{TS}},\quad a_{SS} = \frac{N\dot{m}_{SS}}{\rho_{SS}V_{SS}},\quad b_{SS} = \frac{A_{SS}\alpha_{SS}}{\rho_{SS}V_{SS}},\quad a_T = \frac{A_{SS}\alpha_{SS}}{M_T c_T},\quad b_T = \frac{A_{TS}\alpha_{TS}}{M_T c_T}$$

Für die Dichten ρ_{TS}, ρ_{SS} und die spezifischen Wärmekapazitäten c_{TS}, c_{SS} gelten die Werte, die für die Medien im mantel- bzw. rohrseitigen Flussnetz eingestellt sind.

Zur Initialisierung werden die Temperaturen der Rohrsegmente auf die Temperatur des rohrseitigen Mediums, berechnet aus dem Druck (Eingang *FNTS.PRESSURE*) und der spezifischen Enthalpie (Eingang *FNTS.HSPEC*) gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Type	Bauart des Wärmetauschers: *ParallelFlow* (Gleichstrom), *CounterFlow* (Gegenstrom), *CrossFlow* (Kreuzstrom)	–	*ParallelFlow*
NbrOfSegments	Anzahl N der Segmente: $4 \leq N \leq 16$	–	4
KvsSS	mantelseitiger Drosselkennwert k_V	m^3/h	360.0
KvsTS	rohrseitiger Drosselkennwert k_V	m^3/h	360.0
VolumeSS	mantelseitiges Volumen V_{SS}	m^3	1.65
VolumeTS	rohrseitiges Volumen V_{TS}	m^3	2.25
SurfaceSS	mantelseitige Oberfläche A_{SS} des Rohrs (Außenfläche des Rohrs)	m^2	720.0
SurfaceTS	rohrseitige Oberfläche A_{TS} des Rohrs (Innenfläche des Rohrs)	m^2	550.0
HeatTransCoefSS	Wärmeübergangskoeffizient α_{SS} auf der Rohraußenseite	kW/m^2K	4.0
HeatTransCoefTS	Wärmeübergangskoeffizient α_{TS} auf der Rohrinnenseite	kW/m^2K	4.0
sHeatCapTube	spezifische Wärmekapazität c_T des Rohrs	kJ/kgK	1.3
MassTube	Masse M_T des Rohrs	kg	10000.0

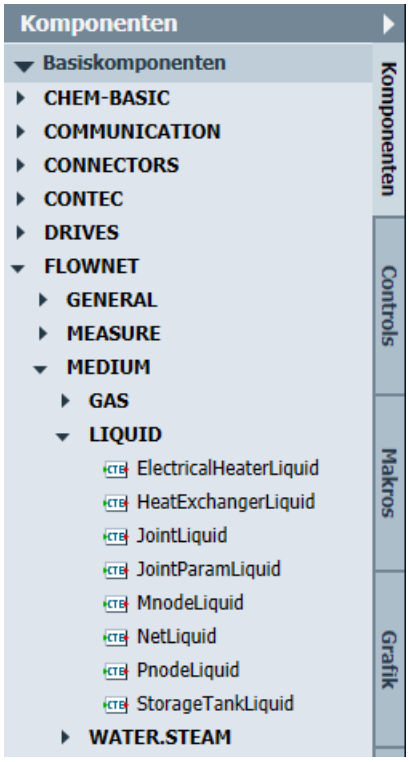
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Durchfluss	$\dot{m}$	kg/s
Druckabfall	Δp	bar
Enthalpiedifferenz	Δh	kJ/kg
Temperaturen für erstes und letztes Segment, getrennt nach • Rohr • Medium (mantelseitig) • Medium (rohrseitig)	T_1 / T_N	°C

8.2.6.4 Komponententypen für Flüssigkeiten als Medium

Im Verzeichnis *MEDIUM\LIQUID* der Bibliothek FLOWNET sind Komponententypen abgelegt, die in Flussnetzen mit Flüssigkeiten als Medium eingesetzt werden können. Der Parameter Medium für ein Flussnetz, das diese Komponenten enthält, ist gegebenenfalls auf den Wert "*Liquid*" zu setzen.



NetLiquid – Netzparametrierung

Symbol



Funktion

Der Komponententyp *NetLiquid* dient zur Parametrierung eines Netzes für Flüssigkeiten. Dazu wird die Komponente an beliebiger Stelle in einen beliebigen Zweig des Flussnetzes eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Density	Dichte der Flüssigkeit	kg/m^3	997.337
sHeatCapacity	spezifische Wärmekapazität der Flüssigkeit	kJ/kgK	4.18
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0
sCompression	spezifisches Kompressionsmodul der Flüssigkeit	bar/kg	100.0
FactorThermal	Faktor für die Enthalpiebilanzierung	1/kg	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck in den internen Knoten des Flussnetzes	bar	1.0
TemperatureInit	Initialisierungswert für die Temperatur in den internen Knoten des Flussnetzes	°C	20.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums in den internen Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

PnodeLiquid – Druckvorgabe

Symbol



Funktion

Der Komponententyp *PnodeLiquid* gibt Werte für Druck und Temperatur an seinem Anschluss vor. Er bildet damit einen Rand für das mit ihm verbundene Flussnetz. In der Betrachtung des Flussnetzes als Graph bildet *PnodeLiquid* einen (externen) Knoten, für den der Druck und die Temperatur fest vorgegeben sind.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Druck (Vorbelegung: 1)	p	bar	Ja
Temperatur (Vorbelegung: 20)	T	°C	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s
Dichte	r	kg/m³
Temperatur	T	°C

Für Ausströmung ist $\dot{m} > 0$, für Einströmung ist $\dot{m} < 0$.

MnodeLiquid – Massenstromvorgabe

Symbol



Funktion

Der Komponententyp *MnodeLiquid* gibt Werte für Massenstrom und Temperatur an seinem Anschluss vor. Er bildet damit einen Rand für das mit ihm verbundene Flussnetz. In der Betrachtung des Flussnetzes als Graph bildet *MnodeLiquid* eine Ein- oder Ausströmung in einen (internen) Knoten oder Zweig. Dazu wird in das Flussnetz ein interner Knoten mit der definierten Ein- oder Ausströmung eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Massenstrom (Vorbelegung: 0)	$\dot{m}$	kg/s	Ja
Temperatur (Vorbelegung: 20)	T	°C	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Dichte	r	kg/m³
Temperatur	T	°C

JointLiquid – Verzweigung

Symbol



Funktion

Über den Komponententyp *JointLiquid* können drei an seinen Anschlüssen angeschlossene Zweige in einem Knoten zusammengeführt werden. Alle Anschlüsse sind gleichwertig. Mit der Komponente *JointLiquid* wird ein interner Knoten in das Flussnetz eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C

JointParamLiquid – parametrierbare Verzweigung

Symbol



Funktion

Über den Komponententyp *JointParamLiquid* können drei an seinen Anschlüssen angeschlossene Zweige in einem Knoten zusammengeführt werden. Alle Anschlüsse sind gleichwertig. Mit der Komponente *JointParamLiquid* wird ein interner Knoten in das Flussnetz eingefügt. Dieser Knoten kann parametriert werden.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
sHeatCapacity	spezifische Wärmekapazität der Flüssigkeit	kJ/kgK	4.18
sCompression	spezifisches Kompressionsmodul der Flüssigkeit	bar/kg	100.0
FactorThermal	Faktor für die Enthalpiebilanzierung	1/kg	0.1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck im Knoten	bar	1.0
TemperatureInit	Initialisierungswert für die Temperatur im Knoten	°C	20.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums im Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

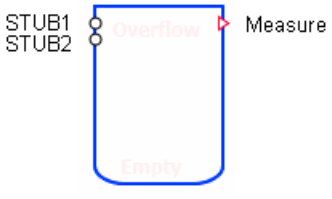
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C

StorageTankLiquid – Speicherbehälter

Symbol



Funktion

Der Komponententyp *StorageTankLiquid* stellt die Simulation eines offenen, d. h. nicht gegen die Umgebung abgeschlossenen Behälters für Flüssigkeiten zur Verfügung.

Zu- und Abflüsse erfolgen über Anschlüsse *STUBx* am Behälter. Für jeden der *N* Anschlüsse wird eine Drosselwirkung gemäß

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

angenommen. Der Behälter kann minimal einen, maximal 16 Anschlüsse haben. Die Anschlüsse können mit der Maus bei gedrückter ALT-Taste beliebig auf dem Umriss des Komponentensymbols verschoben werden.

Es wird eine unmittelbare vollständige Durchmischung der Flüssigkeit im Behälter angenommen, d. h. ein homogenes Medium im Behälter mit überall gleicher Dichte und Temperatur.

Die Zu- und Abflüsse werden über die *N* Anschlüsse des Behälters bilanziert. Man erhält so die über die *N* Massenströme $\dot{m}_i$ bilanzierte Masse *M* der Flüssigkeit im Behälter gemäß

$$\frac{dM}{dt} = \sum_{i=1}^{N} \dot{m}_i$$

Die Temperatur der Flüssigkeit wird als Mischungstemperatur aus den Zuflüssen ($i \in Z$) gemäß

$$\frac{dT}{dt} = \frac{1}{M} \left(\frac{1}{c_L} \sum_{i=1, i \in Z}^{N} h_i \dot{m}_i - T \sum_{i=1, i \in Z}^{N} \frac{1}{\rho_i} \dot{m}_i \right)$$

bilanziert.

Mit diesen Bilanzen für Masse *M* und Temperatur *T* ist das dynamische Verhalten der Flüssigkeit im Behälter beschrieben.

Am Anschluss *Measure* werden die berechneten Größen für das Niveau *l* der Flüssigkeit im Behälter, seine Masse *M* und Temperatur *T* sowie der Druck p_0 am Behälterboden ausgegeben. Der Druck *p* am Behälterboden bildet sich aus dem Gewichtsdruck $\rho g l$ der Flüssigkeit und dem Umgebungsdruck p_U zu

$p = p_U + \rho g l$

Grenzfall "Behälter leer"

Für einen leeren Behälter wird der Abfluss stark gedrosselt. Der Ventilkennwert wird dazu für alle Anschlüsse, über die Medium abfließt, auf den Wert k_{V0} für maximale Drosselung gesetzt.

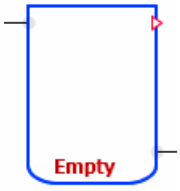
Der Behälter wird als leer angenommen, wenn die Füllung *V* kleiner als eine gegebene minimale Füllung V_{min} wird:

$V < V_{min}$

Die Bilanzierung der Zustände wird für einen leeren Behälter gestoppt, d. h. Änderungen der Masse und Temperatur der Flüssigkeit werden verworfen. Der Zustand leer wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Füllung vorliegt. In jedem Zeitschritt wird dazu geprüft, ob

$$M \geq V_{min}\,\rho\left(1+\frac{H_{min}}{100\%}\right)$$

gilt. Der erforderliche Zuwachs kann über die Hysterese H_{min} gesetzt werden. Im Symbol einer Komponente wird für einen leeren Behälter ein entsprechender Hinweis eingeblendet.

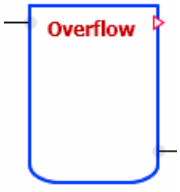


Grenzfall "Behälter voll"

Für einen vollen Behälter wird in der Berechnung lediglich die bilanzierte Füllung begrenzt:

$M = V\rho$

Es wird ein entsprechender Hinweis im Symbol eingeblendet.



Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Behälters; online änderbar	m^3	10.0
Height	Höhe des Behälters; online änderbar	m	5.0
NbrOfStubs	Anzahl *N* der Anschlüsse	–	1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureOutside	Umgebungsdruck p_U; online änderbar	bar	1.0
LevelInit	Initialisierungswert für das Niveau	%	50.0
TemperatureInit	Initialisierungswert für die Temperatur der Füllung	°C	20.0
Kvs	für alle Anschlüsse einheitlicher Ventilkennwert k_V	m^3/h	360.0
Kv0	Ventilkennwert k_{V0} für starke Drosselung im Behälteranschluss	m^3/h	0.000001

Parametername	Beschreibung	Einheit	Standardwert
Density	Dichte ρ der Flüssigkeit im Behälter	kg/m³	997.337
sHeatCapacity	spezifische Wärmekapazität c_L der Flüssigkeit im Behälter	kJ/kgK	4.18
MinVolume	Minimales Volumen V_{min} des Behälters; online änderbar	m³	0.001
MinVolumeHys	Hysterese H_{min}; online änderbar	%	50.0

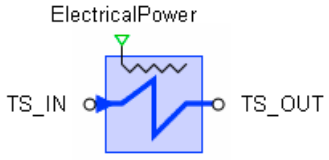
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Niveau	l	m
Druck	p	bar
Temperatur	T	°C
Gewicht	M	kg

ElectricalHeaterLiquid – Elektrisch beheizter Wärmetauscher

Symbol



Funktion

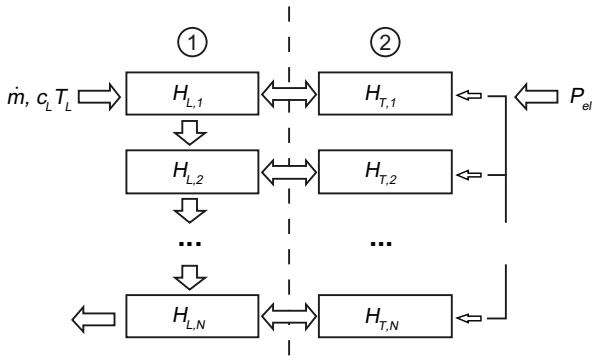
Der Komponententyp *ElectricalHeaterLiquid* dient zur Simulation eines elektrisch beheizten Wärmetauschers. Über den Anschluss *ElectricalPower* wird die elektrische Heizleistung P_{el} in kW vorgegeben.

Die beheizte Flüssigkeit wird über die Anschlüsse *TS_IN* und *TS_OUT* geführt. Der Durchfluss $\dot{m}$ wird gedrosselt gemäß der Beziehung

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

mit dem Drosselkennwert k_V. Die Bezugsrichtung für den Durchfluss ist von *TS_IN* nach *TS_OUT* gewählt.

Für die Wärmeübertragung wird ein einfaches Einrohrmodell mit Segmentierung angesetzt. Die Anzahl *N* der Segmente kann auf einen Wert von 4 bis 16 gesetzt werden.



① Rohrseitig
② Rohr

Es wird angenommen, dass die zugeführte elektrische Energie vollständig in Wärme umgewandelt wird. Die Wärmebilanz für ein Segment *i* des Ersatzrohrs mit einem stark vereinfachten Wärmeübergang lautet dann

$$\frac{dT_{T,i}}{dt} = a_T P_{el} + b_T \left(T_{L,i} - T_{T,i} \right)$$

mit

$$a_T = \frac{1}{M_T c_T}, \quad b_T = \frac{A\,\alpha}{M_T c_T}$$

Weiter ist dann die Wärmebilanz der Flüssigkeit für ein Segment *i* beschrieben durch

$$\frac{dT_{L,i}}{dt} = a \left(T_{L,i-1} - T_{L,i} \right) + b \left(T_{T,i} - T_{L,i} \right)$$

mit

$$a = \frac{N\,\dot{m}}{\rho V}, \quad b = \frac{A\,\alpha}{c_L \rho V}$$

$T_{T,i}$ und $T_{L,i}$ sind die Temperaturen des Rohrsegments und der Flüssigkeit im Segment *i*. Für die Dichte ρ und die spezifische Wärmekapazität c_L der Flüssigkeit gelten die Werte des Flussnetzes.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
NbrOfSegments	Anzahl *N* der Segmente: 4 ≤ *N* ≤ 16	–	4
Kvs	Drosselkennwert k_V	m^3/h	360.0
Volume	Volumen *V* der Flüssigkeit (Rohrinnenvolumen)	m^3	2.25

Parametername	Beschreibung	Einheit	Standardwert
Surface	Oberfläche *A* des Rohrs auf der Innenseite	m^2	550.0
HeatTransCoef	Wärmeübergangskoeffizient α vom Rohr auf die Flüssigkeit	kW/m^2K	4.0
sHeatCapTube	spezifische Wärmekapazität c_T des Rohrs	kJ/kgK	1.3
MassTube	Masse M_T des Rohrs	kg	10000.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
TemperatureInit	Temperatur für Flüssigkeit und Rohr bei Initialisierung des Wärmetauschers	°C	20.0

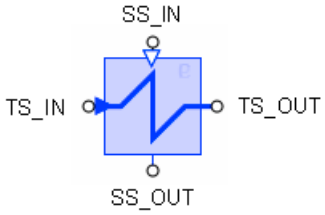
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Durchfluss	$\dot{m}$	kg/s
Druckabfall	Δp	bar
Temperaturdifferenz	ΔT	°C
Temperaturen für erstes und letztes Segment, getrennt nach Rohr und der beheizten Flüssigkeit	T_1/ T_N	°C
Elektrische Heizleistung	P_{el}	kW

HeatExchangerLiquid – Wärmetauscher Flüssigkeit auf Flüssigkeit

Symbol



Funktion

Der Komponententyp *HeatExchangerLiquid* dient zur Simulation eines Wärmetauschers für Flüssigkeiten auf der Rohr- wie auch Mantelseite. Implementiert ist die Simulation der drei Bauarten:

- Gleichstromwärmetauscher
- Gegenstromwärmetauscher
- Kreuzstromwärmetauscher

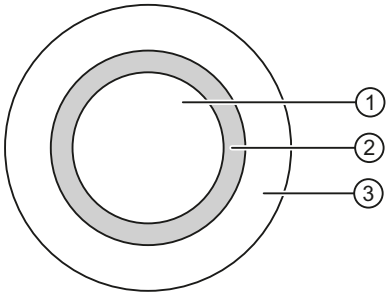
Die beiden Medien werden rohrseitig über die Anschlüsse *TS_IN* und *TS_OUT* geführt, mantelseitig über die Anschlüsse *SS_IN* und *SS_OUT*.

Der Durchfluss $\dot{m}$ wird rohr- und mantelseitig gedrosselt gemäß der Beziehung

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho_{TS} \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

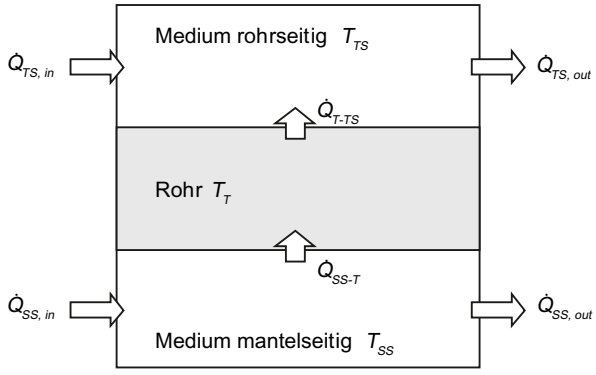
mit dem jeweiligen Drosselkennwert k_v. Die Bezugsrichtung für den Durchfluss ist rohr- wie mantelseitig vom Anschluss *_IN* nach *_OUT* gewählt.

Für die Wärmeübertragung wird ein einfaches Einrohrmodell mit Segmentierung angesetzt. Die Anzahl *N* der Segmente kann auf einen Wert von 4 bis 16 gesetzt werden.



① Rohrseitiger Raum
② Rohr
③ Mantelseitiger Raum

Bilanziert werden die Wärmen für die Flüssigkeiten rohr- und mantelseitig und im Rohr selbst.



Die beiden Wärmeübergänge von der mantelseitigen Flüssigkeit auf das Rohr und vom Rohr auf die rohrseitige Flüssigkeit werden gemäß

$$\dot{Q}_{SS-T} = A_{SS}\dot{q}_{SS-T} = A_{SS}\alpha_{SS}\left(T_{SS} - T_T\right)$$

$$\dot{Q}_{T-TS} = A_{TS}\dot{q}_{T-TS} = A_{TS}\alpha_{TS}\left(T_T - T_{TS}\right)$$

stark vereinfacht angesetzt. Für jedes Segment *i* gelten dann die Wärmebilanzen

$$\frac{dT_{TS,i}}{dt} = a_{TS}\left(T_{TS,i-1} - T_{TS,i}\right) + b_{TS}\left(T_{T,i} - T_{TS,i}\right)$$

$$\frac{dT_{T,i}}{dt} = a_T\left(T_{SS,i} - T_{T,i}\right) + b_T\left(T_{TS,i} - T_{T,i}\right)$$

$$\frac{dT_{SS,i}}{dt} = a_{SS}\left(T_{SS,i-1} - T_{SS,i}\right) + b_{SS}\left(T_{T,i} - T_{SS,i}\right)$$

mit den für alle Segmente gleichen Koeffizienten

$$a_{TS} = \frac{N\dot{m}_{TS}}{\rho_{TS}V_{TS}},\quad b_{TS} = \frac{A_{TS}\alpha_{TS}}{\rho_{TS}V_{TS}},\quad a_{SS} = \frac{N\dot{m}_{SS}}{\rho_{SS}V_{SS}},\quad b_{SS} = \frac{A_{SS}\alpha_{SS}}{\rho_{SS}V_{SS}},\quad a_T = \frac{A_{SS}\alpha_{SS}}{M_T c_T},\quad b_T = \frac{A_{TS}\alpha_{TS}}{M_T c_T}$$

Für die Dichten ρ_{TS}, ρ_{SS} und die spezifischen Wärmekapazitäten c_{TS}, c_{SS} gelten die Werte, die für die Flüssigkeiten im mantel- bzw. rohrseitigen Flussnetz eingestellt sind.

Zur Initialisierung werden die Temperaturen der Rohrsegmente auf die Temperatur des rohrseitigen Mediums, berechnet aus der spezifischen Enthalpie (Eingang *FNTS.HSPEC*), gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Type	Bauart des Wärmetauschers: *ParallelFlow* (Gleichstrom), *CounterFlow* (Gegenstrom), *CrossFlow* (Kreuzstrom)	–	*ParallelFlow*
NbrOfSegments	Anzahl *N* der Segmente: 4 ≤ *N* ≤ 16	–	4
KvsSS	mantelseitiger Drosselkennwert k_V	m^3/h	360.0
KvsTS	rohrseitiger Drosselkennwert k_V	m^3/h	360.0
VolumeSS	mantelseitiges Volumen V_{SS}	m^3	1.65
VolumeTS	rohrseitiges Volumen V_{TS}	m^3	2.25
SurfaceSS	mantelseitige Oberfläche A_{SS} des Rohrs (Außenfläche des Rohrs)	m^2	720.0
SurfaceTS	rohrseitige Oberfläche A_{TS} des Rohrs (Innenfläche des Rohrs)	m^2	550.0

Parametername	Beschreibung	Einheit	Standardwert
HeatTransCoefSS	Wärmeübergangskoeffizient α_{SS} auf der Rohraußenseite	kW/m²K	4.0
HeatTransCoefTS	Wärmeübergangskoeffizient α_{TS} auf der Rohrinnenseite	kW/m²K	4.0
sHeatCapTube	spezifische Wärmekapazität c_T des Rohrs	kJ/kgK	1.3
MassTube	Masse M_T des Rohrs	kg	10000.0

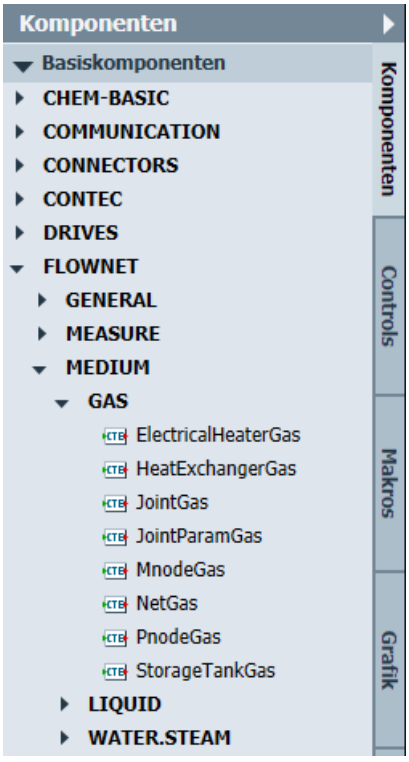
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Durchfluss	$\dot{m}$	kg/s
Druckabfall	Δp	bar
Temperaturdifferenz	ΔT	°C
Temperaturen für erstes und letztes Segment, getrennt nach • Rohr • Flüssigkeit (mantelseitig) • Flüssigkeit (rohrseitig)	T_1 / T_N	°C

8.2.6.5 Komponententypen für Gase als Medium

Im Verzeichnis *MEDIUM\GAS* der Bibliothek FLOWNET sind Komponententypen abgelegt, die in Flussnetzen mit Gasen als Medium eingesetzt werden können. Der Parameter Medium für ein Flussnetz, das diese Komponenten enthält, ist gegebenenfalls auf den Wert "*Ideal Gas*" zu setzen.



NetGas – Netzparametrierung

Symbol



Funktion

Der Komponententyp *NetGas* dient zur Parametrierung eines Netzes für Gase. Dazu wird die Komponente an beliebiger Stelle in einen beliebigen Zweig des Flussnetzes eingefügt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
GasConstant	spezifische Gaskonstante	kJ/kgK	0.287
sHeatCapacity	spezifische Wärmekapazität des Gases	kJ/kgK	1.0
FactorMomentum	Impulsfaktor für die Strömung in den Zweigen des Flussnetzes	m	450.0
sCompression	spezifisches Kompressionsmodul des Gases	bar/kg	10.0
FactorThermal	Faktor für die Enthalpiebilanzierung	1/kg	100.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck in den internen Knoten des Flussnetzes	bar	1.0
TemperatureInit	Initialisierungswert für die Temperatur in den internen Knoten des Flussnetzes	°C	20.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums in den internen Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

PnodeGas – Druckvorgabe

Symbol



Funktion

Der Komponententyp *PnodeGas* gibt Werte für Druck und Temperatur an seinem Anschluss vor. Er bildet damit einen Rand für das mit ihm verbundene Flussnetz. In der Betrachtung des Flussnetzes als Graph bildet *PnodeGas* einen (externen) Knoten, für den der Druck und die Temperatur fest vorgegeben sind.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Druck (Vorbelegung: 1)	p	bar	Ja
Temperatur (Vorbelegung: 20)	T	°C	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Massenstrom	$\dot{m}$	kg/s
Dichte	r	kg/m³
Temperatur	T	°C

Für Ausströmung ist $\dot{m} > 0$, für Einströmung ist $\dot{m} < 0$.

MnodeGas – Massenstromvorgabe

Symbol



Funktion

Der Komponententyp *MnodeGas* gibt Werte für Massenstrom und Temperatur an seinem Anschluss vor. Er bildet damit einen Rand für das mit ihm verbundene Flussnetz. In der Betrachtung des Flussnetzes als Graph bildet *Mnode* eine Ein- oder Ausströmung in einen (internen) Knoten oder Zweig. Dazu wird in das Flussnetz ein interner Knoten mit der definierten Ein- oder Ausströmung eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit	Größe änderbar?
Massenstrom (Vorbelegung: 0)	$\dot{m}$	kg/s	Ja
Temperatur (Vorbelegung: 20)	T	°C	Ja

Im erweiterten Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Dichte	r	kg/m³
Temperatur	T	°C

Für Ausströmung ist $\dot{m} > 0$, für Einströmung ist $\dot{m} < 0$.

JointGas – Verzweigung

Symbol



Funktion

Über den Komponententyp *JointGas* können drei an seinen Anschlüssen angeschlossene Zweige in einem Knoten zusammengeführt werden. Alle Anschlüsse sind gleichwertig. Mit der Komponente *JointGas* wird ein interner Knoten in das Flussnetz eingefügt.

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C
Dichte	r	kg/m³

JointParamGas – parametrierbare Verzweigung

Symbol



Funktion

Über den Komponententyp *JointParamGas* können drei an seinen Anschlüssen angeschlossene Zweige in einem Knoten zusammengeführt werden. Alle Anschlüsse sind gleichwertig. Mit der Komponente *JointParamGas* wird ein interner Knoten in das Flussnetz eingefügt. Dieser Knoten kann parametriert werden.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
GasConstant	spezifische Gaskonstante	kJ/kgK	0.287
sHeatCapacity	spezifische Wärmekapazität des Gases	kJ/kgK	1.0
sCompression	spezifisches Kompressionsmodul des Gases	bar/kg	10.0
FactorThermal	Faktor für die Enthalpiebilanzierung	1/kg	100.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck im Knoten	bar	1.0
TemperatureInit	Initialisierungswert für die Temperatur im Knoten	°C	20.0
TemperatureEnvironment	Umgebungstemperatur	°C	20.0
FactorHeatExchangeEnv	Proportionalitätsfaktor für Wärmeaustausch des Mediums im Knoten mit der Umgebung; für einen Wert null erfolgt kein Wärmeaustausch	kW/K	0.0

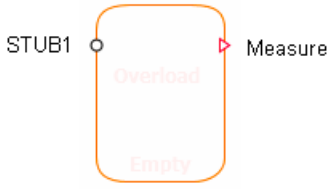
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C
Dichte	r	kg/m³

StorageTankGas – Speicherbehälter

Symbol



Funktion

Der Komponententyp *StorageTankGas* stellt die Simulation eines Gasbehälters zur Verfügung.

Zu- und Abflüsse erfolgen über Anschlüsse *STUBx* am Behälter. Für jeden der *N* Anschlüsse wird eine Drosselwirkung gemäß

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

angenommen. Der Behälter kann minimal einen, maximal 16 Anschlüsse haben. Die Anschlüsse können mit der Maus bei gedrückter ALT-Taste beliebig auf dem Umriss des Komponentensymbols verschoben werden.

Es wird eine unmittelbare vollständige Durchmischung des Gases im Behälter angenommen, d. h. ein homogenes Medium im Behälter mit überall gleicher Dichte und Temperatur.

Die Zu- und Abflüsse werden über die *N* Anschlüsse des Behälters bilanziert. Man erhält so die über die *N* Massenströme $\dot{m}_i$ bilanzierte Masse *M* des Gases im Behälter gemäß

$$\frac{dM}{dt} = \sum_{i=1}^{N} \dot{m}_i$$

Die spezifische Enthalpie *h* des Gases wird als Mischenthalpie aus den Zuflüssen ($i \in Z$) gemäß

$$\frac{dh}{dt} = \frac{1}{M} \left(\sum_{i=1, i \in Z}^{N} h_i \dot{m}_i - h \sum_{i=1, i \in Z}^{N} \dot{m}_i \right)$$

bilanziert.

Mit diesen Bilanzen für Masse *M* und spezifische Enthalpie *h* ist das dynamische Verhalten des Gases im Behälter beschrieben.

Am Anschluss *Measure* werden die Größen für den Druck *p*, die Temperatur

$$T = \frac{h}{c_G}$$

und die Masse *M* Gases ausgegeben. Der Druck berechnet sich aus der Gleichung für ideales Gas zu

$$p = \frac{R_S \left(T + 273{,}15\text{K} \right) M}{V \cdot 10^5 \frac{\text{Pa}}{\text{bar}}}$$

Grenzfall "Behälter leer"

Für einen leeren Behälter wird der Abfluss stark gedrosselt. Der Ventilkennwert wird dazu für alle Anschlüsse, über die Gas abfließt, auf den Wert k_{V0} für maximale Drosselung gesetzt.

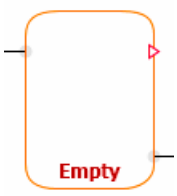
Der Behälter wird als leer angenommen, wenn die Füllung *M* kleiner als eine gegebene minimale Füllung M_{min} wird:

$M < M_{min}$

Die Bilanzierung der Zustände wird für einen leeren Behälter gestoppt, d. h. Änderungen der Masse und spezifischen Enthalpie des Gases werden verworfen. Der Zustand leer wird erst wieder verlassen, wenn ein genügend großer Zuwachs der Füllung vorliegt. In jedem Zeitschritt wird dazu geprüft, ob

$$M \geq M_{min}\left(1+\frac{H_{min}}{100\%}\right)$$

gilt. Der erforderliche Zuwachs kann über die Hysterese H_{min} gesetzt werden. Im Symbol einer Komponente wird für einen leeren Behälter ein entsprechender Hinweis eingeblendet.



Grenzfall "Behälter voll"

Um bei laufender Simulation den Druck im Behälter auf sinnvolle Werte zu begrenzen, wird der Behälter als voll angenommen, wenn sein Druck p einen gegebenen Maximaldruck p_{max} erreicht:

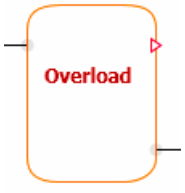
$p < p_{max}$

Für einen vollen Behälter wird der Zufluss stark gedrosselt. Der Ventilkennwert wird dazu für alle Anschlüsse, über die Gas zufließt, auf den Wert k_{V0} für maximale Drosselung gesetzt.

Weiter wird die Bilanzierung der Zustände für einen vollen Behälter gestoppt, d. h. Änderungen der Masse und spezifischen Enthalpie des Gases werden verworfen. Der Zustand voll wird erst wieder verlassen, wenn eine genügend große Abnahme ΔM der Füllung vorliegt. In jedem Zeitschritt wird dazu geprüft, ob

$$\Delta M \geq M_{min}\left(1+\frac{H_{min}}{100\%}\right)$$

gilt. Die erforderliche Abnahme kann über die Hysterese H_{min} gesetzt werden. Im Symbol einer Komponente wird für einen vollen Behälter ein entsprechender Hinweis eingeblendet.



Parameter

Parametername	Beschreibung	Einheit	Standardwert
Volume	Volumen *V* des Behälters; online änderbar	m^3	10.0
NbrOfStubs	Anzahl *N* der Anschlüsse	–	1

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
PressureInit	Initialisierungswert für den Druck; online änderbar	bar	1.0
TemperatureInit	Initialisierungswert für die Temperatur des Gases	°C	20.0
PressureMax	Maximaler Behälterdruck p_{max}; online änderbar	bar	100.0
Kvs	für alle Anschlüsse einheitlicher Ventilkennwert k_V	m^3/h	36.0
Kv0	Ventilkennwert k_{V0} für maximale Drosselung im Behälteranschluss	m^3/h	0.000001
GasConstant	spezifische Gaskonstante R_S für das Gas im Behälter	kJ/kgK	0.287
sHeatCapacity	spezifische Wärmekapazität c_G des Gases im Behälter	kJ/kgK	1.0
MinMass	Minimale Füllmenge M_{min} des Behälters; online änderbar	kg	0.015
MinMassHys	Hysterese H_{min}; online änderbar	%	50.0

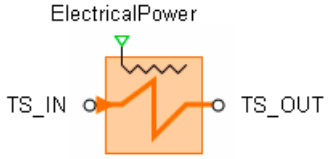
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Druck	p	bar
Temperatur	T	°C
Masse des Gases	M	kg

ElectricalHeaterGas – Elektrisch beheizter Wärmetauscher

Symbol



Funktion

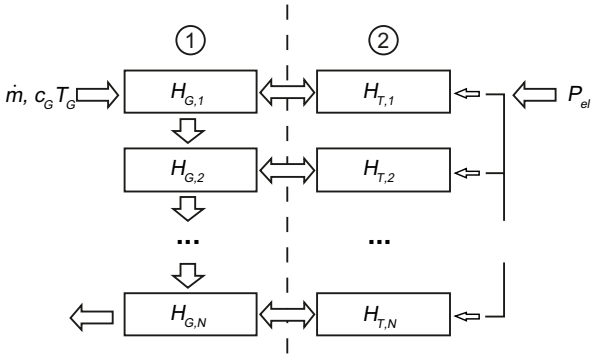
Der Komponententyp *ElectricalHeaterGas* dient zur Simulation eines elektrisch beheizten Gaswärmetauschers. Über den Anschluss *ElectricalPower* wird die elektrische Heizleistung P_{el} in kW vorgegeben.

Das beheizte Gas wird über die Anschlüsse *TS_IN* und *TS_OUT* geführt. Der Durchfluss $\dot{m}$ wird gedrosselt gemäß der Beziehung

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho_{TS} \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

mit dem Drosselkennwert k_V. Die Bezugsrichtung für den Durchfluss ist von *TS_IN* nach *TS_OUT* gewählt.

Für die Wärmeübertragung wird ein einfaches Einrohrmodell mit Segmentierung angesetzt. Die Anzahl *N* der Segmente kann auf einen Wert von 4 bis 16 gesetzt werden.



① Rohrseitig
② Rohr

Es wird angenommen, dass die zugeführte elektrische Energie vollständig in Wärme umgewandelt wird. Die Wärmebilanz für ein Segment *i* des Ersatzrohrs mit einem stark vereinfachten Wärmeübergang lautet dann

$$\frac{dT_{G,i}}{dt} = a_T P_{el} + b_T \left(T_{G,i} - T_{T,i} \right)$$

mit

$$a_T = \frac{1}{M_T c_T}, \quad b_T = \frac{A\,\alpha}{M_T c_T}$$

Weiter ist dann die Wärmebilanz des Gases für ein Segment *i* beschrieben durch

$$\frac{dT_{G,i}}{dt} = a \left(T_{G,i-1} - T_{G,i} \right) + b \left(T_{T,i} - T_{G,i} \right)$$

mit

$$a = \frac{N\,\dot{m}}{\rho V}, \quad b = \frac{A\,\alpha}{c_G \rho V}$$

$T_{T,i}$ und $T_{G,i}$ sind die Temperaturen des Rohrsegments und des Gases im Segment *i*. Für die Dichte ρ und die spezifische Wärmekapazität c_G des Gases gelten die Werte des Flussnetzes.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
NbrOfSegments	Anzahl *N* der Segmente: 4 ≤ *N* ≤ 16	–	4
Kvs	Drosselkennwert k_V	m^3/h	360.0
Volume	Volumen *V* des Gases (Rohrinnenvolumen)	m^3	2.25
Surface	Oberfläche *A* des Rohrs auf der Innenseite	m^2	550.0
HeatTransCoef	Wärmeübergangskoeffizient *α* vom Rohr auf das Gas	kW/m^2K	0.1
sHeatCapTube	spezifische Wärmekapazität c_T des Rohrs	kJ/kgK	1.3
MassTube	Masse M_T des Rohrs	kg	1000.0

Zusatzparameter

Parametername	Beschreibung	Einheit	Standardwert
TemperatureInit	Temperatur für Flüssigkeit und Rohr bei Initialisierung des Wärmetauschers	°C	20.0

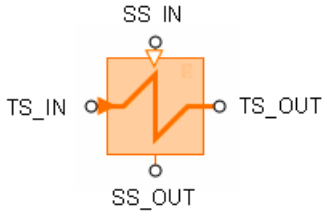
Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Durchfluss	$\dot{m}$	kg/s
Druckabfall	Δp	bar
Temperaturdifferenz	ΔT	°C
Temperaturen für erstes und letztes Segment, getrennt nach Rohr und des Gases	T_1/ T_N	°C
Elektrische Heizleistung	P_{el}	kW

HeatExchangerGas – Gas-Gas-Wärmetauscher

Symbol



Funktion

Der Komponententyp *HeatExchangerGas* dient zur Simulation eines Wärmetauschers für Gase auf der Rohr- wie auch Mantelseite. Implementiert ist die Simulation dieser drei Bauarten:

- Gleichstromwärmetauscher
- Gegenstromwärmetauscher
- Kreuzstromwärmetauscher

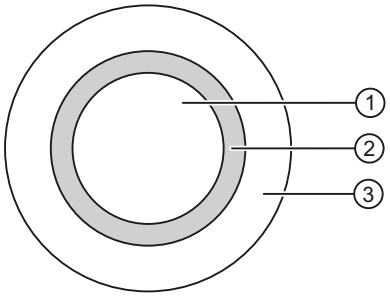
Die beiden Medien werden rohrseitig über die Anschlüsse *TS_IN* und *TS_OUT* geführt, mantelseitig über die Anschlüsse *SS_IN* und *SS_OUT*.

Der Durchfluss $\dot{m}$ wird rohr- und mantelseitig gedrosselt gemäß der Beziehung

$$\frac{\Delta p}{\text{bar}} = \frac{-\dot{m}^2}{k_V^2 \, \rho_{TS} \frac{\text{kg}}{\text{m}^3}} 12900 \left(\frac{\text{sec}}{\text{h}} \right)^2$$

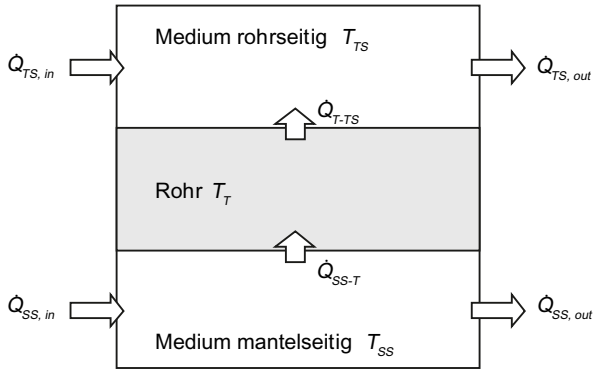
mit dem jeweiligen Drosselkennwert k_v. Die Bezugsrichtung für den Durchfluss ist rohr- wie mantelseitig vom Anschluss *_IN* nach *_OUT* gewählt.

Für die Wärmeübertragung wird ein einfaches Einrohrmodell mit Segmentierung angesetzt. Die Anzahl *N* der Segmente kann auf einen Wert von 4 bis 16 gesetzt werden.



① Rohrseitiger Raum
② Rohr
③ Mantelseitiger Raum

Bilanziert werden die Wärmen für die Gase rohr- und mantelseitig und im Rohr selbst.



Die beiden Wärmeübergänge vom mantelseitigen Gas auf das Rohr und vom Rohr auf das rohrseitige Gas werden gemäß

$$\dot{Q}_{SS-T} = A_{SS}\dot{q}_{SS-T} = A_{SS}\alpha_{SS}\left(T_{SS} - T_T\right)$$

$$\dot{Q}_{T-TS} = A_{TS}\dot{q}_{T-TS} = A_{TS}\alpha_{TS}\left(T_T - T_{TS}\right)$$

stark vereinfacht angesetzt. Für jedes Segment *i* gelten dann die Wärmebilanzen

$$\frac{dT_{TS,i}}{dt} = a_{TS}\left(T_{TS,i-1} - T_{TS,i}\right) + b_{TS}\left(T_{T,i} - T_{TS,i}\right)$$

$$\frac{dT_{T,i}}{dt} = a_T\left(T_{SS,i} - T_{T,i}\right) + b_T\left(T_{TS,i} - T_{T,i}\right)$$

$$\frac{dh_{SS,i}}{dt} = a_{SS}\left(h_{SS,i-1} - h_{SS,i}\right) + b_{SS}\left(T_{T,i} - T_{SS,i}\right)$$

mit den für alle Segmente gleichen Koeffizienten

$$a_{TS} = \frac{N\dot{m}_{TS}}{\rho_{TS}V_{TS}}, \quad b_{TS} = \frac{A_{TS}\alpha_{TS}}{c_{TS}\rho_{TS}V_{TS}}, \quad a_{SS} = \frac{N\dot{m}_{SS}}{\rho_{SS}V_{SS}}, \quad b_{SS} = \frac{A_{SS}\alpha_{SS}}{c_{SS}\rho_{SS}V_{SS}}, \quad a_T = \frac{A_{SS}\alpha_{SS}}{M_T c_T}, \quad b_T = \frac{A_{TS}\alpha_{TS}}{M_T c_T}$$

Für die Dichten ρ_{TS}, ρ_{SS} und die spezifischen Wärmekapazitäten c_{TS}, c_{SS} gelten die Werte, die für die Gase im mantel- bzw. rohrseitigen Flussnetz eingestellt sind.

Zur Initialisierung werden die Temperaturen der Rohrsegmente auf die Temperatur des rohrseitigen Gases, berechnet aus der spezifischen Enthalpie (Eingang *FNTS.HSPEC*), gesetzt.

Parameter

Parametername	Beschreibung	Einheit	Standardwert
Type	Bauart des Wärmetauschers: *ParallelFlow* (Gleichstrom), *CounterFlow* (Gegenstrom), *CrossFlow* (Kreuzstrom)	–	*ParallelFlow*
NbrOfSegments	Anzahl N der Segmente: $4 \le N \le 16$	–	4
KvsSS	mantelseitiger Drosselkennwert k_V	m^3/h	360.0
KvsTS	rohrseitiger Drosselkennwert k_V	m^3/h	360.0
VolumeSS	mantelseitiges Volumen V_{SS}	m^3	1.65
VolumeTS	rohrseitiges Volumen V_{TS}	m^3	2.25
SurfaceSS	mantelseitige Oberfläche A_{SS} des Rohrs (Außenfläche des Rohrs)	m^2	720.0
SurfaceTS	rohrseitige Oberfläche A_{TS} des Rohrs (Innenfläche des Rohrs)	m^2	550.0
HeatTransCoefSS	Wärmeübergangskoeffizient α_{SS} auf der Rohraußenseite	kW/m^2K	0.1
HeatTransCoefTS	Wärmeübergangskoeffizient α_{TS} auf der Rohrinnenseite	kW/m^2K	0.1
sHeatCapTube	spezifische Wärmekapazität c_T des Rohrs	kJ/kgK	1.3
MassTube	Masse M_T des Rohrs	kg	1000.0

Bedienfenster

Im Bedienfenster werden folgende Größen angezeigt:

Größe	Formelzeichen	Einheit
Durchfluss	$\dot{m}$	kg/s
Druckabfall	Δp	bar
Temperaturdifferenz	ΔT	°C
Temperaturen für erstes und letztes Segment, getrennt nach • Rohr • Gas (mantelseitig) • Gas (rohrseitig)	T_1/ T_N	°C

8.2.7 Komponententypen für Flussnetze selbst erstellen

Mit dem Komponententypeditor CTE sind Sie in der Lage, eigene Komponententypen zu erstellen, die die Mechanismen des FLOWNET-Verfahrens nutzen. Sie können so die mit der Bibliothek FLOWNET gelieferten Komponenten funktional erweitern, also beispielsweise physikalische Effekte hinzufügen, die in den gelieferten Bibliothekskomponenten vernachlässigt worden sind und so die Tiefe Ihrer Simulationen von Flussnetzen erhöhen. Sie können auch völlig eigene Komponententypen erstellen und so die Bibliothek FLOWNET erweitern.

Bei der Erstellung von Komponenten sind zwei Aspekte zu berücksichtigen:

- die topologischen Aspekte
- die Anbindung an das Lösungsverfahren

Der topologische Aspekt wird durch entsprechende Ergänzungen der Komponententypdefinition abgedeckt. Für die Anbindung an das Lösungsverfahren werden spezielle Verbindungstypen bereitgestellt.

Im Übrigen bilden die allgemein im Handbuch "SIMIT – Component Type Editor" beschriebenen Eigenschaften von Komponenten die Grundlage der Erstellung von Flussnetzkomponenten. Die allgemeinen Eigenschaften sind ohne Einschränkung auch für Flussnetzkomponenten gültig.

8.2.7.1 Topologische Eigenschaften

Die Topologie eines Flussnetzes wird beim Übersetzen des Simulationsprojekts aus der Zusammenschaltung der Flussnetzkomponenten automatisch ermittelt. Somit muss jede Flussnetzkomponente die sie betreffende topologische Information bereitstellen, also die Information, wie sie topologisch im Flussnetz zu betrachten ist. Unter topologischen Gesichtspunkten ist es erforderlich, zu wissen, wie für eine Komponente die Bezugsrichtungen für Größen des Flussnetzes definiert sind und wie der Datenaustausch der Komponente mit dem Lösungsverfahren für das Flussnetz gerichtet ist.

Elemente des Flussnetzes mit topologischer Information sind:

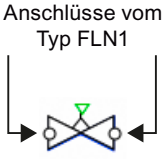
- interne Knoten
- externe Knoten
- Zweigobjekte

Für jedes dieser Elemente sind seine Beziehungen zu den topologischen Anschlüssen der Flussnetzkomponente zu definieren. Zur Definition öffnen Sie den Topologieeditor mit einem Doppelklick auf das Element *Topologie* in der Navigation des Komponententyps.

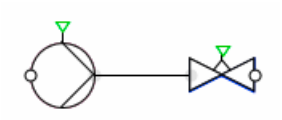


Anschlüsse vom Verbindungstyp FLN1

Der Verbindungstyp *FLN1* kennzeichnet die Anschlüsse von Flussnetzkomponenten, über die sie miteinander verbunden werden. Aus den miteinander verbundenen Anschlüssen dieses Typs und der topologischen Information der einzelnen Flussnetzkomponenten wird die Topologie eines Flussnetzes gewonnen. Der Typ *FLN1* ist damit ein rein topologischer Verbindungstyp. Er trägt folglich auch keine Signale. Anschlüsse dieses Typs werden daher im Folgenden kurz **topologische Anschlüsse** genannt. Topologische Anschlüsse werden durch Kreise dargestellt.



Werden topologische Anschlüsse im Diagrammeditor durch eine Verbindungslinie miteinander verbunden, dann werden die verbundenen Anschlüsse ausgeblendet. Sie sind nur noch als schwacher Schatten sichtbar.



Die Anschlüsse können auch durch Aufeinanderlegen verbunden werden. Beide Anschlüsse werden nach dem Verbinden ausgeblendet.



Topologie des internen Knotens

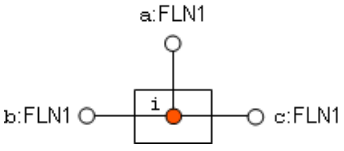
Ein interner Knoten des Flussnetzes wird in einem Komponententyp in der Topologiebeschreibung wie folgt definiert:

```
INTERNAL_NODE i;
```

Zusätzlich sind diesem internen Knoten topologische Anschlüsse der Komponente zuzuordnen. Drei Anschlüsse können in der Topologiebeschreibung beispielsweise wie folgt zugeordnet werden:

```
FROM i TO a;
FROM i TO b;
FROM i TO c;
```

Die drei Anschlüsse `a`, `b` und `c` sind im Komponententyp als Anschlüsse vom Typ *FLN1* zu definieren. Das Schema der damit definierten topologischen Struktur des Komponententyps zeigt folgende Abbildung.



Topologie des externen Knotens

Externe Knoten stellen die Berandung des Flussnetzes dar. Mit Hilfe von externen Knoten können Drücke und Enthalpien in Knoten und Ein- oder Ausströmungen für das Flussnetz vorgegeben werden.

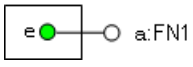
In der topologischen Beschreibung eines Komponententyps wird ein externer Knoten e wie folgt definiert:

```
EXTERNAL_NODE e;
```

Weiter ist dieser externe Knoten mit mindestens einem topologischen Anschluss der Komponente zu verbinden. Die Topologiebeschreibung ist beispielsweise gemäß

```
FROM e TO a;
```

zu ergänzen. Der topologische Anschluss `a` ist im Komponententyp als Anschluss vom Typ *FLN1* zu definieren. Das Schema der damit definierten topologischen Struktur des Komponententyps zeigt folgende Abbildung.



Topologie eines Zweigobjekts

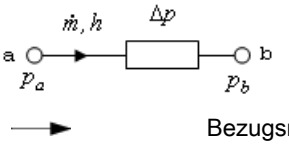
Zweigobjekte sind Teile eines Zweiges eines Flussnetzes, wie beispielsweise eine Drosselung des Druckes.

In der Topologiebeschreibung der Flussnetzkomponente wird die Bezugsrichtung für die Größen des Zweigobjekts festgelegt. Beispielsweise wird mit der Definition

```
FROM a TO b;
```

das Zweigobjekt mit einer Bezugsrichtung vom Anschluss `a` zum Anschluss `b` der Komponente festgelegt. Die beiden topologischen Anschlüsse `a` und `b` sind im Komponententyp als Anschlüsse vom Typ *FLN1* zu definieren.

In Bezugsrichtung werden Strömungsgrößen positiv gezählt. Für ein Zweigobjekt sind das der Massenfluss $\dot{m}$ und der Enthalpiestrom $H = \dot{m}h$.



Bezugsrichtung

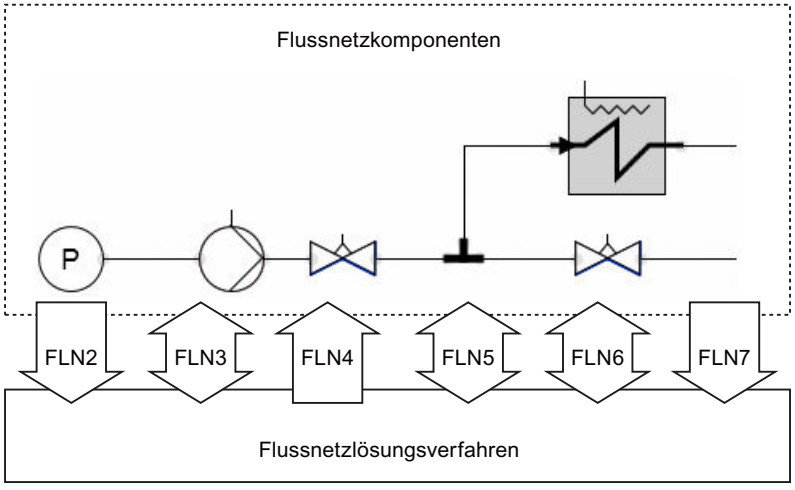
Druckerhöhungen werden in Bezugsrichtung angesetzt, d. h. die Druckänderung wird hier als

$\Delta p = p_b - p_a$

definiert. Für Druckverluste in Bezugsrichtung (Drosselung) ist Δp somit negativ, für Druckerhöhungen in Bezugsrichtung resultiert ein positives Δp.

8.2.7.2 Anbindung an das Lösungsverfahren

Die Flussnetzobjekte einer Komponente werden über spezielle Flussnetzanschlüsse datentechnisch mit dem Lösungsverfahren für das Flussnetz verbunden. Zu diesem Zweck stehen sechs verschiedene Verbindungstypen *FLN2* bis *FLN7* zur Verfügung. Ihr Einsatz in Flussnetzkomponenten ist in den Abschnitten Verbindungstyp FLN2 für Zweigobjekte (Seite 807) bis Verbindungstyp FLN7 zur Parametrierung eines internen Knotens (Seite 813) erläutert.



Anschlüsse der Verbindungstypen *FNL2* bis *FNL7* stellen eine Verbindung zum Flussnetzlöser her und sind somit als **nicht sichtbar** im Symbol der Komponente zu setzen. In den Eigenschaften des Anschlusses ist dazu die Verwendung "Nur im Eigenschaftsfenster" oder "Nur im CTE" zu setzen.

Eigenschaft	Wert
Verwendung	Nur im Eigenschaftsfenster
Voreinstellung Sichtbarkeit	

Siehe auch

Verbindungstyp FLN6 zur Parametrierung eines Zweiges (Seite 813)

Verbindungstyp FLN2 für Zweigobjekte

Über einen Anschluss vom Typ *FLN2* kann ein Zweigobjekt Größen mit dem Flussnetzlöser austauschen. Für einen Anschluss mit dem Namen *FN* wird die Topologiebeschreibung des Zweigobjekts dazu wie folgt ergänzt:

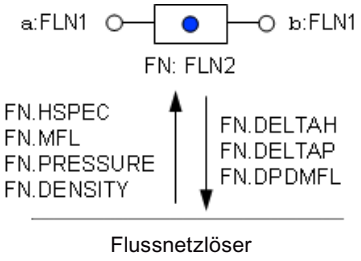
```
FROM a TO b : FN;
```

Der Anschluss ist immer in Richtung *OUT* zu definieren. Er schließt die Komponente an den Flussnetzlöser an, um verschiedene, das Zweigobjekt betreffende Größen zwischen dem Flussnetzlösungsverfahren und der Komponente auszutauschen. Über ihn wird seine

zweigbeeinflussende Wirkung, also die Wirkung, die das Zweigobjekt auf den Zweig hat, auf das Flussnetz abgebildet.

Name	Verbindungstyp	Richtung	Anzahl
FN	FLN2	OUT	1

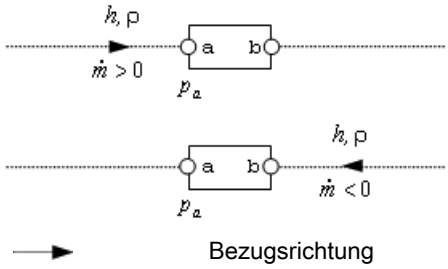
Folgende Abbildung zeigt die Ein- und Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Flussnetzlösungsverfahren.



Über die Eingangssignale erhält die Komponente vier Größen zur Berechnung der Wirkung des Zweigobjekts:

1. Spezifische Enthalpie *h* (*FN.HSPEC*)
2. Massenstrom $\dot{m}$ (*FN.MFL*)
3. Druck p_a (*FN.PRESSURE*)
4. Dichte ρ (*FN.DENSITY*)

Der Druck p_a ist auf den "von"-Anschluss (*FROM*) der Topologiebeschreibung, also hier auf den Anschluss a bezogen. Dichte ρ und Enthalpie *h* sind Größen des zufließenden Mediums.



Die Ausgangssignale werden für das Zweigobjekt in der Komponente berechnet und an das Flussnetzlösungsverfahren gegeben:

1. Änderung der spezifischen Enthalpie Δ*h* (*FN.DELTAH*)
2. Druckänderung $\Delta p = p_b - p_a$ (*FN.DELTAP*)
3. Ableitung der Druckänderung nach dem Massenstrom $d\Delta p/d\dot{M}$ (*FN.DPDMFL*)

Verbindungstyp FLN3 für externe Knoten

Über Anschlüsse vom Verbindungstyp *FLN3* werden Größen zwischen externen Knoten und dem Flussnetzlösungsverfahren ausgetauscht. Dazu wird dem externen Knoten in der Topologiebeschreibung ein Anschluss des Typs *FLN3* zugeordnet, für einen Anschluss mit Namen *FN* wie folgt:

```
EXTERNAL_NODE e : FN;
```

Der Verbindungstyp *FLN3* stellt Signale in Vorwärts- und Rückwärtsrichtung zur Verfügung wie in der folgenden Tabelle gelistet.

Tabelle 8-44 Signale des Verbindungstyps FLN3

Signale vorwärts		Signale rückwärts	
Name	**Größe**	**Name**	**Größe**
HFW	spezifische Enthalpie	HBW	spezifische Enthalpie
PRESSURE	Druck	MFL	Massenstrom

Somit gilt folgendes: Ist der Anschluss *FN* mit Richtung *OUT* definiert, dann werden der Druck und die spezifische Enthalpie über diesen Anschluss für den externen Knoten des Flussnetzes vorgegeben. Ist der Anschluss mit Richtung *IN* definiert, dann werden der Massenstrom und die spezifische Enthalpie für den externen Knoten vorgegeben.

Anschluss vom Typ FLN3 mit Richtung OUT

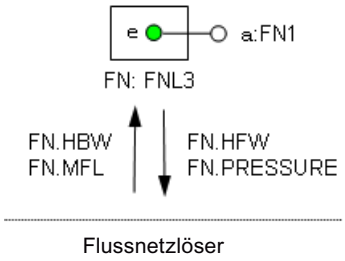
Wird der Anschluss mit Richtung *OUT* definiert, dann wird der in der Komponente berechnete Druck über diesen Anschluss für das Flussnetzlösungsverfahren vorgebbar. Der externe Knoten ist damit ein **Druckknoten**. An seinen Ausgängen sind folgende Größen vorgebbar:

1. Druck p_e (*FN.PRESSURE*)
2. spezifische Enthalpie h_e (*FN.HFW*)

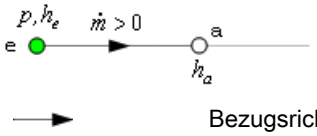
An seinen Eingängen wird der vom Flussnetzlöser berechnete, über den externen Knoten in das Flussnetz ein- oder ausgespeiste Massenstrom an die Komponente zurückgegeben. Bei Ausströmungen aus dem Flussnetz wird auch die spezifische Enthalpie des Stromes an die Komponente zurückgegeben:

1. Massenstrom $\dot{m}$ (*FN.MFL*)
2. spezifische Enthalpie h_a (*FN.HBW*)

Folgende Abbildung verdeutlicht die Signalrichtung des Datenaustauschs zwischen der Komponente und dem Flussnetzlösungsverfahren.



Der Massenstrom $\dot{m}$ ist unabhängig von der in der Topologiebeschreibung gewählten Richtung immer für Einströmungen in das Flussnetz positiv ($\dot{m} > 0$) und für Ausströmungen negativ ($\dot{m} > 0$).



Anschluss vom Typ FLN3 mit Richtung IN

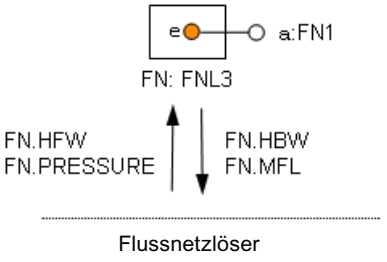
Wird der Anschluss *FN* mit Richtung *IN* definiert, dann wird der in der Komponente gesetzte Massenstrom über diesen Anschluss für das Flussnetz vorgebbar. Der im Flussnetz wirkende externe Knoten ist damit ein Knoten mit Masseneinströmung, kurz ein Massenstromknoten. An seinen Ausgängen sind folgende Größen vorgebbar:

1. Massenstrom $\dot{m}$ (FN.MFL)
2. spezifische Enthalpie *h* (FN.HBW)

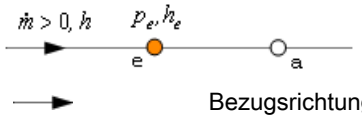
An seinen Eingängen wird der vom Flussnetzlöser für den externen Knoten berechnete Druck an die Komponente zurückgegeben. Wird mit negativem Massenstrom eine Ausströmung aus dem Flussnetz vorgegeben, dann wird die spezifische Enthalpie des Mediums an die Komponente zurückgegeben:

1. Druck p_e (FN.PRESSURE)
2. spezifische Enthalpie h_e (FN.HFW)

Folgende Abbildung verdeutlicht die Signalrichtung des Datenaustauschs zwischen der Komponente und dem Flussnetzlösungsverfahren.



Massenstromknoten wirken im Flussnetz wie ein interner Knoten mit Masseneinströmung oder -ausströmung. Diese Knoten werden daher im Flussnetz als interner Knoten behandelt, d. h. für einen externen Knoten wird im Flussnetz ein interner Knoten mit zusätzlicher Ein- oder Ausströmung angelegt. Der an die Komponente zurückgegebene Druck p_e entspricht damit dem Druck in diesem internen Knoten und die spezifische Enthalpie h_e entspricht der spezifischen Enthalpie dieses internen Knoten:



Verbindungstyp FLN4 für interne Knoten

Über einen Anschluss vom Typ *FLN4* können die für einen internen Knoten im Flussnetzlösungsverfahren berechneten Größen Druck, spezifische Enthalpie und Dichte in einer Komponente verwendet werden. Für einen Anschluss mit dem Namen *FN* wird die Topologiebeschreibung des Komponententyps dazu wie folgt ergänzt:

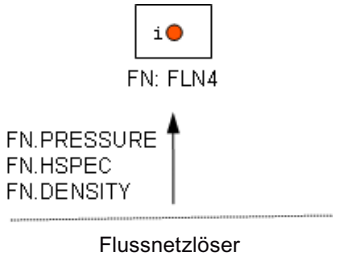
```
INTERNAL_NODE i : FN;
```

Der Anschluss ist immer in **Richtung IN** zu definieren. Er schließt die Komponente an das Flussnetz an, um die den internen Knoten *i* betreffenden Größen vom Flussnetzlösungsverfahren auf die Komponente abzubilden.

Name	Verbindungstyp	Richtung	Anzahl
FN	FLN4	IN	1

Über die Eingänge des Anschlusses werden die im Knoten *i* berechneten Größen zur Verfügung gestellt:

1. Druck p_i (FN.PRESSURE)
2. spezifische Enthalpie h_i (FN.HSPEC)
3. Dichte ρ_i (FN.DENSITY)



Verbindungstyp FLN5 für Parameter eines Flussnetzes

Über einen Anschluss vom Verbindungstyp *FLN5* können die Parameter eines Flussnetzes zwischen einer Komponente und dem Flussnetzlösungsverfahren ausgetauscht werden. Dieser Anschluss ist einem beliebigen Flussnetzobjekt zuzuordnen: einem Zweigobjekt oder einem internen oder externen Knoten. Die Topologiedefinition des Objekts ist entsprechend zu ergänzen, für einen Anschluss mit Namen *FN* gemäß

- `FROM a TO b : FN;`
- `INTERNAL_NODE i : FN;`
- `EXTERNAL_NODE e : FN;`

Die Signale des Verbindungstyps FLN5 sind in der folgenden Tabelle gelistet. Ihre Bedeutung im mathematischen Modell des Flussnetzes finden Sie im Kapitel: Parametrierung von Flussnetzen (Seite 623).

Hinweis

Über den Parameter *MEDIUM* werden die für das jeweilige Medium anzuwendenden Beziehungen zur Berechnung von Flussnetzgrößen, wie beispielsweise der Temperatur, definiert. Dies geschieht ausschließlich in der Initialisierung der Simulation. Änderungen des Parameters zur Laufzeit der Simulation sind somit wirkungslos.

Tabelle 8-45 Signale des Verbindungstyps FLN5

Signal	Bedeutung
MEDIUM	Medium des Flussnetzes: MEDIUM := 0 für Wasser/Dampf (Water/Steam), MEDIUM := 1 für ideales Gas (Ideal Gas), MEDIUM := 2 für Flüssigkeit (Liquid); darf bei laufender Simulation nicht geändert werden
CG	spezifisches Kompressionsmodul K/M für Medium Wasser/Dampf mit Dichten ρ < 500 kg/m³ und für Medium ideales Gas
CL	spezifisches Kompressionsmodul K/M für Medium Wasser/Dampf mit Dichten ρ ≥ 500 kg/m³ und für Medium Flüssigkeit
MG	Thermischer Faktor $1/M$ für Medium Wasser/Dampf mit Dichten ρ < 500 kg/m³ und für Medium ideales Gas
ML	Thermischer Faktor $1/M$ für Medium Wasser/Dampf mit Dichten ρ ≥ 500 kg/m³ und für Medium Flüssigkeit
P_INIT	Initialwert für den Druck in den internen Knoten des Flussnetzes
H_INIT	Initialwert für die spezifische Enthalpie in den internen Knoten des Flussnetzes
DENSITY	Dichte des Mediums im Flussnetz, wenn Flüssigkeit (Liquid) als Medium eingestellt ist
T_ENV	Umgebungstemperatur
C_ENV	Wärmeübergangsfaktor $c = \alpha A$ für Wärmeaustausch in den internen Knoten des Flussnetzes mit Umgebung
L_CR	spezifische Wärmekapazität c_p für Flüssigkeiten als Medium
IG_R	spezifische Gaskonstante R_S für ideales Gas als Medium
IG_CR	spezifische Wärmekapazität c_p für ideales Gas als Medium
ST	binäres Signal; falls ST := "True", wird ein linearer Übergang für die Parameter CL, CG und ML, MG für das Medium Wasser/Dampf angesetzt
AL	Impulsfaktor A für die Zweige im Flussnetz

Je nach Richtung des Anschlusses *FN* sind seine Signale Ein- oder Ausgänge. Somit können je nach Richtung Parameter des Flussnetzes gesetzt oder in der Komponente verwendet werden. Wird der Anschluss mit Richtung *OUT* definiert, dann werden die in der Komponente gesetzten Größen dem Flussnetz als Parameter vorgegeben. Die Parameter stehen zur Auswertung in der Komponente zur Verfügung, wenn der Anschluss mit Richtung *IN* definiert ist.

Verbindungstyp FLN6 zur Parametrierung eines Zweiges

Die Dynamik des Durchflusses kann für jeden Zweig individuell parametriert werden. Die Topologiedefinition eines Zweigobjekts ist dafür zu ergänzen durch einen Anschluss über den der Impulsfaktor A an den Flussnetzlöser übermittelt wird, für einen Anschluss *FN* wie folgt:

```
FROM a TO b: FN;
```

Wird der Anschluss vom Typ *FLN6* mit Richtung *OUT* definiert, dann wird über diesen Anschluss der Impulsfaktor A für den Zweig an den Flussnetzlöser übermittelt, in dem das Zweigobjekt liegt.

Name	Verbindungstyp	Richtung	Anzahl
FN	FLN6	OUT	1

Ist der Anschluss mit Richtung *IN* definiert, dann wird der für den Zweig geltende Impulsfaktor A vom Flussnetzlöser an das Zweigobjekt übergeben.

Verbindungstyp FLN7 zur Parametrierung eines internen Knotens

Jeder interne Knoten kann individuell parametriert werden. Die Topologiedefinition eines internen Knotens ist dafür zu ergänzen durch einen Anschluss über den die Parameter an den Flussnetzlöser übermittelt werden. Für einen Anschluss *FN* wird die Definition wie folgt ergänzt:

```
Internal_Node i: FN;
```

FN wird immer **als Austritt** vom Typ *FLN7* angelegt.

Name	Verbindungstyp	Richtung	Anzahl
FN	FLN7	OUT	1

Die in der folgenden Tabelle aufgeführten Größen werden für den internen Knoten an den Flussnetzlöser übermittelt. Ihre Bedeutung im mathematischen Modell des Flussnetzes finden Sie im Kapitel: Parametrierung von Flussnetzen (Seite 623).

Tabelle 8-46 Signale des Verbindungstyps FLN7

Signal	Bedeutung
CG	spezifisches Kompressionsmodul K_i / M_i für Medium Wasser/Dampf mit Dichten ρ < 500 kg/m³ und für Medium ideales Gas
CL	spezifisches Kompressionsmodul K_i / M_i für Medium Wasser/Dampf mit Dichten ρ ≥ 500 kg/m³ und für Medium Flüssigkeit
MG	Thermischer Faktor $1 / M_i$ für Medium Wasser/Dampf mit Dichten ρ < 500 kg/m³ und für Medium ideales Gas
ML	Thermischer Faktor $1 / M_i$ für Medium Wasser/Dampf mit Dichten ρ ≥ 500 kg/m³ und für Medium Flüssigkeit
P_INIT	Initialwert für den Druck
H_INIT	Initialwert für die spezifische Enthalpie
T_ENV	Umgebungstemperatur
C_ENV	Wärmeübergangsfaktor $c_i = \alpha A$ für Wärmeaustausch mit Umgebung

Signal	Bedeutung
L_CR	spezifische Wärmekapazität c_p für Flüssigkeiten als Medium
IG_R	spezifische Gaskonstante R_S für ideales Gas als Medium
IG_CR	spezifische Wärmekapazität c_p für ideales Gas als Medium

8.2.7.3 Konstanten und Funktionen

Beim Erstellen von eigenen Flussnetzkomponenten können Sie verschiedene Konstanten und Funktionen nutzen.

Konstanten

Die verwendbaren Konstanten sind in der folgenden Tabelle beschrieben.

Tabelle 8-47 Konstanten für Flussnetzkomponenten

Name	Datentyp	Wert	Beschreibung
_GRAVITY	analog	9,81	Gravitätskonstante (Erdbeschleunigung)
_T0	analog	273,15	Nullpunktstemperatur

Funktionen

Für Flussnetzkomponenten mit Wasser/Dampf als Medium stehen Ihnen Funktionen zur Berechnung von Zustandsgrößen zur Verfügung. Die verwendbaren Zustandsgrößen sind mit den verwendeten Einheiten in der folgenden Tabelle zusammengefasst.

Tabelle 8-48 Zustandsgrößen für Wasser/Dampf

Größe		Einheit
p	Druck	bar
p_S	Sattdampfdruck	bar
T	Temperatur	°C
T_S	Sattdampftemperatur	°C
h	spezifische Enthalpie	kJ/kg
h'	spezifische Enthalpie gesättigtes Wasser	kJ/kg
h''	spezifische Enthalpie gesättigter Dampf	kJ/kg
ρ	Dichte	kg/m³
ρ'	Dichte gesättigtes Wasser	kg/m³
ρ"	Dichte gesättigter Dampf	kg/m³

Alle Zustandsfunktionen *FNAME* geben eine Zustandsgröße *ZVAL* als Analogwert zurück. Der Aufruf erfolgt gemäß

ZVAL = _WaterSteam.*FNAME*(*PARAM1* (, *PARAM2*)).

Die zur Verfügung stehenden Zustandsfunktionen sind in der folgenden Tabelle beschrieben.

Tabelle 8-49 Zustandsfunktionen für Wasser/Dampf

ZVAL	FNAME	PARAM1	PARAM2	Zustandsfunktion
ρ	rph	p	h	ρ = ρ(p,h)
T	trh	ρ	h	T = T(ρ,h)
T_S	tvp	p_S	–	T_S = $T(p_S)$
p_S	pvt	T_S	–	p_S = $p(T_S)$
ρ'	rsvt	T_S	–	ρ' = ρ'(T_S)
ρ"	rssvt	T_S	–	ρ" = ρ"(T_S)
ρ'	rsvp	p_S	–	ρ' = ρ'(p_S)
ρ"	rssvp	p_S	–	ρ" = ρ"(p_S)
h'	hsvt	T_S	–	h' = h'(T_S)
h''	hssvt	T_S	–	h'' = h''(T_S)
h'	hsvp	p_S	–	h' = h'(p_S)
h''	hssvp	p_S	–	h'' = h''(p_S)

Für die Sättigungsfunktionen sind der Sattdampfdruck p_S und Sattdampftemperatur T_S wie folgt begrenzt:

0 ≤ T_S ≤ 373,94 °C und 0,0065 bar ≤ p_S ≤ 220,64 bar.

Für Werte, die außerhalb dieses Bereichs liegen, wird der jeweilige Grenzwert gesetzt.

Für die Zustandsfunktion ρ = ρ(p,h) erfolgt eine Eingrenzung der Werte für den Druck p und die spezifische Enthalpie h auf die folgenden Bereiche:

0,007 bar ≤ p ≤ 490 bar und 20 kJ/kg ≤ h ≤ 3998 kJ/kg

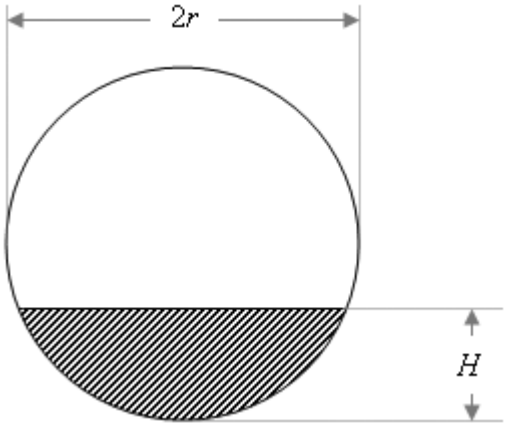
Für die Zustandsfunktion T = T(ρ,h) ist die spezifische Enthalpie auf den Bereich

9 kJ/kg ≤ h ≤ 4158 kJ/kg

eingegrenzt. Die Grenzen für die Dichte ρ sind abhängig vom Enthalpiewert: Der kleinste zulässige Dichtewert liegt bei 0,0012344 kg/m^3 (bei h = 4160 kJ/kg), der größte zulässige Dichtewert liegt bei 1045.239 kg/m^3 (bei h = 96 kJ/kg).

Für die Berechnung von Füllständen in liegenden Zylindern steht die Hilfsfunktion *Cylniv* zur Verfügung. Sie berechnet für einen Zylinder der Länge L und Radius r bei gegebener Füllung V die Füllhöhe H. Ihr Aufruf lautet:

H = _Utilities.*Cylniv*(V, L, r).



Eigene Funktionen

Die Bibliothek FLOWNET stellt Ihnen drei verschiedene Medien zur Verfügung: Wasser/ Dampf, ideales Gas und Flüssigkeit. Das zu verwendende Medium wird durch eine entsprechende Kennung (0, 1 oder 2) im Signal *MEDIUM* des Verbindungstypen *FLN5* gesetzt. Weitere Informationen hierzu finden Sie im Kapitel: Verbindungstyp FLN5 für Parameter eines Flussnetzes (Seite 811).

Vom Flussnetzlöser werden dementsprechend die für das Medium definierten Zustandsberechnungen für die Dichte und Temperatur in den Knoten angesetzt.

Mit der FLOWNET-Bibliothek wird Ihnen auch die Möglichkeit gegeben, Flussnetzsimulationen mit anderen als den vorgegebenen Medien durchzuführen. Sie müssen dazu die medienspezifische Zustandsberechnung der Dichte und Temperatur in den Knoten durch eine "Globale Funktion" selbst zur Verfügung stellen.

Eine solche globale Funktion muss aus einer .NET-Assembly mit dem Namen *Userdefined.dll* bestehen, die in den Projekteigenschaften den "Assembly Name" "Userdefined" besitzt. Sie muss den Namespace *Userdefined* haben und eine öffentliche Klasse mit Namen *FlownetFunctions* besitzen. Diese Klasse muss zwei öffentliche statische Funktionen zur Berechnung der Zustandsgrößen enthalten:

- *public static double trh(long m, double r, double h)*
- *public static double rph(long m, double p, double h)*

Die Funktion *trh* muss die Berechnung der Temperatur aus der Dichte *r* und der spezifischen Enthalpie *h* zur Verfügung stellen, die Funktion *rph* die Berechnung der Dichte aus dem Druck *p* und der spezifischen Enthalpie *h*. Die Einheiten der Größen sind wie folgt anzunehmen:

- Temperatur in °C,
- Dichte in kg/m^3,
- Druck in bar,
- spezifische Enthalpie in kJ/kg.

Das Medium wird durch eine beliebig frei wählbare negative Zahl spezifiziert. Diese negative Medienkennzahl bildet den ersten Übergabeparameter *m* der beiden Funktionen. Sie müssen in Ihren Funktionen nur selbstdefinierte negative Medienkennzahlen beachten, bei Werten ≥ 0

wird die Berechnung von SIMIT durchgeführt, Ihre Funktionen werden in diesem Fall nicht aufgerufen.

Sie haben somit die Möglichkeit, beliebige Medien in diesen beiden Funktionen selbst zu definieren und in einem Simulationsprojekt zu verwenden. In Ihren selbstdefinierten Funktionen müssen Sie nur negative Medienkennzahlen beachten. Bei nicht negativen Medienkennzahlen werden die in SIMIT enthaltenen Zustandsfunktionen benutzt, Ihre selbstdefinierten Funktionen werden in diesem Fall nicht aufgerufen.

Den Rahmen für Ihre selbstdefinierten Funktionen können Sie beispielsweise wie folgt anlegen:

```
namespace Userdefined
{
 public class FlownetFunctions
 {
  public static double trh(long m, double r, double h)
  {
   switch (m)
   {
    case -1:
     // Berechnung für Medium mit Kennzahl -1 …
     return 0.0;
    case -2:
     // Berechnung für Medium mit Kennzahl -2 …
     return 0.0;
    default:
     // Sollte nicht vorkommen!
     return 0.0;
   }
  }

  public static double rph(long m, double p, double h)
  {
   switch (m)
   {
    case -1:
     // Berechnung für Medium mit Kennzahl -1 …
     return 0.0;
    case -2:
     // Berechnung für Medium mit Kennzahl -2 …
     return 0.0;
    default:
     // Sollte nicht vorkommen!
     return 0.0;
   }
  }
 }
}
```

Um einem Flussnetz ein bestimmtes Medium zuzuordnen, setzen Sie im Signal *MEDIUM* des Verbindungstypen *FLN5* die entsprechende Medienkennzahl, d. h. im Falle von selbst definierten Medien die entsprechende negative Größe.

Hinweis

Wenn Sie für das Medium einen negativen Wert vorgeben, ohne dass eine entsprechende Funktion zur Verfügung gestellt worden ist, werden im Flussnetzlöser die Dichte und Temperatur mit dem Wert null angenommen.

Sie können diese beiden Funktionen mit dem Aufruf

_FlownetFunctions.trh(*m*, *r*, *p*)

bzw.

_FlownetFunctions.rph(*m, p, h*)

auch in eigenen Komponententypen verwenden. Darüber hinaus können Sie in der Klasse *FlownetFunctions* auch weitere Funktionen implementieren und über entsprechende Funktionsaufrufe in Komponententypen verwenden.

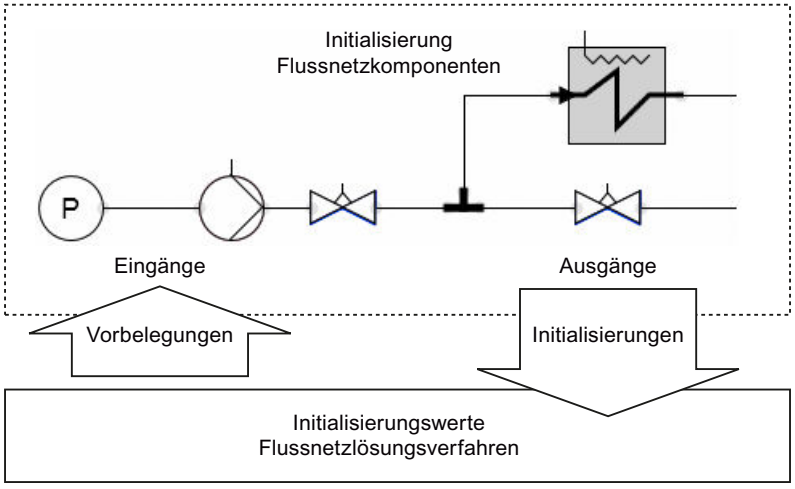
Die Assembly müssen Sie im Workspace von SIMIT im Unterordner *GlobalFunctions* ablegen. Sie wird dann automatisch in jedes neu angelegte Projekt kopiert und folglich auch mit dem Projekt archiviert.

Hinweis

Eine Modifikation der Assembly wirkt sich nicht auf Ihre bereits angelegten Projekte aus. Die modifizierte Assembly wird in Projekte übernommen, die Sie neu anlegen. Wenn Sie die modifizierte Assembly auch in bereits angelegten Projekten nutzen möchten, kopieren Sie sie in den Ordner *globalFunctions* in diesen Projekten.

8.2.7.4 Initialisierung von Flussnetzsimulationen

Die Initialisierung von Flussnetzsimulationen erfolgt in zwei Schritten. Wenn die Simulation gestartet wird, werden zunächst die Komponenten initialisiert, d. h. das aus den verbundenen Komponenten gebildete Simulationsmodell wird initialisiert. Im nächsten Schritt wird der Flussnetzlöser initialisiert. Der Flussnetzlöser verwendet dabei die in den Komponenten initialisierten und an ihn über die Anschlüsse vom Typ FLN2 bis FLN7 übertragenen Werte.



Da für die Initialisierung der Komponenten keine Größen des Flussnetzlösers zur Verfügung stehen, ist es ratsam, die Eingänge der Komponententypen mit geeigneten Werten vorzubelegen. Empfohlen werden Werte, die konsistent mit den Vorbelegungen des Flussnetzlösers sind.

8.3 Bibliothek CONTEC

8.3.1 Einleitung

Die Bibliothek *CONTEC* ist eine Erweiterung von SIMIT, die Komponententypen zum Erstellen von Simulationen von Förderanlagen zur Verfügung stellt. Durch Verschalten von Komponenten dieser Bibliothek wird in SIMIT ein Modell einer Förderanlage erzeugt, mit dem der Transport von Fördergütern in dieser Förderanlage simuliert werden kann. Mit der Bibliothek *CONTEC* wird zu diesem Zweck ein spezielles Lösungsverfahren in SIMIT nutzbar, das nach dem Starten der Simulation laufend die Positionen der Fördergüter in der modellierten Förderanlage berechnet.

Die Bibliothek *CONTEC* ermöglicht nur die Simulation von Förderanlagen mit Stückgütern, Förderanlagen mit Schüttgut können nicht simuliert werden. Weiter wird die Simulation eingeschränkt auf Förderanlagen, bei denen die Transportwege für die Güter durch die Anordnung der Fördermittel fest vorgegeben sind. Fördersysteme mit autonom operierenden Fahrzeugen, d. h. mit Fahrzeugen, die ihren Transportweg selbst bestimmen sind damit folglich ausgeschlossen.

SIMIT stellt eine Echtzeitsimulation zur virtuellen Inbetriebsetzung der Anwenderprogramme von Automatisierungssystemen zur Verfügung. Auch die Simulation von Förderanlagen mit der

Bibliothek CONTEC wird so immer in einem geschlossenen Wirkungskreis mit realen oder simulierten Automatisierungssystemen eingesetzt. Derartige Simulationen mit SIMIT sind somit grundlegend verschieden von Simulationen des Materialflusses, die zur Planung und Auslegung von Förderanlagen eingesetzt werden. Wird dort der Materialfluss in einer Förderanlage auf Basis einer vorgegebenen Steuerungsstrategie simuliert, so liegt hier die Umsetzung der Steuerstrategie in eine konkrete Automatisierung vor, der mit SIMIT eine Simulation der rein maschinellen Aspekte der Förderanlage zur Seite gestellt wird. Wir bezeichnen diese Art von Simulation daher hier im Folgenden zur Unterscheidung als Fördertechniksimulation.

Die Fördertechniksimulation mit Hilfe der Bibliothek *CONTEC* lässt sich, wie in SIMIT üblich, einfach aus Komponenten auf der grafischen Oberfläche erstellen. Bei der Implementierung der Komponententypen der Bibliothek stand nicht die detaillierte Nachbildung der mechanischen Aspekte von Förderanlagen im Vordergrund, sondern die einfache Erstellung und Parametrierung des Simulationsmodells sowie die Stabilität dieses Modells in der Simulation. Der Fördertechniksimulation liegt dazu ein Bewegungsmodell zugrunde, das die Bewegung der Fördergüter entlang von Bahnen nachbildet, wobei die Bahnen durch die Fördermittel vorgegeben sind und die Bewegung eines Förderguts über diese Bahnen mit vorgegebener Geschwindigkeit erfolgt. Die Masse des Förderguts und andere Einflüsse auf die Bewegung, wie beispielsweise Reibung, werden vernachlässigt.

Mit den in der Bibliothek *CONTEC* bereitgestellten Komponententypen können zwei Arten von Fördersystemen simuliert werden:

- Fördersysteme mit schienengebundenen Fahrzeugen
  wie Elektrohängebahnen, Bodentransportsysteme etc. und
- fahrzeuglose Fördersysteme
  wie Ketten-, Rollen-, Gurt- und Bandfördersysteme etc.

Weiter stehen in *CONTEC* auch Komponententypen zur Verfügung, mit deren Hilfe Identifikationssysteme wie RFID, Moby, Barcode, etc. in die Simulation einbezogen werden können.

Mit SIMIT können Sie eigene Bibliothekskomponenten für Fördertechniksimulationen erstellen und so Ihre Fördertechnikbibliothek spezifisch erweitern. Das spezifische Lösungsverfahren von *CONTEC* zur Fördertechniksimulation können Sie dabei in den Komponententypen über spezielle Verbindungstypen nutzen.

Hinweis

Sind in Ihrem Simulationsprojekt Fördertechnikkomponenten vorhanden, also Komponenten, die das Lösungsverfahren für Fördertechnik nutzen, dann kann die Simulation nur gestartet werden, wenn Sie die Lizenz für Fördertechnik besitzen.

8.3.2 Fördertechniksimulation

Unter Fördertechnik versteht man die Technik des Fortbewegens von Gütern in beliebiger Richtung über begrenzte Entfernungen. Die Förderung kann im Allgemeinen beliebig im Raum erfolgen. Als Fördermittel werden die einzelnen Fördermaschinen bezeichnet, mit denen das Fördergut, also das Material, in der Förderanlage bewegt wird.

Die Fördertechnikbibliothek *CONTEC* ermöglicht es, Simulationsmodelle einer Förderanlage zu erstellen und damit Simulationen durchzuführen. Für die Fördermittel bestehen die folgenden Einschränkungen:

- Alle Transportwege werden als linienförmige Bahnen vorgegeben.
- Die Transportwege und mögliche Verzweigungen müssen als Struktur der Förderanlage für eine Simulation fest vorgegeben sein.

Weiter ist die Simulation von Fördergütern wie folgt eingeschränkt:

- Als Fördergut wird nur Stückgut betrachtet, kein Schüttgut.
- Alle Fördergüter werden mit einem quaderförmigen Umriss angenommen.
- Außer den Abmessungen werden keine weiteren physikalischen Eigenschaften des Fördergutes wie Masse, Reibung etc. berücksichtigt.

Die in der Bibliothek *CONTEC* enthaltenen Komponententypen sind in zwei verschiedene Kategorien unterteilt:

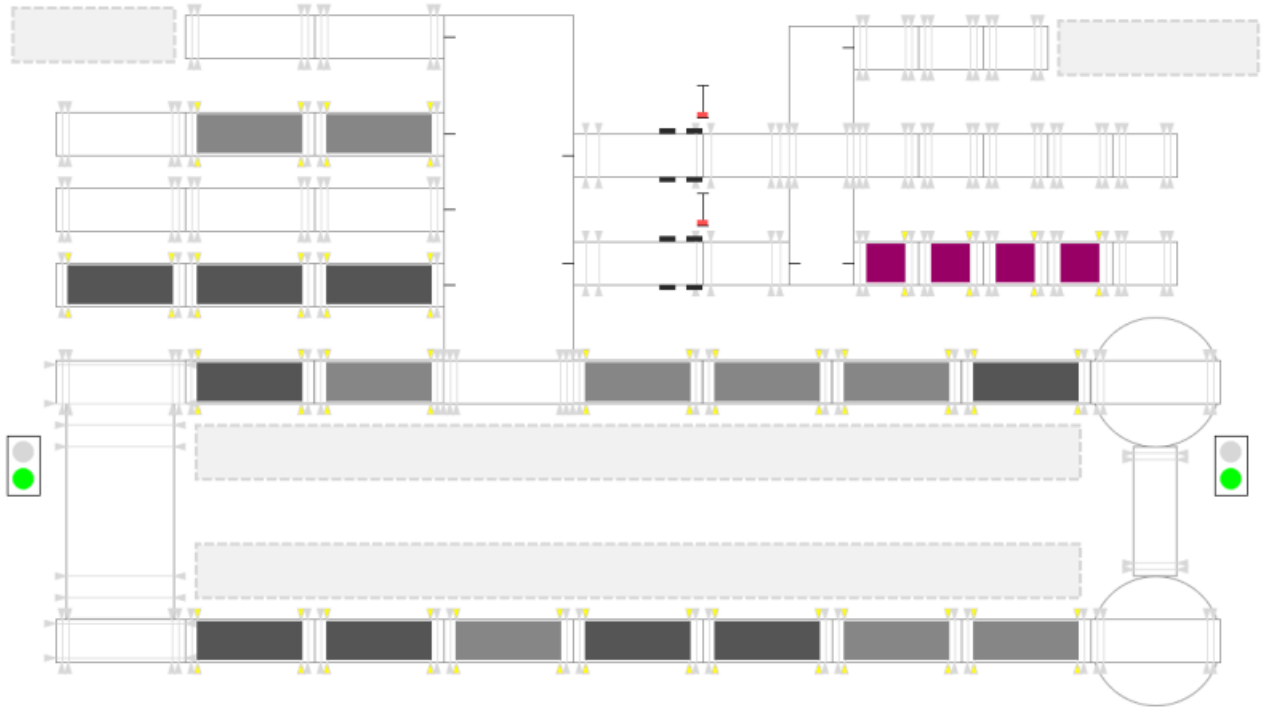
- Komponententypen von Fördergütern und
- Komponententypen von Fördermitteln.

Wie in SIMIT üblich, werden die Komponententypen von Fördermitteln mit Hilfe ihres Symbols als Komponenten in Diagramme eingefügt. Die Symbole dieser Komponententypen sind dazu so gestaltet, dass sie zusammenfügt einen maßstäblichen Grundriss (Layout) der Förderanlage ergeben, wie es beispielhaft in der Abbildung unten zu sehen ist. Das so entstehende Anlagenmodell muss lediglich noch mit passenden Parametern versehen und mit der Automatisierung verknüpft werden.

Die für ein Simulationsprojekt definierten Fördergüter werden ergänzend zum Anlagenmodell in Form von Tabellen gehalten. Auch für die Komponententypen von Fördergütern sind Symbole definiert, mit denen die einzelnen Fördergüter bei laufender Simulation visuell korrekt im Grundriss der Förderanlage dargestellt werden.

Die Fördertechniksimulation basiert auf einem speziellen Lösungsverfahren, das durch die einzelnen Komponenten des Simulationsmodells konfiguriert und parametriert wird. Dieses Lösungsverfahren basiert auf einem Bewegungsmodell das zu jedem Zeitpunkt der Simulation die Position der Fördergüter auf den durch die Fördermittel vorgegebenen Bahnen ermittelt. Das Lösungsverfahren berücksichtigt dabei, dass es zum Aufstauen von Fördergütern kommen kann und es aktiviert anhand der Position und Abmessungen der Fördergüter auch die Sensoren in der Simulation.

Die Bewegung eines Förderguts auf einer Bahn kann in der Simulation sowohl von den Fördermittelkomponenten, als auch von den Fördergutkomponenten veranlasst werden. Damit können sowohl Fördersysteme wie beispielsweise Rollenfördersysteme simuliert werden, bei denen der Antrieb auf die Fördermittel wirkt, als auch schienengebundene Fördersysteme, bei denen der Antrieb auf das Fahrzeug, also das Fördergut wirkt.



8.3.2.1 Grundlagen der Fördertechniksimulation

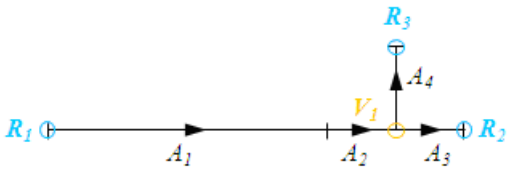
Das Verfahren zur Fördertechniksimulation beruht auf einem Bewegungsmodell, bei dem angenommen wird, dass sich die Fördergüter längs linienförmiger Bahnen bewegen, wobei die möglichen Bahnen durch die Fördermittel vorgegebenen sind. Durch die Anordnung aller Fördermittel wird so ein Streckennetz definiert, das die möglichen Transportwege für die Fördergüter in einer Förderanlage repräsentiert.

Komplexere Bewegungen können durch selbstdefinierte Komponenten realisiert und mit dem System linienförmiger Bahnen verbunden werden. Weitere Informationen hierzu finden Sie im Abschnitt: Verbindungstyp MT8 zur Übergabe von Fördergütern an Randpunkten (Seite 936).

Ein Streckennetz besteht aus einzelnen Strecken und Verzweigungen. Es kann in sich geschlossen oder offen sein. Bei einem in sich geschlossenen Netz schließt jede Strecke mit beiden Endpunkten an einer Verzweigung an, bei einem offenen Netz ist mindestens eine Strecke mit einem Endpunkt nicht an einer Verzweigung angeschlossen und bildet dort einen Randpunkt des Netzes. Das minimale Streckennetz besteht aus einer Strecke.

Eine Strecke des Netzes setzt sich aus einem oder mehreren Abschnitten (Teilstrecken) zusammen. Die Geometrie eines Abschnitts kann entweder als gerade Bahn gegebener Länge oder als Kreisbogen mit gegebenem Radius und Winkel vorliegen. Weiter ist jeder Abschnitt auch gerichtet. Die Richtung eines Abschnitts stellt die Bezugsrichtung für die Bewegung des Förderguts über diesen Abschnitt dar und ist so definiert, dass ein positiver Geschwindigkeitswert einen Transport des Förderguts in Bezugsrichtung bewirkt, ein negativer Geschwindigkeitswert einen Transport des Förderguts entgegen der Bezugsrichtung.

In folgender Abbildung ist zur Veranschaulichung ein Streckennetz aus drei Strecken bzw. vier Abschnitten A_1 bis A_4 und einer Verzweigung V_1 schematisch dargestellt.



Offensichtlich handelt es sich bei diesem Streckennetz um ein offenes Netz mit den drei Randpunkten R_1, R_2 und R_3. Beim Überfahren der Ränder eines Streckennetzes geht das Fördergut verloren, d. h. es wird vom Fördermittel entfernt und steht in der Materialliste wieder ganz allgemein zur Verfügung. Über geeignete Verbindungen zum Lösungsverfahren können Komponenten Fördergüter an den Rändern aus dem Streckennetz nehmen oder auch in das Streckennetz einspeisen.

Das spezielle Lösungsverfahren der Fördertechniksimulation berechnet die Positionen der Fördergüter, wie in SIMIT üblich, zu äquidistanten Zeitpunkten. Es ergänzt das Standardlösungsverfahren von SIMIT, so dass Fördertechnikkomponententypen ergänzend zu anderen Komponententypen, wie beispielsweise den Komponententypen der Basisbibliothek eingesetzt werden können. Für den Datenaustausch sind Fördertechnikkomponenten über spezifische Verbindungen an das Fördertechniklösungsverfahren angebunden. Sie erhalten über diese Verbindungen beispielsweise Werte, die vom Lösungsverfahren berechnet worden sind und geben ihrerseits Größen an das Lösungsverfahren.

Für jeden Abschnitt im Streckennetz können Positionen definiert sein, an denen Sensoren das dort vorbei bewegte Fördergut detektieren. Das Fördertechniklösungsverfahren sorgt dann in der Simulation dafür, dass entsprechend den Positionen und Abmessungen der Fördergüter die betroffenen Sensorsignale aktiviert werden.

8.3.2.2 Modellierung der Fördergüter

Fördergüter werden in SIMIT als Komponententypen modelliert. Für jeden Typ müssen mindestens die Größe und die grafische Darstellung des Förderguts festgelegt sein. Anschlüsse, Parameter, Zustände und Verhalten für ein Fördergut können ebenfalls in seinem Komponententyp definiert sein.

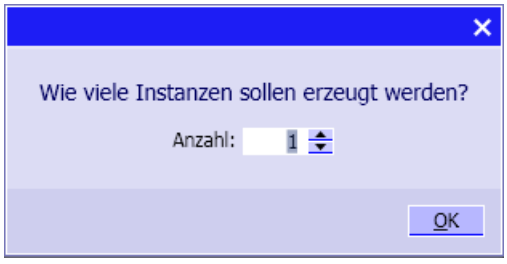
Für die Darstellung in Diagrammen ist das Verknüpfungssymbol priorisiert: falls ein Verknüpfungssymbol definiert ist, wird dieses zur Darstellung verwendet, andernfalls wird das Grundsymbol verwendet, wobei eventuell am Symbol vorhandene Anschlüsse ausgeblendet sind.

Zur Simulation von Fördergütern muss im SIMIT-Projekt der verfügbare Vorrat an Fördergutkomponenten definiert werden. Wählen Sie hierzu im Projektfenster unter dem Projektordner "Listen" den Baumeintrag "Neue Liste" aus. Damit erstellen Sie eine Materialliste, d. h. eine Liste für die in der Simulation verfügbaren Fördergutkomponenten.

Der Editor zum Anlegen der Materialliste wird geöffnet:



Hier können Sie nun definieren, welche Fördergutkomponenten in der Simulation zur Verfügung stehen. Für jeden Komponententyp können Sie die Anzahl der zu bildenden Instanzen festlegen. Um eine Instanz anzulegen, ziehen Sie einfach den gewünschten Komponententyp aus dem Verzeichnis *MATERIAL* der Bibliothek *CONTEC* in den Editor. Wenn Sie mehr als eine Instanz anlegen möchten, halten Sie beim Ziehen zusätzlich die Alt-Taste gedrückt. Es erscheint dann ein Abfragedialog zur Anzahl der gewünschten Instanzen wie in folgender Abbildung zu sehen:



Sie können im Projektordner "Listen" auch mehrere Materiallisten anlegen und die Materiallisten gegebenenfalls in verschiedene Ordner einsortieren.

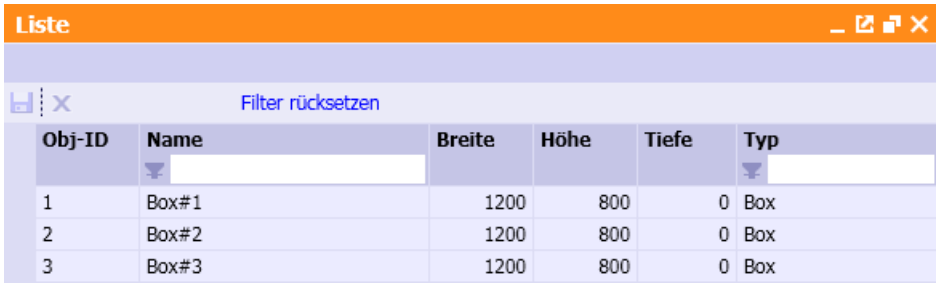
Um für den Start der Simulation Fördergutkomponenten auf einer konkreten Strecke des simulierten Transportnetzes aufzusetzen, können Sie der Komponente, die die Strecke simuliert eine oder mehrere Fördergutkomponenten gleichen Typs aus den im Projekt vorhandenen Materiallisten zuordnen.

Beispiele dafür finden Sie in den folgenden Kapiteln:

- Rail-S4 – Gerade Schiene mit vier Sensoren (Seite 840)
- Conveyor-S4 – Geradförderer mit vier Sensoren (Seite 856)

Falls Fördergutkomponenten ein Verhaltensmodell (Verhaltensbeschreibung) besitzen, wird dieses mit dem Start der Simulation genauso ausgeführt als wären die Fördergutkomponenten wie andere Komponenten auch auf einem Diagramm platziert. Die Simulation des dadurch definierten Verhaltens eines Förderguts erfolgt völlig unabhängig von der Behandlung dieses Förderguts im Lösungsverfahren zur Fördertechniksimulation.

Nach dem Starten der Simulation erhält jedes in den Materiallisten definierte Fördergut eine Nummer zur eindeutigen Identifikation, die von SIMIT automatisch gesetzt und in den Materiallisten eingeblendet wird.



Obj-ID	Name	Breite	Höhe	Tiefe	Typ
1	Box#1	1200	800	0	Box
2	Box#2	1200	800	0	Box
3	Box#3	1200	800	0	Box

Wird in der Simulation Fördergut an einem Sensor detektiert, dann liefert das Lösungsverfahren die ID der detektierten Fördergutkomponente.

8.3.2.3 Modellierung des Streckennetzes

Streckennetze setzen sich aus Streckenabschnitten, Verzweigungen und Rändern zusammen. Für die Modellierung des Streckennetzes von Förderanlagen stehen in der Bibliothek *CONTEC* verschiedene Komponententypen zur Verfügung, die elementare Fördermittel repräsentieren und dazu entsprechende Strukturen aus Streckenabschnitten, Verzweigungen und Berandungen kapseln. Diese in den Komponententypen modellierten Teilstreckennetze sind dementsprechend von unterschiedlicher Komplexität. Beispielsweise ist in einem Komponententyp, der einen Geradförderer nachbildet, nur ein einzelner Streckenabschnitt modelliert. Komponententypen, die komplexere Fördermittel wie beispielsweise einen Eckumsetzer nachbilden, beinhalten hingegen ein Modell aus mehreren Streckenabschnitten und Verzweigungen.

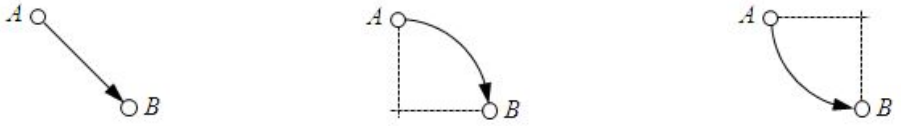
Die einzelnen Komponententypen haben Anschlüsse, die Endpunkte von Streckenabschnitten oder auch Verzweigungspunkte repräsentieren. Verbindet man die Anschlüsse von zwei Komponenten dieser Typen auf Diagrammen, so werden die Teilstreckenmodelle der beiden Komponenten entsprechend verbunden. Durch Verbinden mit weiteren Komponenten wird so schließlich das gesamte Streckennetz einer Förderanlage aufgebaut.

Für jeden Streckenabschnitt müssen die Geometrie mit Koordinaten und die Bezugsrichtung gegeben sein. Optional können für einen Streckenabschnitt noch Positionen von Blockern und Sensoren definiert sein. Für jede Verzweigung muss definiert sein, welche Streckenabschnitte sie verbindet und sie müssen mit entsprechenden Angaben zu ihren Positionen im Streckennetz versehen sein.

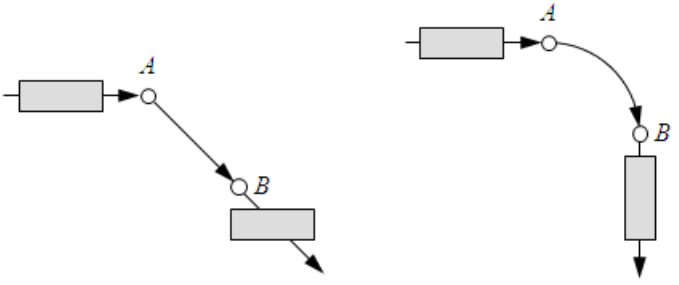
Modellierung der Streckenabschnitte

Eine Linie bildet das Modell eines Streckenabschnitts. Sie wird definiert durch die beiden Endpunkte, die Geometrie und die Richtung des Streckenabschnitts. Die Geometrie kann als Gerade oder als Kreisbogen gegeben sein. Sie wird durch die Position der beiden Endpunkte *A* und *B* und die Angabe des Streckenwinkels festgelegt. Für Winkel verschieden von null sind zwei mögliche Strecken durch das Vorzeichen des Winkels unterscheidbar: ein positiver Winkel bewirkt, dass der Mittelpunkt des Kreisbogens rechts der Teilstrecke in Zählrichtung gesehen liegt, ein negativer Winkel bewirkt, dass der Mittelpunkt des Kreisbogens links der Teilstrecke liegt. Der Radius des Kreisbogens ergibt sich aus diesen drei Größen. Die Richtung des Streckenabschnitts entspricht der Bezugsrichtung für den Transport der Fördergüter.

Winkel 0 Winkel +90 Winkel −90



Das über einen Streckenabschnitt transportierte Fördergut wird um den Winkel der Strecke gedreht. Offensichtlich wird somit auf geraden Streckenabschnitten das Fördergut ohne Drehung transportiert, die absolute Ausrichtung des Förderguts bleibt erhalten. In folgender Abbildung ist der Transport von Fördergut über einen geraden bzw. gekrümmten Streckenabschnitt veranschaulicht:

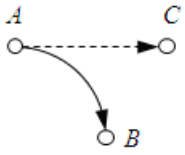


Für jeden Streckenabschnitt ist eine Geschwindigkeit definiert, mit der Fördergüter über ihn transportiert werden. Positive Geschwindigkeitswerte bewirken einen Transport von Fördergut in Bezugsrichtung, negative Geschwindigkeitswerte einen Transport von Fördergut entgegen der Bezugsrichtung. Eine eventuell gegebene Eigengeschwindigkeit für ein Fördergut wird zur Geschwindigkeit des Streckenabschnitts addiert.

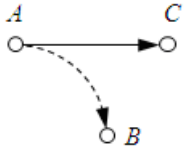
Jedem Streckenabschnitt ist jeweils für die Bewegung in Bezugsrichtung und für die Bewegung entgegen der Bezugsrichtung ein Zustand zugeordnet. Dieser Zustand kennzeichnet einen Streckenabschnitt in die jeweilige Richtung als "aktiv" oder "inaktiv". Inaktive Strecken werden im Fördertechniklösungsverfahren als nicht existent betrachtet, d. h. es können weder Fördergüter auf diese Abschnitte verschoben werden, noch werden für Fördergüter, die bereits auf diesen Abschnitten liegen, die Positionen aktualisiert. Es werden auf inaktiven Streckenabschnitten auch keine Sensoren aktualisiert. Das Umschalten des Zustands von Streckenabschnitten wird benutzt, um an Verzweigungen einen eindeutigen Transportweg festzulegen. In folgender Abbildung ist beispielhaft skizziert, wie durch wechselseitiges

Aktivieren und Deaktivieren der beiden Streckenabschnitte in Bezugsrichtung eine Weichenfunktion modelliert werden kann:

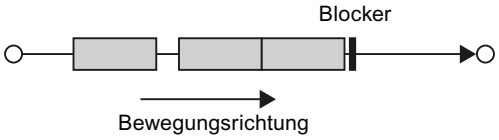
Abschnitt *AC* inaktiv



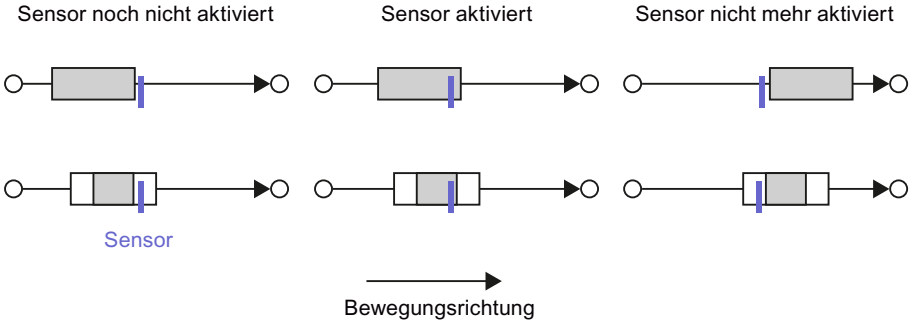
Abschnitt *AB* inaktiv



Auf einem Streckenabschnitt können Positionen definiert sein, an denen der Transport von Fördergut blockiert werden kann. Der Zustand einer solchen Blockerposition kann in den Zustand "aktiv" oder "inaktiv" gesetzt werden. Ist ein Blocker aktiv, dann werden alle Fördergüter an seiner Position angehalten und alle folgenden Fördergüter werden entsprechend aufgestaut, wie beispielhaft in folgender Abbildung skizziert:



Weiter können auf einem Streckenabschnitt Positionen als sogenannte Sensorpositionen definiert sein, an denen Fördergut detektiert werden kann. An jeder dieser Sensorpositionen wird dann im Fördertechniklösungsverfahren die Kennung (ID) des Förderguts erfasst, das diese Position überdeckt. Für jede Sensorposition kann zusätzlich der zu erfassende Bereich der Fördergüter definiert werden. Dieser Bereich wird als prozentualer Wert der Fördergutgröße und als symmetrisch zur Fördergutmitte angenommen. In der folgenden Abbildung ist die Erfassung von Fördergut für zwei unterschiedliche Erfassungsbereiche skizziert: die obere Reihe zeigt die Aktivierung eines Sensors für den vollen Erfassungsbereich (100 %) des Förderguts, die untere Reihe zeigt die Aktivierung eines Sensors für den halben Erfassungsbereich (50 %). Der Erfassungsbereich des Förderguts ist jeweils grau dargestellt; die Bewegung ist von links nach rechts angenommen.

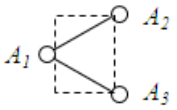
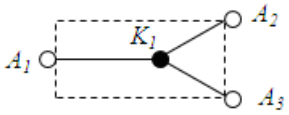


Komponententypen, in denen nur ein Streckenabschnitt mit Sensoren modelliert ist, sind beispielsweise die gerade Schiene *Rail-S4* und der Geradförderer *Conveyor-S4*.

Modellierung der Verzweigungen

Da alle Strecken als linienförmig angenommen sind, stellt eine Verzweigung einen Punkt im Streckennetz dar, an den mehr als zwei Streckenabschnitte anschließen.

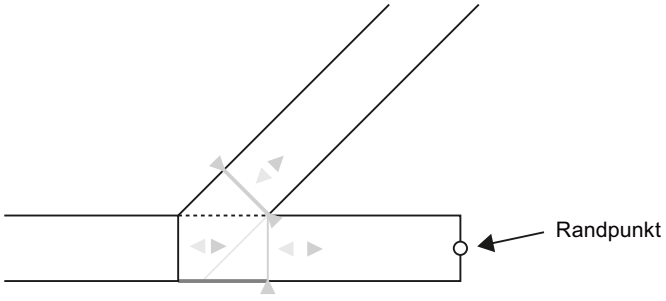
Im Teilstreckennetz eines Komponententyps kann eine Verzweigung entweder als Anschluss, d. h. als gemeinsamer Endpunkt mehrerer Streckenabschnitte gegeben sein (s. Abbildung unten links) oder innerhalb des Teilstreckennetzes liegen (s. Abbildung unten rechts). Im ersten Fall ist die Position des Verzweigungspunktes unmittelbar durch die Position des Anschlusses gegeben, im zweiten Fall ist die Verzweigungsposition relativ zur Komponente definiert.

(a) Verzweigung am Anschluss	(b) Verzweigung in der Komponente
	

An Verzweigungspunkten muss der Transportweg zu jedem Zeitpunkt der Simulation eindeutig bestimmt sein, d. h. von den an einer Verzweigung anschließenden Streckenabschnitten dürfen zu jedem Zeitpunkt höchstens zwei Abschnitte als Transportweg aktiv sein. Im Komponententyp muss folglich die Aktivierung der Streckenabschnitte funktional enthalten sein.

Modellierung der Ränder

Ränder sind Endpunkte eines Streckenabschnitts, die nicht mit anderen Abschnitten verbunden sind. Für eine Komponente wie beispielsweise den Geradförderer entsteht so ein Randpunkt des Streckennetzes, wenn ein Anschluss dieser Komponente nicht mit anderen Komponenten verbunden ist. Wenn sich ein Fördergut in der Simulation über einen solchen Randpunkt hinaus bewegt, wird es von der Förderstrecke entfernt und steht in der Materialliste wieder zur Verfügung. In der Meldezeile wird auf diesen Vorgang entsprechend hingewiesen.



Randpunkte können aber auch als Streckenendpunkte im Teilstreckennetz von Komponententypen definiert sein. Im Komponententyp kann dann mit geeigneten Funktionen das Verhalten von Fördergut, das diese Randpunkte passiert, frei definiert werden. Beispielsweise kann die Position des Förderguts völlig frei berechnet werden, um komplexe Bewegungen nachbilden zu können. Beispiele für solche Komponenten finden Sie in den folgenden Kapiteln:

- Turntable-R60 – Drehtisch (Seite 879)
- TransferCarriage – Querverschiebewagen (Seite 883)

8.3.2.4 Besondere Aspekte der Fördertechniksimulation

Aufsetzen und Entfernen von Fördergütern

Fördergüter, die im verfügbaren Materialvorrat des Simulationsprojekts definiert sind, können in der Simulation auf jeden Streckenabschnitt aufgesetzt werden. Der Aufsetzpunkt ist der Beginn des Streckenabschnitts entsprechend der angegebenen Bezugsrichtung (FROM-TO), wobei Fördergüter stets mit ihrem geometrischen Mittelpunkt aufgesetzt werden.

Fördergüter können auch aus dem Streckensystem entfernen werden. Für das Aufsetzen wie auch das Entfernen von Fördergütern steht ein entsprechender Systemaufruf zur Verfügung, der die Nachbildung dieser Funktion in Komponententypen ermöglicht.

Simulation des Stauverhaltens

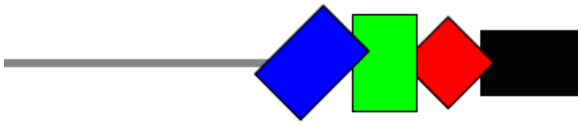
Aufstauen auf geraden Förderstrecken

Das Stauverhalten ist stark vereinfacht modelliert. Die Positionen von in der Simulation aufgestauten Fördergütern entsprechen damit nicht in allen Fällen den realen Positionen.

Das Stauverhalten auf geraden Strecken ist korrekt für alle Fördergüter, die entweder gar nicht oder um Vielfache von 90° gedreht sind.



Fördergüter mit anderen Drehwinkeln ϕ werden auch aufgestaut. Da deren Position allerdings nur näherungsweise berechnet wird, kommt es zu Überlappungen.



Die für das Aufstauen vereinfacht angenommene wirksame Breite $\tilde{b}$ eines Objektes wird nach folgender Formel berechnet:

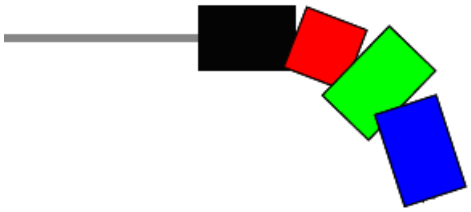
$$\tilde{b} = \frac{\varphi h + (\pi - 2\varphi) b}{\pi}$$

Dabei ist *h* die Höhe und *b* die Breite des Objekts.

Aufstauen in Kurven

Das Stauverhalten ist stark vereinfacht modelliert. Die Positionen von in der Simulation aufgestauten Fördergütern entsprechen damit nicht in allen Fällen den realen Positionen.

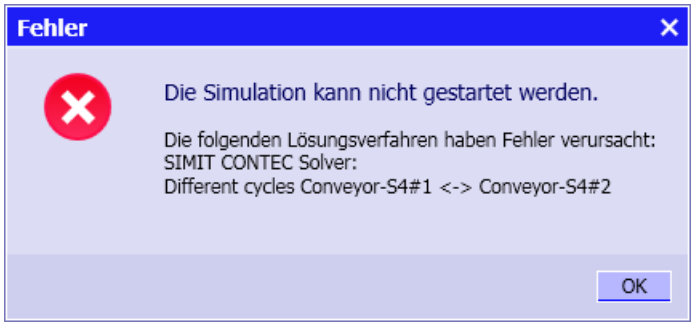
In Kurven wird die Position aufgestauter Fördergüter ebenfalls nur näherungsweise bestimmt. Die aufgestauten Fördergüter überlappen, wie in folgender Abbildung zu sehen.



Die Überlappungen kommen dadurch zustande, dass für die Wegberechnung die Krümmung der Kurve vernachlässigt wird.

Zeitscheibenzuordnung von Komponenten

Alle Komponenten, die ein zusammenhängendes Streckennetz bilden, müssen derselben Zeitscheibe zugeordnet sein. Folglich müssen alle Komponenten, die Fördermittel simulieren und direkt miteinander über Konnektoren oder Signallinien verbunden sind, derselben Zeitscheibe zugeordnet sein. Andernfalls wird das Simulationsmodell eines zusammenhängenden Streckennetzes aus mehreren Teilmodelle gebildet, die unterschiedliche Zykluszeiten haben. Das Starten der Simulation wird dann mit einer entsprechenden Fehlermeldung verweigert.

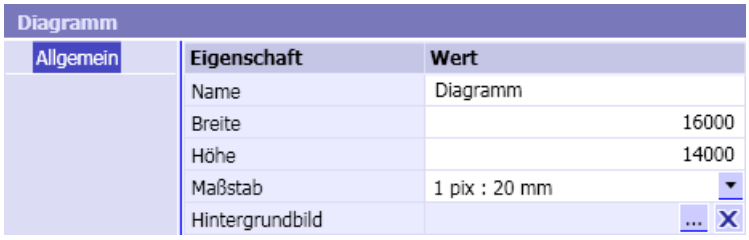


8.3.2.5 Maßstäblichkeit

Beim Einsatz der Fördertechnikbibliothek spielen im Gegensatz zur Standardbibliothek die Abmessungen der Komponenten eine entscheidende Rolle, da sich daraus unmittelbar die Längen von Förderstrecken bzw. die Abmessungen der Fördergüter ergeben. Als Längeneinheit werden orientiert an maschinenbaulichen Gepflogenheiten durchgängig Millimeter [mm] verwendet, Geschwindigkeiten werden in Meter pro Sekunde [m/s] angegeben.

Maßstäblichkeit von Förderstrecken

Um Förderanlagen verschiedenster Größe mit angemessener Auflösung auf Diagrammen abbilden zu können, kann jedem Diagramm per Auswahl ein Maßstab zugewiesen werden.



Einstellbar sind die Maßstäbe

1 pix : 1 mm,

1 pix : 5 mm,

1 pix : 10 mm,

1 pix : 20 mm,

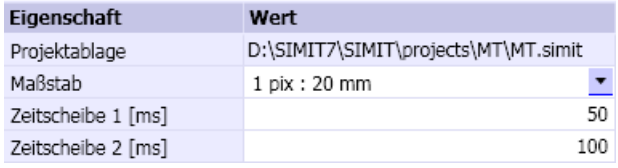
1 pix : 50 mm,

1 pix : 100 mm.

Hinweis

Wenn Sie die Fördertechnikbibliothek für Ihre SIMIT-Installation lizenziert haben, werden die Größen von allen Komponenten, also auch von den Komponenten der Basisbibliothek mit dem eingestellten Maßstab wiedergegeben. Für die Komponenten der Basisbibliothek erhalten Sie die vorgesehenen Größen, wenn Sie für den Maßstab von Diagrammen mit Basiskomponenten die Einstellung 1 pix : 1 mm wählen.

Damit Sie nicht für jedes Diagramm den Maßstab einzeln anpassen müssen, können Sie im Eigenschaftsfenster des Projektmanagers eine projektweit gültige Voreinstellung wählen.



Dieser Maßstab wird als Voreinstellung für alle Diagramme übernommen, die Sie neu anlegen.

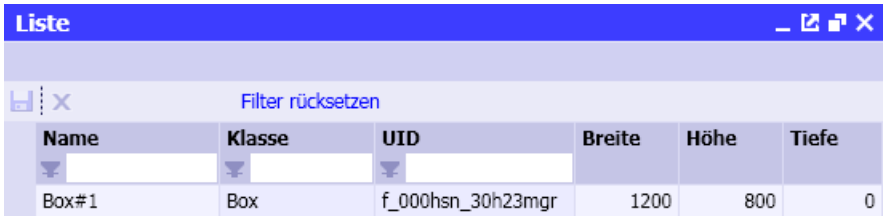
Hinweis

Der Maßstab bereits vorhandener Diagramme wird durch die Voreinstellung des Maßstabs im Projektmanager nicht verändert.

Beachten Sie, dass die Veränderung des Maßstabs eines Diagramms nicht die dargestellte Größe einer Komponente verändert, sondern deren wirksame Abmessung in Millimetern.

Maßstäblichkeit von Fördergütern

Fördergüter werden in den Materiallisten eines SIMIT-Projekts mit ihrer absoluten Größe (Breite und Höhe) in Millimetern vorgehalten. Auf Diagrammen werden Fördergüter somit je nach eingestelltem Maßstab des Diagramms unterschiedlich groß dargestellt.



Die Abmessungen von Fördergut sind im Komponententyp voreingestellt. Wenn Sie Fördergut in die Materialliste einfügen, dann wird es mit der durch den Komponententyp vorgegebenen Breite, Höhe und Tiefe eingefügt. Sie können dann die Abmessung für jedes Fördergut in der Materialliste verändern, sofern die Komponente skalierbar ist. Die mit der Bibliothek CONTEC zur Verfügung gestellten Typen von Fördergütern sind alle skalierbar.

Hinweis

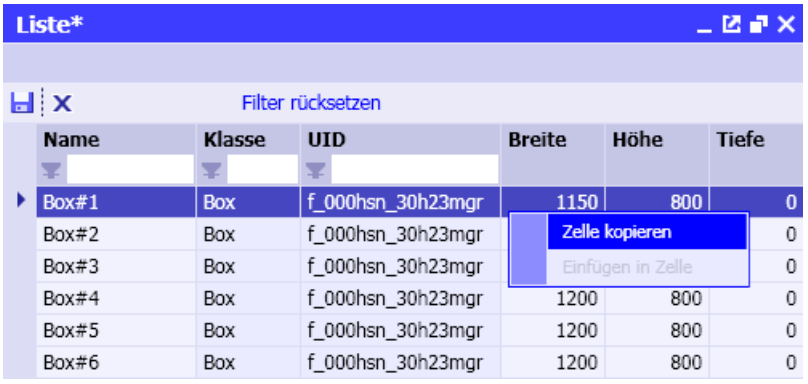
Beachten Sie, dass mit *Breite* und *Höhe* die Abmessungen des Förderguts in der Draufsicht wie im Diagramm dargestellt gemeint sind. Die *Tiefe* gibt die Ausdehnung in der nicht dargestellten dritten Dimension wieder. Diese Definition wurde gewählt, um die Bedeutung von Höhe und Breite von Komponenten und Grafiken auf einem Diagramm beibehalten zu können. Die mit der Fördertechnikbibliothek zur Verfügung gestellten Komponententypen nutzen die Tiefe von Fördergut nicht. Fördergut egal welcher Tiefe wird mit den in der Bibliothek zur Verfügung stehenden Komponententypen über Sensoren ausschließlich über die Breite und Höhe detektiert. Tiefenangaben für Fördergüter sind daher hier mit null vorbelegt. Die Tiefe von Fördergütern gewinnt erst dann an Bedeutung, wenn Sie einen eigenen Typ einer Fördertechnikkomponente erstellen und in Ihrer Fördertechniksimulation einsetzen, der die Tiefe von Fördergut auswertet, also beispielsweise eine Höhenkontrolle nachbildet.

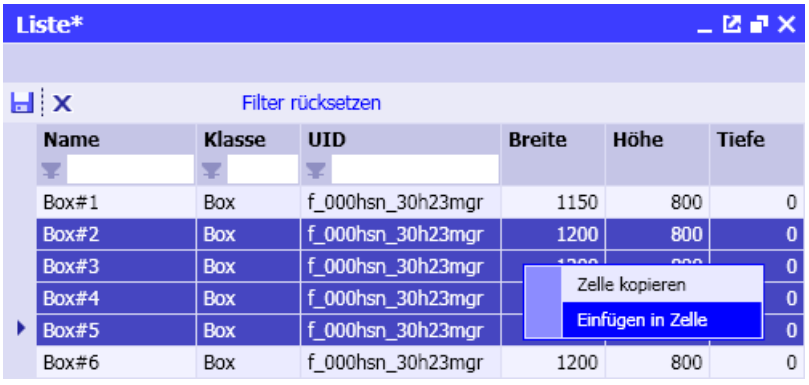
Weitere Informationen hierzu finden Sie im Kapitel: Komponententypen für Fördertechniksimulation selbst erstellen (Seite 925).

Wenn Sie die Abmessungen mehrerer Fördergüter auf den gleichen Wert setzen möchten, können Sie den Wert von einem Fördergut, also aus einer Zeile dieser Tabelle auf beliebig viele andere übertragen. Gehen Sie dazu wie folgt vor:

- Selektieren Sie die Zeile des Förderguts, das den gewünschten Wert enthält.
- Klicken Sie mit der rechten Maustaste auf die Zelle, die den zu kopierenden Wert enthält und wählen Sie "Zelle kopieren" im Kontextmenü.

- Selektieren Sie die Zeilen, in die Sie den Wert übertragen möchten.
- Klicken Sie mit der rechten Maustaste auf die Spalte, in die Sie den Wert übertragen möchten und wählen Sie "Einfügen in Zelle" im Kontextmenü.





8.3.2.6 Generierung der Simulation von Antrieben und Sensoren

Da Förderstrecken mit Hilfe von entsprechenden Komponenten im Sinne eines Anlagenlayouts auf den Diagrammen maßstäblich dargestellt werden, ist es nicht sinnvoll, zusätzlich alle weiteren Ein- und Ausgangssignale dieser Komponenten grafisch in diesen Diagrammen darzustellen. Die Übersichtlichkeit des Anlagen-Layouts würde dadurch zu sehr beeinträchtigt. Die Komponententypen der *CONTEC*-Bibliothek besitzen daher als sichtbare Anschlüsse nur die Anschlüsse über die sie zum Aufbau des Anlagenlayouts miteinander verbunden werden. Die Anschlüsse von Komponenten zur Verbindung mit den zugehörigen Antriebs- und Sensorsignalen, und damit über die Kopplungen letztendlich mit den Signalen der Steuerung, erfolgt über eine implizite Belegung der Eingänge. Ihnen stehen verschiedene Möglichkeiten zur Verfügung, diese Verschaltung vorzunehmen, die im Folgenden am Beispiel einer einfachen Schiene erläutert werden.

Manuelle Verschaltung der Komponenten

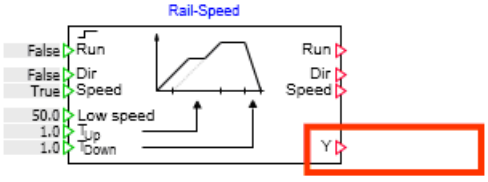
Der Komponententyp einer Schiene ist so angelegt, dass die Komponente ihren prozentualen Geschwindigkeitswert als Eingangswert vom Ausgang *Y* einer anderen Komponente erhält, deren Namen sich aus dem Namen der Schiene und dem Zusatz *Speed* zusammensetzt. In folgender Abbildung ist beispielsweise eine Schiene mit Namen *Rail* dargestellt.



Ihr Eingang *Speed* ist folglich so voreingestellt, dass er mit dem Ausgang *Y* der Komponente mit dem Namen *Rail-Speed* implizit verschaltet ist.



Sie können also ein Diagramm erstellen, das in diesem Beispiel eine Komponente mit dem Namen *Rail-Speed* enthält und einen Ausgang *Y* besitzt, der dann die prozentuale Geschwindigkeit für diese Schiene vorgibt.



Alternativ können Sie auch jeden anderen (analogen) Ausgang einer Komponente im Eingang *Speed* der Schiene eintragen, indem Sie die Einstellung von {$} auf ändern und das entsprechende Ausgangssignal eintragen, wie beispielhaft in folgender Abbildung zu sehen.



Sollte die von der Schiene vorgegebene Geschwindigkeit konstant sein, können Sie die Einstellung des Eingangs *Speed* auch auf 123 stellen und den gewünschten Prozentwert direkt eingeben.



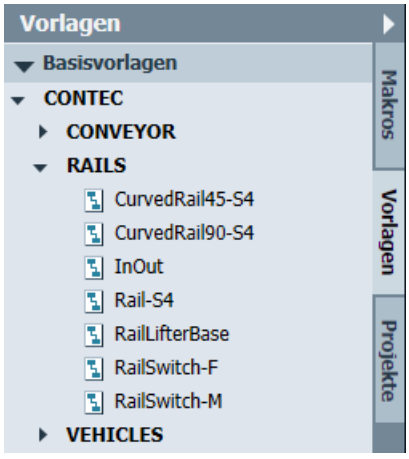
Um Ausgangssignale für die Sensorik der Schiene weiterverarbeiten zu können, nutzen Sie ebenfalls implizite Verschaltungen. Sie können beliebige Komponenten verwenden, und deren (binäre) Eingänge entsprechend belegen, wie beispielhaft in folgender Abbildung zu sehen.



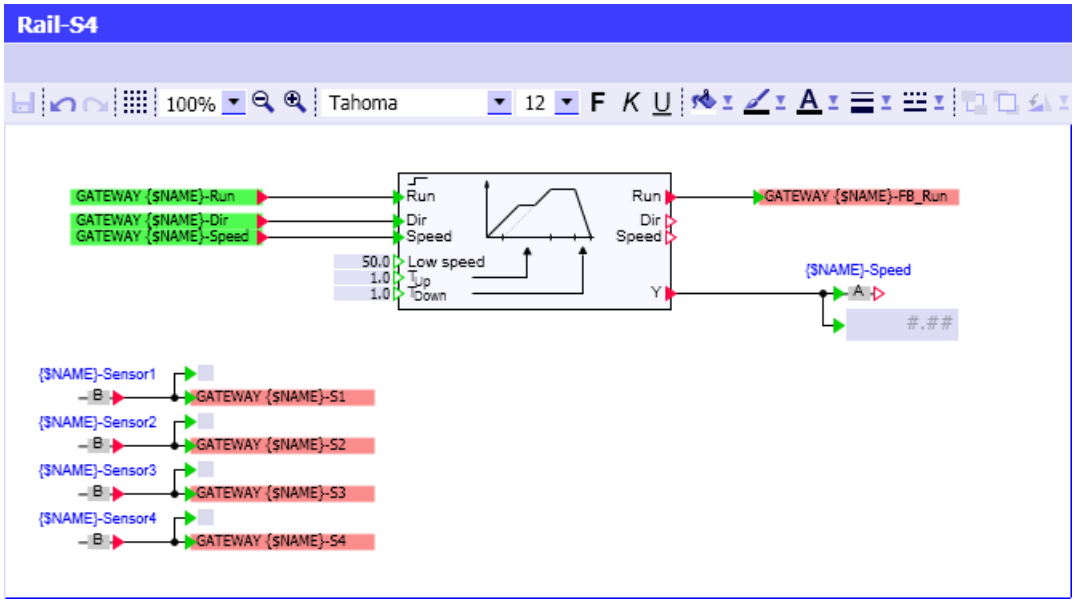
Verwendung von Vorlagen

Um mit minimalem Aufwand die Diagramme für die Antriebs- und Sensorsimulation zu erstellen, können Vorlagen verwendet werden.

Mit der Fördertechnik-Bibliothek werden Vorlagen ausgeliefert, die zu jedem Komponententyp einer Förderstrecke wie auch für die Fahrzeuge eine passende Vorlage zur Anbindung der Antriebs- und Sensorsignale zur Verfügung stellen. Die Namen der Vorlagen sind identisch mit den Namen der Komponententypen.



Die Vorlagen enthalten alle eine Komponente aus der Basisbibliothek von SIMIT zur Simulation des Antriebs. Zusätzlich können über die Vorlage die simulierten Sensoren der Förderstrecke über Kopplungssignale an die Steuerung angebunden werden. Für den Komponententyp "Rail-S4" einer Schiene ist die Vorlage in folgender Abbildung zu sehen.



Falls Sie ein Kennzeichnungssystem benutzen, mit dem sich die symbolischen Namen der E/A-Adressen aus den Namen der Förderstrecken ableiten lassen, können Sie diese Vorlagen entsprechend anpassen. Sie erhalten dann durch die Instanziierung einer solchen Vorlage die vollständige Anbindung der entsprechenden Förderstrecke über die Kopplung an die Steuerung.

Um alle Förderstrecken im Simulationsprojekt möglichst aufwandsarm an die Steuerung anzuschließen, verwenden Sie die Funktion "Vorlagen instanziieren".

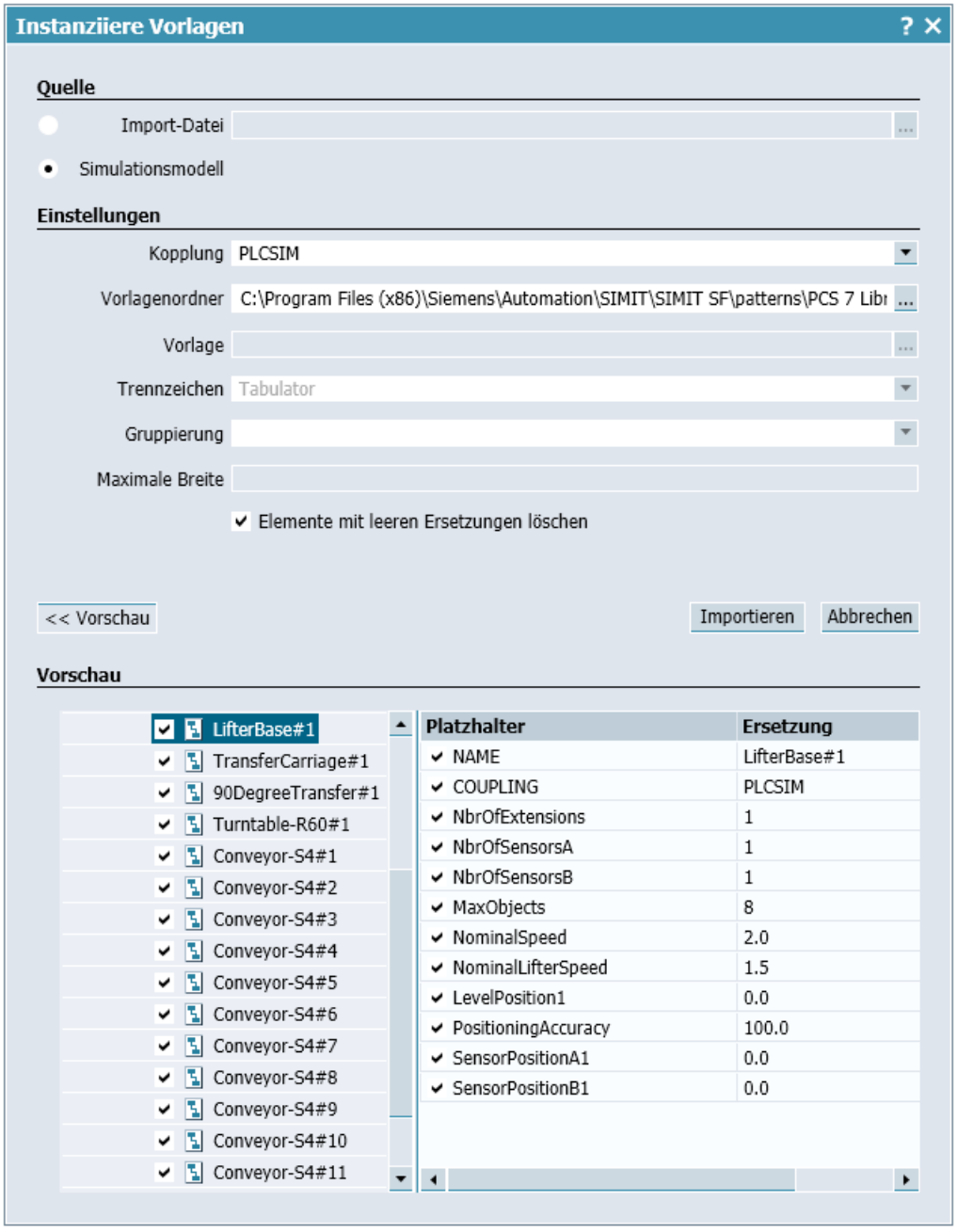
Es gibt folgende Möglichkeiten, diese Funktion aufzurufen:

- Wählen Sie in der Portalansicht "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie in der Projektansicht den Menübefehl "Automatische Modellerstellung > Vorlagen instanziieren".
- Wählen Sie im Kontextmenü eines Diagramm-Ordners "Automatische Modellerstellung > Vorlagen instanziieren".

Mit der Funktion "Vorlagen instanziieren" > "Simulationsmodell" erzeugen Sie für jede Komponente in Ihrem Simulationsprojekt, die einen Parameter *TEMPLATE* besitzt, eine Instanz einer Vorlage. Der Name der zu instanziierenden Vorlage wird mit eben diesem Parameter *TEMPLATE* bestimmt. Zusätzlich kann die Komponente über ihren Parameter *HIERARCHY* eine Ordner-Hierarchie definieren, in der die Vorlage instanziiert wird.

Die Komponenten der *CONTEC*-Bibliothek sind auf dieses Verfahren abgestimmt. Sie enthalten die (Zusatz-)Parameter *TEMPLATE* und *HIERARCHY*. Wenn Sie CONTEC-Komponenten in Ihrem SIMIT-Projekt benutzt haben, liefert die Funktion "Vorlagen instanziieren" > "Simulationsmodell" die zugehörigen Diagramme zur Anbindung der Steuerung.

Da in den Vorlagen zur Generierung der Geräteebene naturgemäß auch auf Kopplungssignale Bezug genommen wird, können Sie den Namen einer Kopplung im Import-Dialog angeben. Er ist dort unter dem Variablennamen *GATEWAY* verfügbar. Sind in Ihrem Simulationsprojekt bereits Kopplungen enthalten, so werden diese in der Auswahlliste aufgeführt.



Hinweis

Sie werden die Namen der Ein-/Ausgangssignale in den Vorlagen auf Ihr Kennzeichnungssystem anpassen müssen, um aus diesem Verfahren den maximalen Nutzen ziehen zu können.

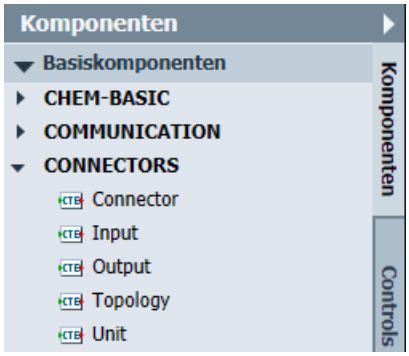
Die Mindestgröße für Vorlagen beträgt 20 × 20 Pixel.

8.3.3 Komponenten der Bibliothek CONTEC

8.3.3.1 Topologischer Konnektor in der CONTEC-Bibliothek

Im Verzeichnis *CONNECTORS* der Basisbibliothek von SIMIT steht ein Konnektor zur Verfügung, mit dem topologische Verbindungen von Fördertechnikkomponenten über Diagrammgrenzen hinweg realisiert werden können:

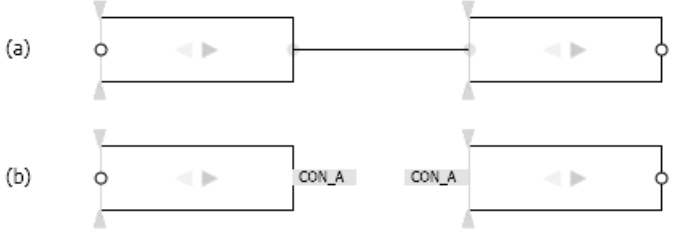
- der topologische Konnektor *Topology*.



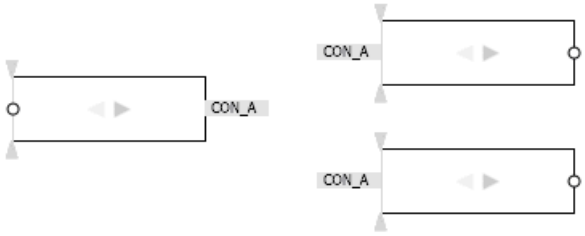
Das Symbol von *Topology* ist in folgender Abbildung dargestellt:



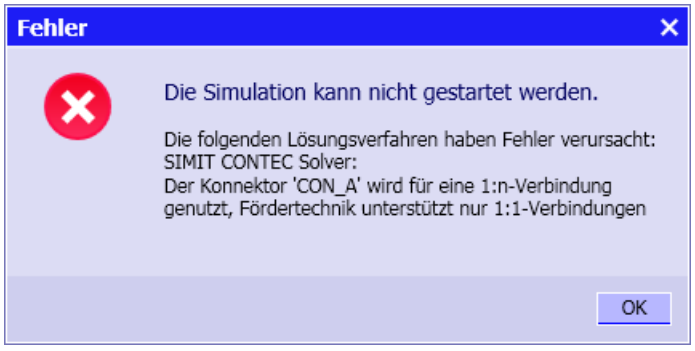
Mit dem Konnektor *Topology* kann eine topologische Verbindung zwischen zwei Fördertechnikkomponenten hergestellt werden. In der Abbildung unten unter (b) sind zwei Komponenten über den Konnektor *CON_A* verbunden. Diese Verbindung ist funktional identisch zur direkten Verbindung der beiden Komponenten über eine Verbindungslinie wie in der Abbildung unten unter (a) dargestellt.



In der folgenden Abbildung sind drei Komponenten über den Konnektor *CON_A* verbunden. Diese Konfiguration ist nicht zulässig, da bei Fördertechnikkomponenten immer nur zwei topologische Anschlüsse miteinander verbunden werden dürfen.

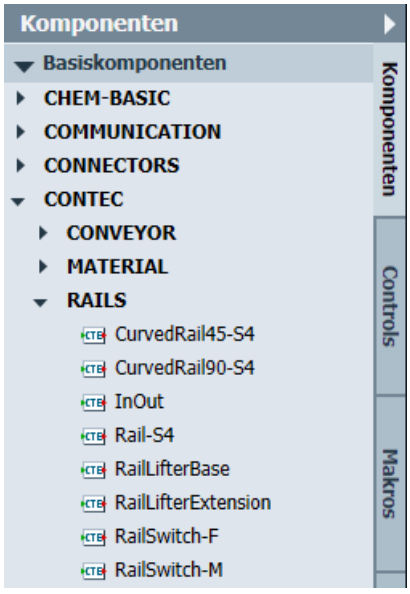


Beim Starten eines Simulationsprojekts, das eine solche Konfiguration enthält, erscheint eine Fehlermeldung wie in der folgenden Abbildung zu sehen und der Start der Simulation wird abgebrochen.



8.3.3.2 Komponententypen für Fördersysteme mit Fahrzeugen

Im Verzeichnis *RAILS* der Bibliothek CONTEC stehen Komponententypen zur Verfügung, die zur Simulation von Fahrzeugfördersystemen eingesetzt werden können. Im weitesten Sinne bilden diese Typen "Schienen" nach, auf denen Fahrzeuge als Fördergüter aufgesetzt werden können. Die Fahrzeuge werden entweder mit einer vom jeweiligen Schienenabschnitt vorgegebenen Geschwindigkeit bewegt, oder die Fahrzeuge geben diese Geschwindigkeit individuell selbst vor. Es ist auch eine Überlagerung beider Geschwindigkeiten möglich, wenngleich dieser Fall nicht praxisrelevant ist.



Rail-S4 – Gerade Schiene mit vier Sensoren

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge des Schienenstücks.

Funktion

Der Komponententyp *Rail-S4* dient zur Simulation eines geraden Schienenstücks. Die Geschwindigkeit, mit der Fahrzeuge über dieses Schienenstück bewegt werden, wird am nicht sichtbaren Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Es können zwischen einem und vier Sensoren über die Länge der Schiene angeordnet werden. Wenn sich ein Fahrzeug im Erfassungsbereich eines Sensors befindet, wird der

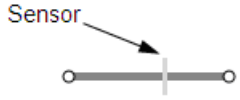
entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Fahrzeugs kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NbrOfSensors*
  Anzahl der verwendeten Sensoren (1 bis 4)
- *SensorPosition*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.

Jeder definierte Sensor wird bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.



Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NbrOfSensors		1
▼ SensorPosition [1]		...
SensorPosition1	[mm]	0.0

Während der Initialisierung der Komponente werden die Parameter auf ihre Gültigkeit geprüft. Folgende Meldungen weisen auf mögliche Fehlparametrierungen hin:

- *Parameter 'SensorPosition' must be less than the component's width*
  Die Sensorposition darf nicht außerhalb der Komponente liegen.
- *Parameter 'SensorPosition' must not be negative*
  Die Sensorposition darf nicht negativ sein.

Zusatzparameter

Die Komponente *Rail-S4* enthält die Zusatzparameter *TEMPLATE* und *HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

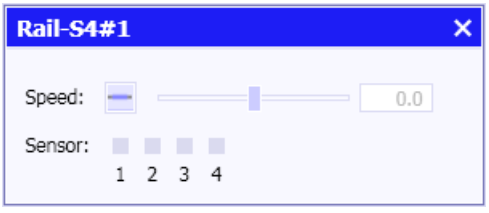
Über weitere Zusatzparameter können während der Initialisierung Fahrzeuge auf die Schiene aufgesetzt werden.

- *MaterialType*
  Der Typ des aufzusetzenden Fahrzeugs
- *MaterialList*
  Der Name der Materialliste, aus der das Fahrzeug genommen werden soll. Wenn dieser Paramter leer ist, werden alle im Simulationsprojekt angelegten Materiallisten durchsucht.
- *InitNbrOfObjects*
  Die Anzahl der initial aufzusetzenden Fahrzeuge
- *Clearance*
  Der Abstand zwischen den initial aufzusetzenden Fahrzeugen

Name		Wert
MaterialType		
MaterialList		
InitNbrOfObjects		0
Clearance	[mm]	0.0
TEMPLATE		Rail-S4
HIERARCHY		RAILS

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal vier Sensoren beobachten.



CurvedRail45-S4 – Schienenbogen 45° mit vier Sensoren

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann proportional in seiner Größe skaliert werden. Die eingestellte Größe entspricht der Größe des Schienenstücks. Durch die proportionale Skalierung wird der Radius des Bogens verändert, der Winkel von 45° ändert sich nicht.

Funktion

Der Komponententyp *CurvedRail45-S4* dient zur Simulation eines Schienenbogens mit 45° Krümmung. Die Geschwindigkeit, mit der Fahrzeuge über dieses Schienenstück bewegt werden, wird am nicht sichtbaren Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

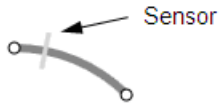
Es können zwischen einem und vier Sensoren über die Länge der Schiene angeordnet werden. Wenn sich ein Fahrzeug im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID der Fahrzeuge kann über die nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über einen Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NbrOfSensors*
  Anzahl der verwendeten Sensoren (1 bis 4)
- *SensorPosition*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.

Jeder definierte Sensor wird bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.



Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NbrOfSensors		1
▾ SensorPosition [1]		...
SensorPosition1	[mm]	0.0

Während der Initialisierung der Komponente werden die Parameter auf ihre Gültigkeit geprüft. Folgende Meldungen weisen auf mögliche Fehlparametrierungen hin:

- *Parameter 'SensorPosition' must be less than the component's length*
  Die Sensorposition darf nicht außerhalb der Komponente liegen.
- *Parameter 'SensorPosition' must not be negative*
  Die Sensorposition darf nicht negativ sein.

Zusatzparameter

Die Komponente *CurvedRail45-S4* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	CurvedRail90-S4
HIERARCHY	RAILS

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal vier Sensoren beobachten.



CurvedRail90-S4 – Schienenbogen 90° mit vier Sensoren

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann proportional in seiner Größe skaliert werden. Die eingestellte Größe entspricht der Größe des Schienenstücks. Durch die proportionale Skalierung wird der Radius des Bogens verändert, der Winkel von 90° ändert sich nicht.

Funktion

Der Komponententyp *CurvedRail90-S4* dient zur Simulation eines Schienenbogens mit 90° Krümmung. Die Geschwindigkeit, mit der Fahrzeuge über dieses Schienenstück bewegt werden, wird am nicht sichtbaren Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Es können zwischen einem und vier Sensoren über die Länge der Schiene angeordnet werden. Wenn sich ein Fahrzeug im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige

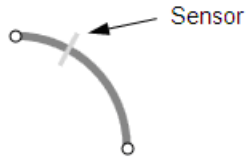
ID der Fahrzeuge kann über die nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *MaxSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NbrOfSensors*
  Anzahl der verwendeten Sensoren (1 bis 4)
- *SensorPosition*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.

Jeder definierte Sensor wird bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.



Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NbrOfSensors		1
SensorPosition [1]		...
SensorPosition1	[mm]	0.0

Während der Initialisierung der Komponente werden die Parameter auf ihre Gültigkeit geprüft. Folgende Meldungen weisen auf mögliche Fehlparametrierungen hin:

- *Parameter 'SensorPosition' must be less than the component's length*
  Die Sensorposition darf nicht außerhalb der Komponente liegen.
- *Parameter 'SensorPosition' must not be negative*
  Die Sensorposition darf nicht negativ sein.

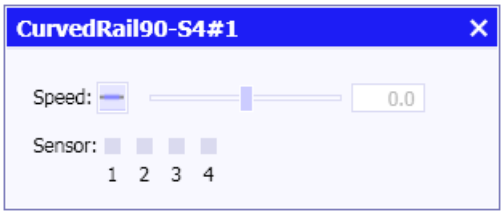
Zusatzparameter

Die Komponente *CurvedRail90-S4* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	CurvedRail90-S4
HIERARCHY	RAILS

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal vier Sensoren beobachten.



RailSwitch-F – Umschaltbare Weiche mit 45°-Abzweig

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann zur Einstellung der Größe der Weiche in seiner Größe proportional skaliert werden. Durch die proportionale Skalierung werden die Länge der Weiche und der Radius des Abzweigbogens verändert, der Winkel des Abzweigs von 45° ändert sich nicht.

Funktion

Der Komponententyp *RailSwitch-F* dient zur Simulation einer Verzweigung (Weiche) bzw. Zusammenführung mit einem 45°-Abzweig. Beim Transport in Bezugsrichtung liegt die Funktion einer Weiche vor, bei Transport entgegen der Bezugsrichtung die Funktion einer Zusammenführung.

Wird der nicht sichtbare binäre Eingang *Switch* auf eins (True) gesetzt, dann wird vom Transport über die gerade Strecke (*A* – *B*) auf den Transport über den Abzweig (*A* – *C*) umgeschaltet. Beachten Sie, dass auch der Transport entgegen der Bezugsrichtung nur über eine aktivierte Strecke möglich ist.

Die Geschwindigkeit, mit der Fahrzeuge über die Schienen bewegt werden, wird am nicht sichtbaren Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Die Weiche kann nicht umgeschaltet werden, wenn sie mit Fahrzeugen belegt ist. Bei Umschaltungen einer belegten Weiche wird in der Meldezeile die folgende Fehlermeldung ausgegeben: "Switch cannot be operated while being occupied".

Parameter

Das Verhalten der Komponente ist über einen Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar

Der Parameter mit seiner Einheit und Vorbelegung ist in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

Die Komponente *RailSwitch-F* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	RailSwitch-F
HIERARCHY	RAILS

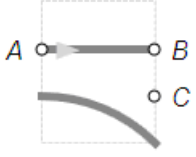
Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und die Strecken umschalten.



RailSwitch-M – Verfahrbare Weiche mit 45°-Abzweig

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann zur Einstellung der Größe der Weiche in seiner Größe proportional skaliert werden. Durch die proportionale Skalierung werden die Länge der Weiche und der Radius des Abzweigbogens verändert, der Winkel des Abzweigs von 45° ändert sich nicht.

Funktion

Der Komponententyp *RailSwitch-M* dient zur Simulation einer Verzweigung (Weiche) bzw. Zusammenführung mit einem 45°-Abzweig. Beim Transport in Bezugsrichtung liegt die Funktion einer Weiche vor, bei Transport entgegen der Bezugsrichtung die Funktion einer Zusammenführung.

Über den nicht sichtbaren analogen Eingang *Switch* wird die Weiche vom Transport über die geraden Strecke (*A* – *B*) auf den Transport über den Abzweig (*A* – *C*) verfahren. Die Verfahrzeit der Weiche ist parametrierbar (Parameter *SwitchingTime*). Mit positiven Werten am Eingang Switch wird die Weiche auf die Strecke *A* – *C* verfahren, mit negativen Werten auf die Strecke *A* – *B*, wobei die Absolutwerte prozentuale Werte der parametrierten Umschaltgeschwindigkeit sind. Der Verfahrvorgang wird für eine Komponente im Symbol animiert dargestellt.

Die Geschwindigkeit, mit der Fahrzeuge über die Schienen bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Die Weiche kann nicht verfahren werden, wenn sie mit Fahrzeugen belegt ist. Beim Verfahren einer belegten Weiche wird in der Meldezeile die folgende Fehlermeldung ausgegeben: "Switch cannot be operated while being occupied".

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *SwitchingTime*
  Umschaltzeit der Weiche bei Ansteuerung mit ±100 %-Werten am Eingang *Switch*.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
SwitchingTime	[s]	1.0

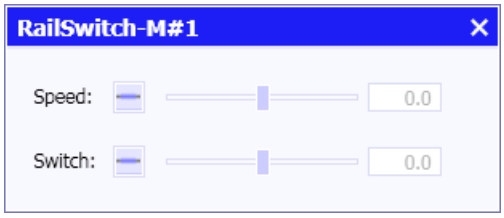
Zusatzparameter

Die Komponente *RailSwitch-M* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	RailSwitch-M
HIERARCHY	RAILS

Bedienfenster

Im Bedienfenster können Sie über einen Schieber jeweils die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) sowie die Umschaltzeit als prozentualen Wert der parametrierten Umschaltzeit (*SwitchingTime*) vorgeben.



InOut – Ein- und Ausschleusstrecke von Fahrzeugen

Symbol



Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge der Ein-/Ausschleusstrecke.

Funktion

Der Komponententyp *InOut* bildet eine Strecke nach, über die Fahrzeuge in ein Streckennetz ein- oder aus einem Streckennetz ausgeschleust werden. Die Ein-/Ausschleuskomponente wird dazu am Anschluss *A* mit einer offenen Strecke des Netzes verbunden und ergänzt so das Streckennetz.

Beim Einschleusen werden einzelne Fahrzeuge durch Setzen des unsichtbaren Binäreingangs *CreateObject* am nicht angeschlossenen Ende der Ein-/Ausschleusstrecke (linker Rand des Symbols) aufgesetzt und von dort zum Anschluss A transportiert. Entsprechend werden auszuschleusende Fahrzeuge vom Anschluss A zum nicht angeschlossenen Ende der Ein-/Ausschleusstrecke transportiert und dort aus dem Streckennetz entfernt. Die Geschwindigkeit, mit der Fahrzeuge auf der Ein-/Ausschleusstrecke bewegt werden, wird am nicht sichtbaren Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Die Ein-/Ausschleusstrecke kann immer nur mit einem Fahrzeug belegt sein. Solange die Strecke durch ein Fahrzeug belegt ist, können keine weiteren Fahrzeuge zum Einschleusen aufgesetzt oder zum Ausschleusen übernommen werden.

Parameter

Der Typ der einzuschleusenden Fahrzeuge, die Materialliste, aus der sie entnommen werden und die Geschwindigkeit, mit der die Fahrzeuge über die Ein-/Ausschleusstrecke bewegt werden, sind über Parameter einstellbar.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
MaterialType		
MaterialList		

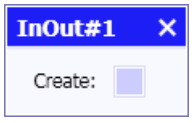
Zusatzparameter

Die Komponente *InOut* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	InOut
HIERARCHY	RAILS

Bedienfenster

Im Bedienfenster können Sie das Aufsetzen eines neuen Fahrzeugs auf die Ein-/ Ausschleusstrecke durch Betätigen des Tasters veranlassen.



RailLifter – Heber

Ein Heber ist eine Schiene, die vertikal in mehrere Ebenen verfahrbar ist. In SIMIT wird ein Heber aus der Basiskomponente vom Typ *RailLifterBase* und bis zu acht Erweiterungskomponenten vom Typ *RailLifterExtension* zusammengesetzt. Die Basiskomponente bildet den Heber in der Basisebene nach, jede weitere Ebene wird durch eine Erweiterungskomponente nachgebildet. Es kann somit ein Heber simuliert werden, der in bis zu acht zusätzliche Ebenen verfahrbar ist.

Die Basiskomponente und die zugehörigen Erweiterungskomponenten müssen Sie dabei nicht auf demselben Diagramm anordnen, Sie können beispielsweise für jede Ebene ein eigenes Diagramm anlegen und die Heberkomponenten so auf die einzelnen Diagramme verteilen.

Für die Parametrierung der Basiskomponente gilt folgendes:

- Der Name der Basiskomponente ist frei wählbar.
- Der Parameter *NbrOfExtensions* gibt die Anzahl der zusätzlichen Ebenen und damit auch die Anzahl der eingesetzten Erweiterungskomponenten an.
- Für jede Ebene ist in der Basiskomponente die Position (*LevelPosition*), d. h. das Niveau über der Basisebene in Millimetern anzugeben. Die Basisebene befindet sich per Definition auf dem Niveau null. Alle Niveauwerte müssen positiv sein und mit der Ebenennummer zunehmen.

Bei der Parametrierung der Erweiterungskomponenten ist folgendes zu beachten:

- Der Name der Erweiterungskomponente setzt sich zusammen aus dem Namen der Basiskomponente, dem Zeichen '#' und einer Zahl, die der Nummer der Ebene entspricht.
- Der Parameter *Level* muss der Nummer der Ebene entsprechen. Die Nummerierung der Ebenen beginnt mit eins.
- Der Parameter *BaseName* ist der Name der Basiskomponente.
- Die Länge des Hebers wird durch die Breite der Basiskomponente vorgegeben.

Hinweis

Alle Erweiterungskomponenten müssen exakt die gleiche Breite haben wie die Basiskomponente. Andernfalls wird Ihnen beim Starten der Simulation diese Unstimmigkeit mit einer Meldung angezeigt.

Der Schienenantrieb und Sensorik des Hebers sind vollständig in der Basiskomponente implementiert. Erweiterungskomponenten sind daher nicht ohne die Basiskomponente funktionstüchtig.

RailLifterBase – Heber (Basiskomponente)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge der Heberschiene.

Funktion

Der Komponententyp *RailLifterBase* dient zur Simulation der Basisstation eines Hebers. Die Geschwindigkeit, mit der Fahrzeuge über die Heberschiene bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Analog wird die Hubgeschwindigkeit, d. h. die Geschwindigkeit mit der die Schiene in die verschiedenen Ebenen verfahren wird, am nicht sichtbaren analogen Eingang *LifterSpeed* als prozentualer Wert der parametrierbaren Nennhubgeschwindigkeit (Parameter *NominalLifterSpeed*) vorgegeben.

Zum Ein- und Ausfahren von Fahrzeugen in den verschiedenen Ebenen des Hebers muss die Heberschiene in die jeweilige Ebene verfahren werden und dort bündig stehen. Bündigkeit muss dabei mit einer für alle Ebenen gleichen Positioniergenauigkeit Δ (Parameter *PositioningAccuracy*) gegeben sein. Die Schiene steht dabei bündig in einer Ebene vom Niveau *H*, wenn für ihr Niveau h gilt:

$H - \Delta \leq h \leq H + \Delta$

Auf der Schiene können bezogen auf jeden der beiden Anschlüsse *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fahrzeug im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Fahrzeugs kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Hinweis

Die Basiskomponente besitzt noch weitere nicht sichtbare Eingänge zum Signalaustausch mit Erweiterungskomponenten. Diese Eingänge sind mit den entsprechenden Signalen vorbelegt. Ändern Sie diese Voreinstellungen nicht, da sich dann die Heberkomponenten nicht mehr wie vorgesehen in der Simulation verhalten.

Parameter

Das Verhalten der Komponente ist parametrierbar.

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NominalLifterSpeed*
  Nennhubgeschwindigkeit; online änderbar
- *NbrOfExtensions*
  Anzahl der zusätzlichen Ebenen (1 bis 8)
- *LevelPosition*
  Niveau der jeweiligen zusätzlichen Ebene
- *PositioningAccuracy*
  Positioniergenauigkeit, innerhalb der die Ebene als angefahren gilt.
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss A bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss B bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols

Die definierten Sensoren werden bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NominalLifterSpeed	[m/s]	1.5
NbrOfExtensions		1
▼ LevelPosition [1]		...
LevelPosition1	[mm]	0.0
PositioningAccuracy	[mm]	100.0
NbrOfSensorsA		1
▼ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▼ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente *RailLifterBase* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

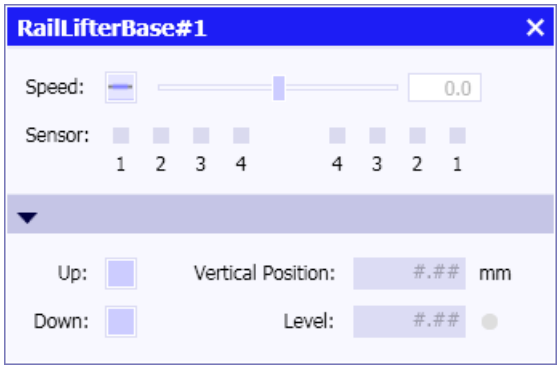
Zusätzlich muss angegeben werden, wie viele Fahrzeuge sich gleichzeitig auf dieser Förderstrecke befinden können.

Name	Wert
MaxObjects	8
TEMPLATE	RailLifterBase
HIERARCHY	RAILS

Bedienfenster

Im Bedienfenster können Sie über einen Schieber die Transportgeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) vorgeben und Sie können den Status der Sensoren beobachten.

Im erweiterten Bedienfenster können Sie den Heber über Taster nach oben oder unten fahren. Die aktuelle Heberposition und Ebene werden angezeigt.



RailLifterExtension – Heber (Erweiterungskomponente)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fahrzeuge an. Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge der Heberschiene.

Hinweis

Die Breite der Erweiterungskomponente muss gleich der Breite der Basiskomponente sein.

Funktion

Der Komponententyp *RailLifterExtension* dient zur Simulation eines Hebers in einer der erreichbaren Etagen. Eine Komponente dieses Typs ist nur in Kombination mit einer Komponente vom Typ *RailLifterBase* einsetzbar.

Parameter

Das Verhalten der Komponente ist parametrierbar:

- *Level*
  Ebene, in der sich die Komponente befindet. Die Zählung beginnt mit eins.
- *BaseName*
  Name der korrespondierenden Basiskomponente

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name	Wert
Level	1
BaseName	

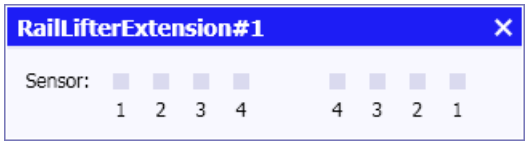
Zusatzparameter

Geben Sie an, wie viele Fahrzeuge sich maximal gleichzeitig auf der durch die Komponente definierten Strecke (Heberschiene) befinden können. Dieser Wert ist zweckmäßig auf den gleichen Wert wie in der Basiskomponente zu setzen.

Name	Wert
MaxObjects	8

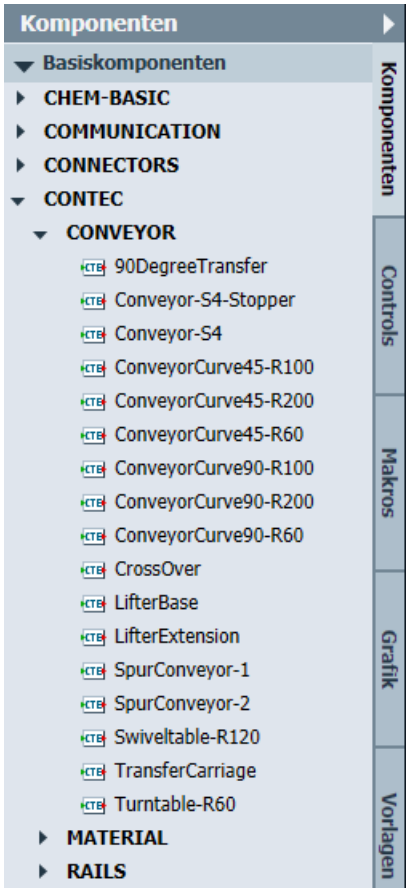
Bedienfenster

Im Bedienfenster können Sie den Status der Sensoren beobachten.



8.3.3.3 Komponententypen für fahrzeuglose Fördersysteme

Im Verzeichnis *CONVEYOR* der Bibliothek *CONTEC* stehen Komponententypen zur Verfügung, die zur Nachbildung von Fördersystemen wie beispielsweise Rollenförder-, Kettenförder und Bandfördersysteme benutzt werden können. Alle diese Komponenten geben die Geschwindigkeit vor, mit der das Fördergut bewegt wird. Das Fördergut verhält sich im Zusammenhang mit diesen Komponenten passiv, besitzt also in der Regel keinen eigenen Antrieb. Typischerweise transportieren diese Komponenten Paletten, Behälter oder Produktionsgüter.



Bei allen Komponententypen der Teilbibliothek *CONVEYOR* ist die Höhe des Symbols mit 40 Pixeln fest vorgegeben. Entsprechend dem eingestellten Maßstab können Sie die Breite der Förderstrecken auf die in folgender Tabelle aufgeführten Werte einstellen.

Tabelle 8-50 Einstellbare Breiten für Förderstrecken

Breite Förderstrecke	Maßstab
40 mm	1 pix : 1 mm
200 mm	1 pix : 5 mm
400 mm	1 pix : 10 mm
800 mm	1 pix : 20 mm
2000 mm	1 pix : 50 mm
4000 mm	1 pix : 100 mm

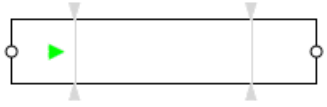
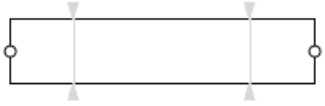
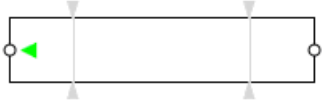
Conveyor-S4 – Geradförderer mit vier Sensoren

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Tabelle 8-51 Anzeige der aktuellen Transportrichtung am Symbol

	Transport in Bezugsrichtung
	Kein Transport
	Transport entgegen Bezugsrichtung

Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge des Förderers.

Funktion

Der Komponententyp *Conveyor-S4* simuliert eine gerade Förderstrecke gegebener Länge. Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Am Förderer können bezogen auf jeden der beiden Anschlüsse *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss *A* bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss *B* bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols

Die definierten Sensoren werden bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NbrOfSensorsA		1
▼ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▼ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente *Conveyor-S4* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Über weitere Zusatzparameter können für die Simulation Fördergüter auf den Förderer aufgelegt werden. Diese werden nach dem Start der Simulation beginnend am Ende *A* des Förderers aneinandergereiht aufgelegt:

- *MaterialType*
  Der Typ des initial aufzulegenden Förderguts
- *MaterialList*
  Der Name der Materialliste, aus der das Fördergut genommen werden soll. Wenn dieser Parameter leer ist, werden alle im Simulationsprojekt angelegten Materiallisten durchsucht.
- *InitNbrOfObjects*
  Die Anzahl der aufzusetzenden Fördergüter
- *Clearance*
  Der Abstand zwischen den aufzusetzenden Fördergütern

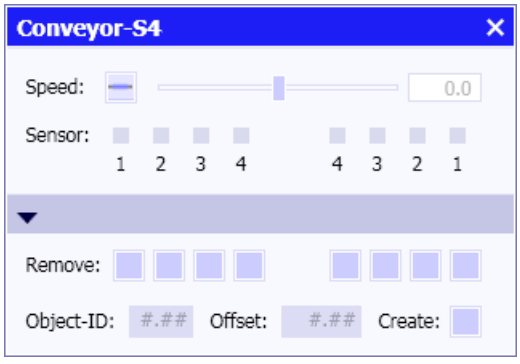
Name		Wert
MaterialType		
MaterialList		
InitNbrOfObjects		0
Clearance	[mm]	0.0
TEMPLATE		Conveyor-S4
HIERARCHY		CONVEYOR

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal acht Sensoren beobachten.

Im erweiterten Bedienfenster ist es möglich, Fördergüter, die aktuell durch eine der acht Sensoren erfasst werden, aus der Förderstrecke zu entfernen (*Remove*). Weiter können Sie mit dem Taster *Create* neue Fördergüter auf die Förderstrecke auflegen (*Create*). Das Fördergut mit der angegebenen Material-ID (*Object-ID*) wird dann mit dem gegebenen Abstand (*Offset)* vom Anschluss A auf dem Förderer aufgelegt. Ist keine Material-ID angegeben, wird ein aufzulegendes Fördergut vom durch den Zusatzparameter *MaterialTyp* vorgegebenen Typ in den Materiallisten (Zusatzparameter *MaterialList*) gesucht.

Damit können manuelle Eingriffe in die Förderanlage wie das Herausnehmen und Auflegen von Fördergütern simuliert werden.



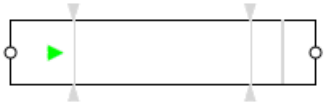
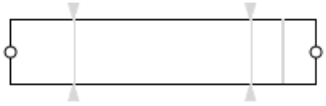
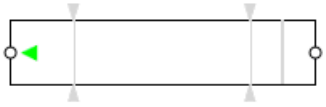
Conveyor-S4-Stopper – Geradförderer mit vier Sensoren und Stopper

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Tabelle 8-52 Anzeige der aktuellen Transportrichtung am Symbol

	Transport in Bezugsrichtung
	Kein Transport
	Transport entgegen Bezugsrichtung

Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge des Förderers.

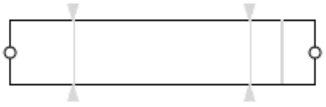
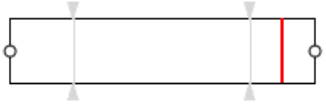
Funktion

Der Komponententyp *Conveyor-S4-Stopper* simuliert eine gerade Förderstrecke gegebener Länge mit einem Stopper in einer auf der Förderstrecke gegebenen Position. Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Am Förderer können bezogen auf jeden der beiden Anschlüsse *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Der Stopper kann aktiviert und deaktiviert werden, um Fördergüter an der Position des Stoppers in beiden Förderrichtungen anzuhalten. Ein aktivierter Stopper wird im Symbol durch einen Farbumschlag nach rot gekennzeichnet.

Tabelle 8-53 Darstellung des Stoppers im Symbol

	Deaktivierter Stopper
	Aktivierter Stopper

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *StopperPosition*
  Position des Stoppers bezogen auf den Anschluss *A*
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss *A* bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss *B* bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols.

Die definierten Sensoren werden bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
StopperPosition	[mm]	0.0
NbrOfSensorsA		1
▼ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▼ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente *Conveyor-S4* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Über weitere Zusatzparameter können für die Simulation Fördergüter auf den Förderer aufgelegt werden. Diese werden nach dem Start der Simulation beginnend am Ende *A* des Förderers aneinandergereiht aufgelegt:

- *MaterialType*
  Der Typ des initial aufzulegenden Förderguts
- *MaterialList*
  Der Name der Materialliste, aus der das Fördergut genommen werden soll. Wenn dieser Parameter leer ist, werden alle im Simulationsprojekt angelegten Materiallisten durchsucht.
- *InitNbrOfObjects*
  Die Anzahl der aufzusetzenden Fördergüter
- *Clearance*
  Der Abstand zwischen den aufzusetzenden Fördergütern:

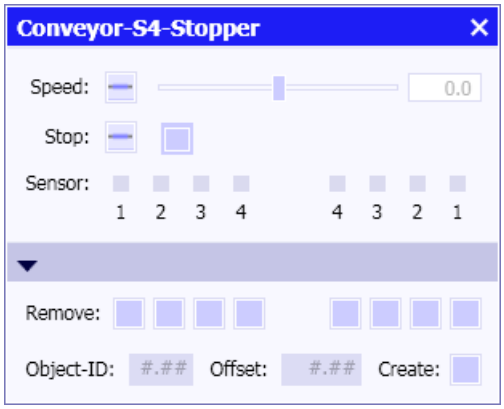
Name	Wert
MaterialType	
MaterialList	
InitNbrOfObjects	0
Clearance	0.0
TEMPLATE	Conveyor-S4-Stopper
HIERARCHY	CONVEYOR

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal acht Sensoren beobachten. Den Stopper können Sie über einen Schalter aktivieren und deaktivieren.

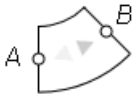
Im erweiterten Bedienfenster ist es möglich, Fördergüter, die aktuell durch eine der acht Sensoren erfasst werden, aus der Förderstrecke zu entfernen (*Remove*). Weiter können Sie mit dem Taster *Create* neue Fördergüter auf die Förderstrecke aufzulegen (*Create*). Das Fördergut mit der angegebenen Material-ID (*Object-ID*) wird dann mit dem gegebenen Abstand (*Offset)* vom Anschluss A auf dem Förderer aufgelegt. Ist keine Material-ID angegeben, wird ein aufzulegendes Fördergut vom durch den Zusatzparameter *MaterialTyp* vorgegebenen Typ in den Materiallisten (Zusatzparameter *MaterialList*) gesucht.

Damit können manuelle Eingriffe in die Förderanlage wie das Herausnehmen und Auflegen von Fördergütern simuliert werden.



ConveyorCurve45-R60 – 45°- Kurvenförderer (Radius 60 Pixel)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *ConveyorCurve45-R60* dient zur Simulation eines Kurvenförderers. Die Förderstrecke bildet einen Kreisbogen von 45°. Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Kreisbogens mit 60 Pixel fest vorgegeben und kann nicht verändert werden. Der für die Komponente wirksame Radius ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm ein wirksamer Radius von 6 Metern.

Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

Die Komponente *ConveyorCurve45-R60* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	ConveyorCurve45-R60
HIERARCHY	CONVEYOR

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben.



ConveyorCurve45-R100 – 45°- Kurvenförderer (Radius 100 Pixel)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *ConveyorCurve45-R100* dient zur Simulation eines Kurvenförderers. Die Förderstrecke bildet einen Kreisbogen von 45°. Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Kreisbogens mit 100 Pixel fest vorgegeben und kann nicht verändert werden. Der für die Komponente wirksame Radius ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm ein wirksamer Radius von 10 Metern.

Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

Die Komponente *ConveyorCurve45-R100* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	ConveyorCurve45-R100
HIERARCHY	CONVEYOR

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben



ConveyorCurve45-R200 – 45°- Kurvenförderer (Radius 200 Pixel)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *ConveyorCurve45-R200* dient zur Simulation eines Kurvenförderers. Die Förderstrecke bildet einen Kreisbogen von 45°. Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Kreisbogens mit 200 Pixel fest vorgegeben und kann nicht verändert werden. Der für die Komponente wirksame Radius ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm ein wirksamer Radius von 20 Metern.

Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

Die Komponente *ConveyorCurve45-R200* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	ConveyorCurve45-R200
HIERARCHY	CONVEYOR

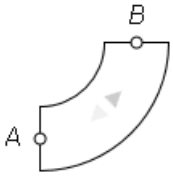
Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben



ConveyorCurve90-R60 – 90°- Kurvenförderer (Radius 60 Pixel)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *ConveyorCurve45-R60* dient zur Simulation eines Kurvenförderers. Die Förderstrecke bildet einen Kreisbogen von 90°. Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Kreisbogens mit 60 Pixel fest vorgegeben und kann nicht verändert werden. Der für die Komponente wirksame Radius ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm ein wirksamer Radius von 6 Metern.

Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

Die Komponente *ConveyorCurve90-R60* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	ConveyorCurve90-R60
HIERARCHY	CONVEYOR

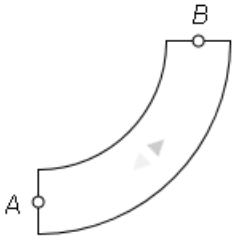
Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben.



ConveyorCurve90-R100 – 90°- Kurvenförderer (Radius 100 Pixel)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *ConveyorCurve90-R100* dient zur Simulation eines Kurvenförderers. Die Förderstrecke bildet einen Kreisbogen von 90°. Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Kreisbogens mit 100 Pixel fest vorgegeben und kann nicht verändert werden. Der für die Komponente wirksame Radius ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm ein wirksamer Radius von 10 Metern.

Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

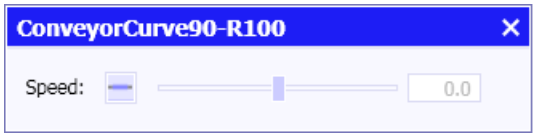
Zusatzparameter

Die Komponente *ConveyorCurve90-R100* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	ConveyorCurve90-R100
HIERARCHY	CONVEYOR

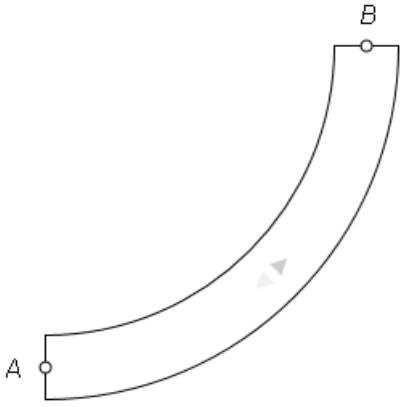
Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben.



ConveyorCurve90-R200 – 90°- Kurvenförderer (Radius 200 Pixel)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *ConveyorCurve90-R200* dient zur Simulation eines Kurvenförderers. Die Förderstrecke bildet einen Kreisbogen von 90°. Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Kreisbogens mit 200 Pixel fest vorgegeben und kann nicht verändert werden. Der für die Komponente wirksame Radius ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm ein wirksamer Radius von 20 Metern.

Die Geschwindigkeit, mit der Fördergüter über diesen Förderer bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

Die Komponente *ConveyorCurve90-R200* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	ConveyorCurve90-R200
HIERARCHY	CONVEYOR

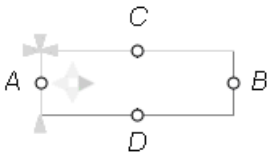
Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben.



90DegreeTransfer – Eckumsetzer

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter auf der Förderstrecke *A* – *B* an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt. Ebenfalls durch grüne Pfeile wird die Richtung des Transfers (nach *D* oder *C*) angezeigt.

Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge des Eckumsetzers. Die Höhe des Symbols ist wie bei allen Komponententypen der Teilbibliothek *CONVEYOR* mit 40 Pixeln fest vorgegeben. Die für den Eckumsetzer wirksame Breite ergibt sich damit aus dem für das Diagramm eingestellten Maßstab.

Die Positionen C und D zum Transfer der Fördergüter können in ihrer Lage verschoben werden. Drücken Sie dazu die ALT-Taste und verschieben Sie den Anschlusspunkt C oder D mit der Maus (gedrückte linke Maustaste) an die gewünschte Position.

Funktion

Der Komponententyp *90DegreeTransfer* dient zur Simulation eines Eckumsetzers, mit dessen Hilfe Fördergüter ohne Drehung von der Förderstrecke *A* – *B* auf einen am Anschluss *C* oder *D* angeschlossenen Förderer umgesetzt werden können.

Die Geschwindigkeit, mit der Fördergüter auf der Förderstrecke *A* – *B* bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Entsprechend wird am nicht sichtbaren analogen Eingang *TransferSpeed* die Geschwindigkeit als prozentualer Wert der parametrierbaren Nenntransfergeschwindigkeit (Parameter *NominalTransferSpeed*) vorgegeben mit der Fördergüter auf der Transferstrecke *C* – *D* umgesetzt werden. Die Bezugsrichtung der Transferstrecken ist vom Anschluss C zur Transportstrecke bzw. von der Transportstrecke zum Anschluss *D* vorgegeben.

Am Umsetzer können bezogen auf jeden der vier Anschlüsse *A*, *B, C* und *D* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NominalTransferSpeed*
  Nenntransfergeschwindigkeit; online änderbar
- *PositioningAccuracy*
  Genauigkeit, mit der das Fördergut zum Transfer auf die an *C* oder *D* anschließende Transferstrecke positioniert sein muss.
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss *A* bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss *B* bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols.
- *NbrOfSensorsC*
  Anzahl der auf den Anschluss *C* bezogenen Sensoren (1 bis 4)
- *SensorPositionC*
  Position des jeweiligen Sensors bezogen auf den Anschluss *C* des Symbols.
- *NbrOfSensorsD*
  Anzahl der auf den Anschluss *D* bezogenen Sensoren (1 bis 4)
- *SensorPositionD*
  Position des jeweiligen Sensors bezogen auf den Anschluss *D* des Symbols.

Die definierten Sensoren werden bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
PositioningAccuracy	[mm]	100.0
NbrOfSensorsA		1
▾ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▾ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0
NbrOfSensorsC		1
▾ SensorPositionC [1]		...
SensorPositionC1	[mm]	0.0
NbrOfSensorsD		1
▾ SensorPositionD [1]		...
SensorPositionD1	[mm]	0.0

Zusatzparameter

Die Komponente *90DegreeTransfer* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

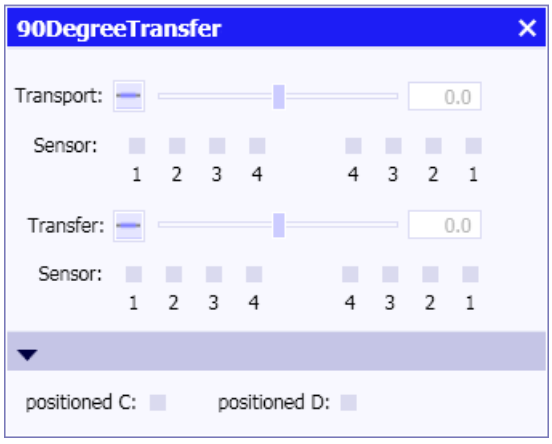
Zusätzlich muss angegeben werden, wie viele Fördergüter sich gleichzeitig auf der Förderstrecke befinden können.

Name	Wert
MaxObjects	8
TEMPLATE	90DegreeTransfer
HIERARCHY	CONVEYOR

Bedienfenster

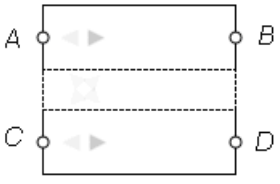
Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) sowie die Transfergeschwindigkeit als prozentualen Wert der Nenntransfergeschwindigkeit (*NominalTransferSpeed*) über einen Schieber vorgeben und den Status der maximal acht Sensoren beobachten.

Im erweiterten Bedienfenster wird angezeigt, ob sich das Fördergut nahe genug am Anschluss C bzw. D befindet, um dort umgesetzt zu werden.



CrossOver – Kreuzweiche

Symbol



Die beiden grauen Pfeilspitzen im Symbol zeigen die Bezugsrichtung für die Bewegung der Fördergüter über die Strecken A – B und C – D an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge der Kreuzweiche.

Funktion

Der Komponententyp *CrossOver* simuliert eine Kreuzweiche. Die Geschwindigkeit, mit der Fördergüter über diese Weiche bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben.

Zum Schalten der Weiche dienen die beiden binären Eingänge *Switch_CB_DA* und *Switch_AD_BC*. Ist der Eingang *Switch_CB_DA* gesetzt (True), dann erfolgt der Transport nicht mehr über die Strecke *A – B*, sondern je nach Transportrichtung über die Strecke *A – D* (Transport in Bezugsrichtung) oder über die Strecke *B – C* (Transport entgegen der Bezugsrichtung). Entsprechend erfolgt der Transport nicht mehr über die Strecke *C – D*, sondern je nach Transportrichtung über die Strecke *C – B* (Transport in Bezugsrichtung) oder über die Strecke *A – D* (Transport entgegen der Bezugsrichtung), wenn der Eingang *Switch_AD_BC* gesetzt ist (True). Gleichzeitiges Setzen beider Eingänge bewirkt ein Abschalten der Weiche, d. h. es findet kein Transport mehr statt.

Parameter

Die Nenngeschwindigkeit, mit der das Fördergut transportiert wird, ist über den Parameter *NominalSpeed* einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

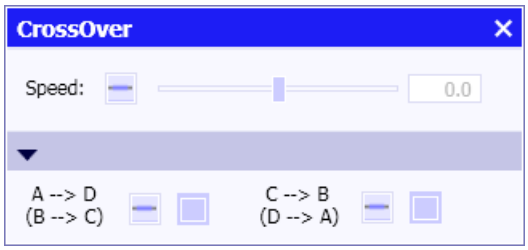
Die Komponente *CrossOver* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	CrossOver
HIERARCHY	CONVEYOR

Bedienfenster

Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal acht Sensoren beobachten.

Im erweiterten Bedienfenster können Sie über Schalter die beiden binären Eingänge *Switch_CB_DA* und *Switch_AD_BC* setzen und so die Transportwege der Kreuzweiche entsprechend manuell vorgeben.



Lifter – Heber

Ein Heber ist eine Förderstrecke, die vertikal in mehrere Ebenen verfahrbar ist. In SIMIT wird ein Heber aus der Basiskomponente vom Typ *LifterBase* und bis zu acht Erweiterungskomponenten vom Typ *LifterExtension* zusammengesetzt. Die Basiskomponente bildet den Heber in der Basisebene nach, jede weitere Ebene wird durch eine Erweiterungskomponente nachgebildet. Es kann somit ein Heber simuliert werden, der in bis zu acht zusätzliche Ebenen verfahrbar ist.

Die Basiskomponente und die zugehörigen Erweiterungskomponenten müssen Sie dabei nicht auf demselben Diagramm anordnen, Sie können beispielsweise für jede Ebene ein eigenes Diagramm anlegen und die Heberkomponenten so auf die einzelnen Diagramme verteilen.

Für die Parametrierung der Basiskomponente gilt folgendes:

- Der Name der Basiskomponente ist frei wählbar.
- Der Parameter *NbrOfExtensions* gibt die Anzahl der zusätzlichen Ebenen und damit auch die Anzahl der eingesetzten Erweiterungskomponenten an.
- Für jede Ebene ist in der Basiskomponente die Position (*LevelPosition*), d. h. das Niveau über der Basisebene, in Millimetern anzugeben. Die Basisebene befindet sich per Definition auf dem Niveau null. Alle Niveauwerte müssen positiv sein und mit der Ebenennummer zunehmen.

Bei der Parametrierung der Erweiterungskomponenten ist folgendes zu beachten:

- Der Name der Erweiterungskomponente setzt sich zusammen aus dem Namen der Basiskomponente, dem Zeichen '#' und einer Zahl, die der Nummer der Ebene entspricht.
- Der Parameter *Level* muss der Nummer der Ebene entsprechen. Die Nummerierung der Ebenen beginnt mit eins.
- Der Parameter *BaseName* ist der Name der Basiskomponente.
- Die Länge des Hebers wird durch die Breite der Basiskomponente vorgegeben.

Hinweis

Alle Erweiterungskomponenten müssen exakt die gleiche Breite haben wie die Basiskomponente. Andernfalls wird Ihnen beim Starten der Simulation diese Unstimmigkeit mit einer Meldung angezeigt.

Die Antriebe und die Sensorik des Hebers sind vollständig in der Basiskomponente implementiert. Erweiterungskomponenten sind daher nicht ohne die Basiskomponente funktionstüchtig.

LifterBase – Heber (Basiskomponente)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge des Hebers.

Funktion

Der Komponententyp *LifterBase* dient zur Simulation der Basisstation eines Hebers. Die Geschwindigkeit, mit der Fördergüter über die Förderstrecke bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Analog wird die Hubgeschwindigkeit, d. h. die Geschwindigkeit mit welcher der Heber in die verschiedenen Ebenen verfahren wird, am nicht sichtbaren analogen Eingang *LifterSpeed* als prozentualer Wert der parametrierbaren Nennhubgeschwindigkeit (Parameter *NominalLifterSpeed*) vorgegeben.

Zum Ein- und Ausfahren von Fördergütern in den verschiedenen Ebenen des Hebers muss der Heber in die jeweilige Ebene verfahren werden und dort bündig stehen. Bündigkeit muss dabei mit einer für alle Ebenen gleichen Positioniergenauigkeit Δ (Parameter *PositioningAccuracy*) gegeben sein. Der Heber steht dabei bündig in einer Ebene vom Niveau *H*, wenn für ihr Niveau h gilt:

$H - \Delta \leq h \leq H + \Delta$

Auf dem Heber können bezogen auf jeden der beiden Anschlüsse *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die

entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Hinweis

Die Basiskomponente besitzt noch weitere nicht sichtbare Eingänge zum Signalaustausch mit Erweiterungskomponenten. Diese Eingänge sind mit den entsprechenden Signalen vorbelegt. Ändern Sie diese Voreinstellungen nicht, da sich sonst die Heberkomponenten nicht mehr wie vorgesehen in der Simulation verhalten.

Parameter

Das Verhalten der Komponente ist parametrierbar.

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NominalLifterSpeed*
  Nennhubgeschwindigkeit; online änderbar
- *NbrOfExtensions*
  Anzahl der zusätzlichen Ebenen (1 bis 8)
- *LevelPosition*
  Niveau der jeweiligen zusätzlichen Ebene
- *PositioningAccuracy*
  Positioniergenauigkeit, innerhalb der die Ebene als angefahren gilt.
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss A bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss B bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols.

Die definierten Sensoren werden bei laufender Simulation positionsrichtig im Symbol dargestellt. Ein aktivierter Sensor wird durch einen Farbumschlag des Sensors nach gelb angezeigt.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NominalLifterSpeed	[m/s]	1.5
NbrOfExtensions		1
▼ LevelPosition [1]		...
LevelPosition1	[mm]	0.0
PositioningAccuracy	[mm]	100.0
NbrOfSensorsA		1
▼ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▼ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente *LifterBase* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

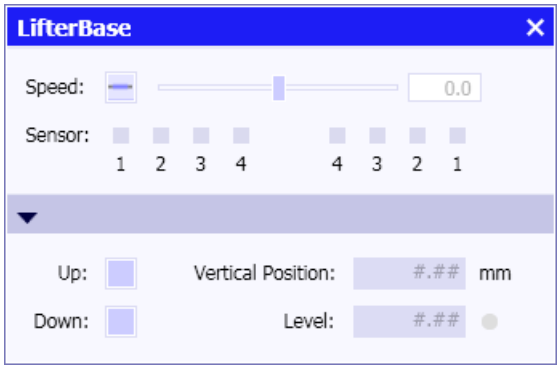
Zusätzlich muss angegeben werden, wie viele Fördergüter sich gleichzeitig auf dieser Förderstrecke befinden können.

Name	Wert
MaxObjects	8
TEMPLATE	LifterBase
HIERARCHY	CONVEYOR

Bedienfenster

Im Bedienfenster können Sie über einen Schieber die Transportgeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) vorgeben und Sie können den Status der Sensoren beobachten.

Im erweiterten Bedienfenster können Sie den Heber über Taster nach oben oder unten fahren. Die aktuelle Heberposition und Ebene werden angezeigt.



LifterExtension – Heber (Erweiterungskomponente)

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für die Bewegung der Fördergüter an. Das Symbol kann in der Breite skaliert werden. Die eingestellte Breite entspricht der Länge der Heberschiene.

Hinweis

Die Breite der Erweiterungskomponente muss gleich der Breite der Basiskomponente sein.

Funktion

Der Komponententyp *LifterExtension* dient zur Simulation eines Hebers in einer der erreichbaren Etagen. Eine Komponente dieses Typs ist nur in Kombination mit einer Komponente vom Typ *LifterBase* einsetzbar.

Parameter

Das Verhalten der Komponente ist parametrierbar:

- *Level*
  Ebene, in der sich die Komponente befindet. Die Zählung beginnt mit eins.
- *BaseName*
  Name der korrespondierenden Basiskomponente

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name	Wert
BaseName	
Level	1

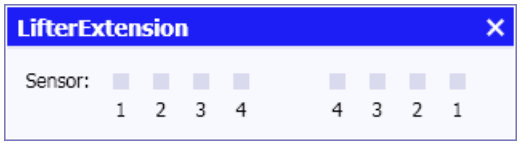
Zusatzparameter

Geben Sie an, wie viele Fördergüter sich maximal gleichzeitig auf der durch die Komponente definierten Strecke befinden können. Dieser Wert ist zweckmäßig auf den gleichen Wert wie in der Basiskomponente zu setzen.

Name	Wert
MaxObjects	8

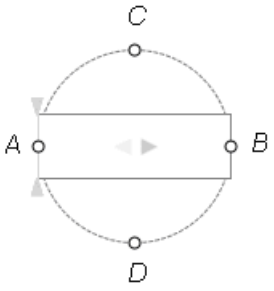
Bedienfenster

Im Bedienfenster können Sie den Status der maximal acht Sensoren beobachten.



Turntable-R60 – Drehtisch

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für den Transport des Förderguts auf dem Drehtisch an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *Turntable-R60* simuliert einen Drehtisch. Mit einem Drehtisch wird eine Förderstrecke um ein Vielfaches von 90° gedreht. Fördergütern können über einen Drehtisch in der ursprünglichen Förderrichtung oder rechtwinklig dazu weiter transportiert werden.

Die Geschwindigkeit, mit der Fördergüter auf dem Drehtisch bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Entsprechend wird die Drehgeschwindigkeit am nicht sichtbaren analogen Eingang *RotationSpeed* als prozentualer Wert der parametrierbaren Nenndrehgeschwindigkeit (Parameter *NominalRotationSpeed*) vorgegeben.

Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Drehtisches mit 60 Pixel fest vorgegeben und kann nicht verändert werden. Entsprechend ist die Länge des Drehtisches mit 120 Pixeln fest vorgegeben. Die für die Komponente wirksame Drehtischlänge ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm eine wirksame Länge von 12 Metern.

Am Drehtisch können bezogen auf jeden der beiden Anschlüsse *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die

entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NominalRotationSpeed*
  Nenndrehgeschwindigkeit; online änderbar
- *PositioningAccuracy*
  Genauigkeit, mit der der Drehtisch positioniert sein muss, damit das Fördergut umgesetzt werden kann.
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss *A* bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss *B* bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NominalRotationSpeed	[°/s]	90.0
PositioningAccuracy	[°]	1.0
NbrOfSensorsA		1
▼ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▼ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente *TurnTable-R60* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

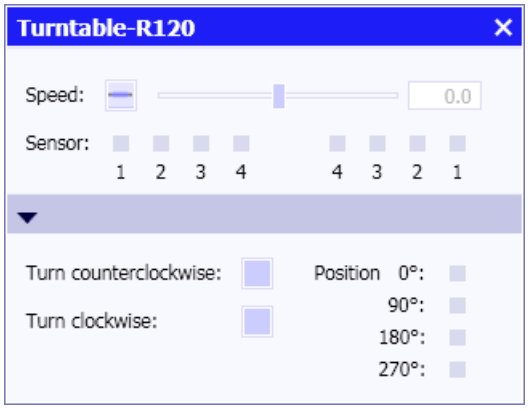
Geben Sie an, wie viele Fördergüter sich gleichzeitig auf dem Drehtisch befinden können.

Name	Wert
MaxObjects	8
TEMPLATE	Turntable-R60
HIERARCHY	CONVEYOR

Bedienfenster

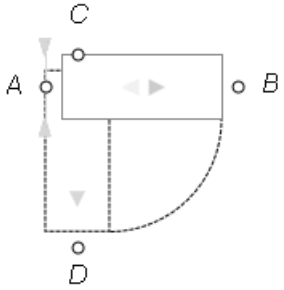
Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal 8 Sensoren beobachten.

Im erweiterten Bedienfenster können Sie den Drehtisch in beide Richtungen manuell über Taster um jeweils 90° drehen. Die jeweilige Endposition wird angezeigt.



Swiveltable-R120 – Schwenktisch

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für den Transport des Förderguts auf dem Schwenktisch an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *Swiveltable-R120* simuliert einen Schwenktisch mit dem eine Förderstrecke um 90° gedreht werden kann. Fördergütern können über einen Schwenktisch in der ursprünglichen Förderrichtung oder rechtwinklig dazu weiter transportiert werden.

Die Geschwindigkeit, mit der Fördergüter auf dem Schwenktisch bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Entsprechend wird die Drehgeschwindigkeit am nicht sichtbaren analogen Eingang

RotationSpeed als prozentualer Wert der parametrierbaren Nenndrehgeschwindigkeit (Parameter *NominalRotationSpeed*) vorgegeben.

Das Symbol der Komponente kann nicht skaliert werden. Somit ist der Radius des Schwenktisches, d. h. die Länge des Schwenktisches mit 120 Pixel fest vorgegeben und kann nicht verändert werden. Die für die Komponente wirksame Schwenktischlänge ergibt sich aus dem für das Diagramm eingestellten Maßstab. Beispielsweise ergibt sich für einen Maßstab von 1 pix : 100 mm eine wirksame Länge von 12 Metern.

Am Schwenktisch können bezogen auf jeden der beiden Anschlüsse *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NominalRotationSpeed*
  Nenndrehgeschwindigkeit; online änderbar
- *PositioningAccuracy*
  Genauigkeit, mit der der Drehtisch positioniert sein muss, damit das Fördergut umgesetzt werden kann.
- *NbrOfSensorsA*
  Anzahl der auf den Anschluss *A* bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf den Anschluss *A* des Symbols.
- *NbrOfSensorsB*
  Anzahl der auf den Anschluss *B* bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf den Anschluss *B* des Symbols.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NominalRotationSpeed	[°/s]	90.0
PositioningAccuracy	[°]	1.0
NbrOfSensorsA		1
▼ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▼ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente Swiveltable-R120 enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

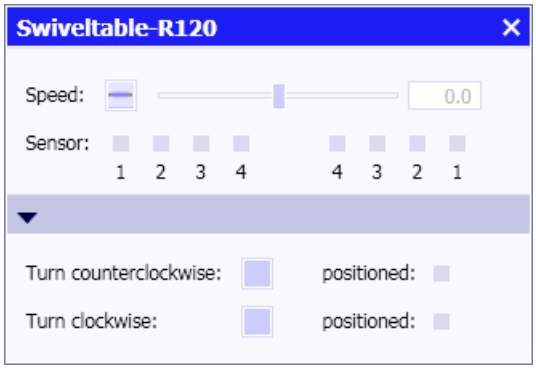
Geben Sie an, wie viele Fördergüter sich gleichzeitig auf dem Schwenktisch befinden können.

Name	Wert
MaxObjects	8
TEMPLATE	Swiveltable-R120
HIERARCHY	CONVEYOR

Bedienfenster

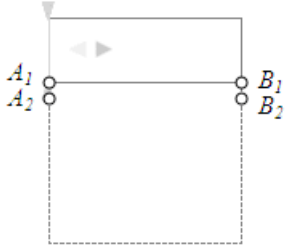
Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal 8 Sensoren beobachten.

Im erweiterten Bedienfenster können Sie den Schwenktisch manuell über einen Taster um 90° drehen und über einen weiteren Taster wieder in die Ausgangslage zurückdrehen. Das Erreichen der jeweiligen Endposition wird mit Indikatoren angezeigt.



TransferCarriage – Querverschiebewagen

Symbol



Die Komponente kann in der Höhe und Breite skaliert werden. Mit der Breite wird die Länge des Verschiebewagens festgelegt, mit der Höhe die Länge des Verschiebeweges.

Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für den Transport des Förderguts auf dem Schwenktisch an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *TransferCarriage* dient zur Simulation eines Querverschiebewagens. Fördergüter können von beiden Seiten auf den Verschiebewagen gefahren werden, dieser kann dann quer zur Transportrichtung an eine andere Position verfahren werden und die Fördergüter können dort über die Anschlüsse A_i und B_i weiter transportiert werden.

Die Anzahl der links und rechts an der Komponente vorhandenen Anschlüsse A_i, B_i kann parametriert werden. Maximal 16 Anschlüsse auf jeder Seite sind möglich; voreingestellt sind zwei Anschlüsse auf jeder Seite: A_1, A_2, B_1, B_2. Jeder Anschluss markiert eine Position, die quer zur Transportrichtung vom Verschiebewagen angefahren werden kann.

Die Geschwindigkeit, mit der Fördergüter auf dem Verschiebewagen transportiert werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Entsprechend wird die Verschiebegeschwindigkeit am nicht sichtbaren analogen Eingang *TransferSpeed* als prozentualer Wert der parametrierbaren Nenndrehgeschwindigkeit (Parameter *NominalTransferSpeed*) vorgegeben.

Am Verschiebewagen können bezogen auf jede der beiden Anschluss-Seiten *A* und *B* jeweils zwischen einem und vier Sensoren angeordnet werden. Wenn sich ein Fördergut im Erfassungsbereich eines Sensors befindet, wird der entsprechende nicht sichtbare Binärausgang *Sensor1* bis *Sensor4* gesetzt. Auf die zugehörige ID des detektierten Förderguts kann über die entsprechenden nicht sichtbaren ganzzahligen Ausgänge *SensorId1* bis *SensorId4* zugegriffen werden.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nennfördergeschwindigkeit; online änderbar
- *NominalTransferSpeed*
  Nennverschiebegeschwindigkeit; online änderbar
- *PositioningAccuracy*
  Genauigkeit, mit der der Verschiebewagen positioniert sein muss, damit das Fördergut umgesetzt werden kann.
- *NbrOfConnectorsA*
  Anzahl der Anschlüsse auf der linken Seite (A)
- *NbrOfConnectorsB*
  Anzahl der Anschlüsse auf der rechten Seite (B)
- *NbrOfSensorsA*
  Anzahl der auf die Anschlussseite *A* bezogenen Sensoren (1 bis 4)
- *SensorPositionA*
  Position des jeweiligen Sensors bezogen auf die Anschlussseite *A* des Symbols.

- *NbrOfSensorsB*
  Anzahl der auf die Anschlussseite *B* bezogenen Sensoren (1 bis 4)
- *SensorPositionB*
  Position des jeweiligen Sensors bezogen auf die Anschlussseite *B* des Symbols.

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
NominalTransferSpeed	[m/s]	1.5
PositioningAccuracy	[mm]	50.0
NbrOfConnectorsA		2
NbrOfConnectorsB		2
NbrOfSensorsA		1
▾ SensorPositionA [1]		...
SensorPositionA1	[mm]	0.0
NbrOfSensorsB		1
▾ SensorPositionB [1]		...
SensorPositionB1	[mm]	0.0

Zusatzparameter

Die Komponente *TransferCarriage* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

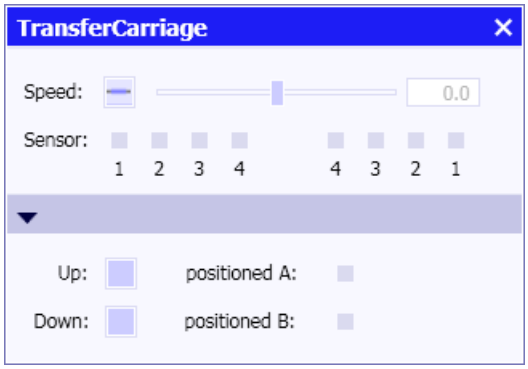
Geben Sie an, wie viele Fördergüter sich gleichzeitig auf dem Verschiebewagen befinden können.

Name	Wert
MaxObjects	8
TEMPLATE	TransferCarriage
HIERARCHY	CONVEYOR

Bedienfenster

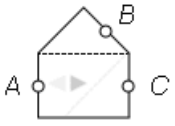
Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben und den Status der maximal 8 Sensoren beobachten.

Im erweiterten Bedienfenster können Sie den Verschiebewagen in beide Richtungen manuell über Taster (*Up*, *Down*) zur jeweils nächsten Position fahren. Die jeweilige Endposition wird angezeigt. Es wird angezeigt, wenn sich der Verschiebewagen an einer Anschlussposition der einen oder anderen Seite befindet.



SpurConveyor-1 – Schrägzulauf mit einem Abzweig

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für den Transport des Förderguts auf dem Schrägzulauf an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *SpurConveyor-1* simuliert einen Schrägzulauf mit einem Abzweig. Fördergüter können über die Strecke *A – C* oder über die Abzweigstrecke *A – B* transportiert werden.

Die Geschwindigkeit, mit der Fördergüter auf dem Schrägzulauf bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Zur Umschaltung der Förderung auf die Abzweigstrecke wird der nicht sichtbare binäre Eingang *Switch* auf eins (True) gesetzt.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

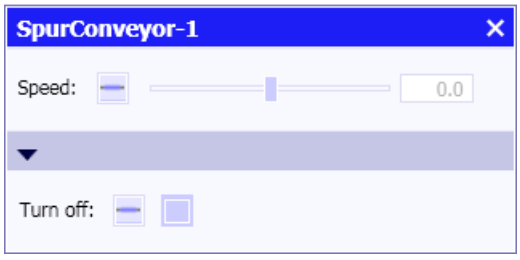
Die Komponente *SpurConveyor-1* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	SpurConveyor-1
HIERARCHY	CONVEYOR

Bedienfenster

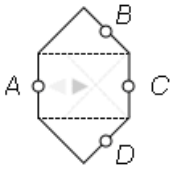
Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben.

Im erweiterten Bedienfenster können Sie mit einem Schalter auf die Abzweigstrecke umschalten.



SpurConveyor-2 – Schrägzulauf mit zwei Abzweigen

Symbol



Die graue Pfeilspitze im Symbol zeigt die Bezugsrichtung für den Transport des Förderguts auf dem Schrägzulauf an. Bei laufender Simulation wird die aktuelle Transportrichtung durch grüne Pfeile im Symbol angezeigt.

Funktion

Der Komponententyp *SpurConveyor-2* simuliert einen Schrägzulauf mit zwei alternativen Abzweigen. Fördergüter können über die Strecke *A – C* oder über eine der beiden Abzweigstrecken *A – B* bzw. *A – D* transportiert werden.

Die Geschwindigkeit, mit der Fördergüter auf dem Schrägzulauf bewegt werden, wird am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der

parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben. Zur Umschaltung der Förderung auf eine der beiden Abzweigstrecke wird einer der beiden nicht sichtbare binäre Eingänge *Switch_AB oder Switch_AD* auf eins (True) gesetzt: *Switch_AB* schaltet die Abzweigstrecke *A – B* und entsprechend *Switch_AD* die Abzweigstrecke *A – D*. Falls einer der beiden Eingänge gesetzt ist, dann ist das Setzen des anderen Eingangs wirkungslos. Werden beide Eingänge gleichzeitig gesetzt, dann wird auf keine der beiden Abzweigstrecken umgeschaltet.

Parameter

Die Nennfördergeschwindigkeit ist über einen Parameter einstellbar. Seine Einheit und Vorbelegung sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0

Zusatzparameter

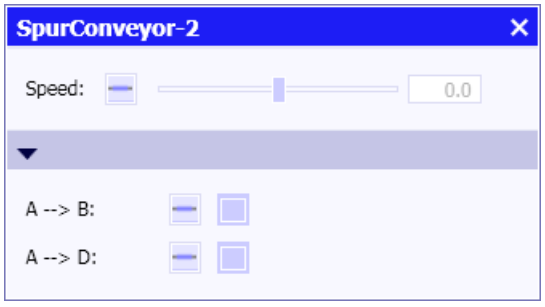
Die Komponente *SpurConveyor-2* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	SpurConveyor-2
HIERARCHY	CONVEYOR

Bedienfenster

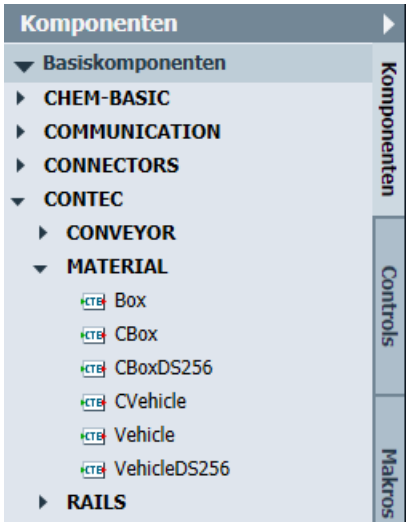
Im Bedienfenster können Sie die Fördergeschwindigkeit als prozentualen Wert der Nennfördergeschwindigkeit (*NominalSpeed*) über einen Schieber vorgeben.

Im erweiterten Bedienfenster können Sie mit einem Schalter auf die entsprechende Abzweigstrecke *A – B* oder *A – D* umschalten.



8.3.3.4 Komponententypen zur Simulation von Fördergütern

Im Verzeichnis *MATERIAL* der Bibliothek *CONTEC* stehen Komponententypen zur Verfügung, mit denen Sie Fördergüter simulieren können. Die Fahrzeuge (*Vehicle*) können Sie sinnvoll mit den Komponententypen im Verzeichnis *RAILS* zur Simulation von Fahrzeugfördersystemen einsetzen. Zur Fördergutsimulation für die Komponententypen im Verzeichnis CONVEYOR verwenden Sie sinnvoll die Kisten (*Box*).



Mit diesen Komponenten legen Sie Materiallisten an. Sie finden weitere Informationen zum Thema auch im Kapitel: Maßstäblichkeit von Fördergütern (Seite 832).

Die Größe einer Komponente (Breite und Höhe des Symbols) wird in der Materialliste in Millimetern definiert.

Box – Einfaches Fördergut

Symbol



Funktion

Der Komponententyp *Box* dient zur Simulation eines Fördergutes mit rechteckigem Grundriss, wie beispielsweise einer Palette oder Kiste. Der Komponententyp definiert die Abmessung des Fördergutes. In der Auflösung 1:20 entspricht die Standard-Abmessung den Maßen einer Europalette (1200 × 800 mm).

CBox – Farbiges Fördergut

Symbol



Funktion

Der Komponententyp C*Box* dient zur Simulation eines Fördergutes mit rechteckigem Grundriss, wie beispielsweise einer Palette oder Kiste. Der Komponententyp definiert die Abmessung des Fördergutes. In der Auflösung 1:20 entspricht die Standard-Abmessung den Maßen einer Europalette (1200 × 800 mm).

Bei laufender Simulation kann das Symbol einer Komponente dieses Typs in 8 unterschiedlichen Farben dargestellt werden. Die Farben entstehen aus den 3 binären Eingängen *R*, *G* und *B* der Komponente als eine additive Farbmischung der 3 Grundfarben rot, grün und blau.

Tabelle 8-54 Farbschema für CBox

Eingang R	Eingang G	Eingang B	Darstellung
False	False	False	
False	False	True	
False	True	False	
False	True	True	
True	False	False	
True	False	True	
True	True	False	
True	True	True	

CBoxDS256 – Farbiges Fördergut mit Datenspeicher

Symbol



Funktion

Der Komponententyp C*BoxDS256* dient zur Simulation eines Fördergutes mit rechteckigem Grundriss, wie beispielsweise einer Palette oder Kiste. Der Komponententyp definiert die Abmessung des Fördergutes. In der Auflösung 1:20 entspricht die Standard-Abmessung den Maßen einer Europalette (1200 × 800 mm).

Bei laufender Simulation kann das Symbol einer Komponente dieses Typs in 8 unterschiedlichen Farben dargestellt werden. Die Farben entstehen aus den 3 binären Eingängen *R*, *G* und *B* der Komponente als eine additive Farbmischung der 3 Grundfarben rot, grün und blau.

Tabelle 8-55 Farbschema für *CBoxDS256*

Eingang R	Eingang G	Eingang B	Darstellung
False	False	False	
False	False	True	
False	True	False	
False	True	True	
True	False	False	
True	False	True	
True	True	False	
True	True	True	

Zusätzlich ist für Komponenten dieses Typs ein Datenspeicher mit einer einstellbaren Größe (Parameter *SizeOfStorage*) definiert. Dieser Speicher bildet den mobilen Datenspeicher eines Förderguts. Er wird als Zustandsvektor *MDS* mit dem Datentyp *byte* angelegt und er kann maximal 256 Byte groß sein. Komponententypen zum Schreiben und Lesen dieses Speichers finden Sie in der Bibliothek *SENSORS* im Verzeichnis *RFID* (Seite 896).

Parameter

Die Größe des Datenspeichers ist über den Parameter *SizeOfStorage* einstellbar. Vorbelegt ist er mit dem Wert eins.

Name	Wert
SizeOfStorage	1

Vehicle – Fahrzeug

Symbol



Funktion

Der Komponententyp *Vehicle* dient zur Simulation von Fahrzeugen wie beispielsweise von Elektrohängebahnen. Angenommen ist ein quadratischer Grundriss des Fahrzeugs. Der Komponententyp definiert die Abmessung des Grundrisses. In der Auflösung 1:20 entspricht die Standard-Abmessung den Maßen 1000 × 300 mm.

Die Geschwindigkeit, mit der sich diese Fahrzeuge in der Simulation im Schienennetz bewegen, wird in der Regel von den einzelnen Schienenstrecken vorgegeben. Die Geschwindigkeit kann aber auch am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben werden. Die beiden Geschwindigkeitsvorgaben für eine Komponente werden addiert.

Eine Komponente dieses Typs kann auch so parametriert werden, dass das Fahrzeug erst verzögert wieder anfährt, wenn es vorher durch Auffahren angehalten worden ist. Diese Funktion ist nur wirksam, wenn die Geschwindigkeit über die Förderstrecke vorgegeben ist.

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nenngeschwindigkeit des Fahrzeugs; online änderbar
- *StartUpDelay*
  Anfahrverzögerung
- *SensorRange*
  Erfassungsbereich der Sensorik

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
StartUpDelay	[s]	0.0
SensorRange	[%]	100.0

Zusatzparameter

Die Komponente *Vehicle* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	Vehicle
HIERARCHY	VEHICLES

CVehicle – Fahrzeug

Symbol



Funktion

Der Komponententyp *CVehicle* dient zur Simulation von Fahrzeugen wie beispielsweise von Elektrohängebahnen. Angenommen ist ein rechteckiger Grundriss des Fahrzeugs. Der Komponententyp definiert die Abmessung des Grundrisses. In der Auflösung 1:20 entspricht die Standard-Abmessung den Maßen 1000 × 300 mm.

Die Geschwindigkeit, mit der sich diese Fahrzeuge in der Simulation im Schienennetz bewegen, wird in der Regel von den einzelnen Schienenstrecken vorgegeben. Die Geschwindigkeit kann aber auch am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben werden. Die beiden Geschwindigkeitsvorgaben für eine Komponente werden addiert.

Eine Komponente dieses Typs kann auch so parametriert werden, dass das Fahrzeug erst verzögert wieder anfährt, wenn es vorher durch Auffahren angehalten worden ist. Diese Funktion ist nur wirksam, wenn die Geschwindigkeit über die Förderstrecke vorgegeben ist.

Bei laufender Simulation kann das Symbol einer Komponente dieses Typs in 8 unterschiedlichen Farben dargestellt werden. Die Farbe ergibt sich aus den 3 binären Eingängen *R*, *G* und *B* der Komponente aus einer additiven Farbmischung der 3 Grundfarben rot, grün und blau.

Tabelle 8-56 Farbschema für *CVehicle*

Eingang R	Eingang G	Eingang B	Darstellung
False	False	False	
False	False	True	
False	True	False	
False	True	True	
True	False	False	
True	False	True	
True	True	False	
True	True	True	

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nenngeschwindigkeit; online änderbar
- *StartUpDelay*
  Anfahrverzögerung
- *SensorRange*
  Erfassungsbereich für Sensorik

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
StartUpDelay	[s]	0.0
SensorRange	[%]	100.0

Zusatzparameter

Die Komponente *CVehicle* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	CVehicle
HIERARCHY	VEHICLES

VehicleDS256 – Fahrzeug mit Datenspeicher

Symbol



Funktion

Der Komponententyp *VehicleDS256* dient zur Simulation von Fahrzeugen wie beispielsweise von Elektrohängebahnen. Angenommen ist ein rechteckiger Grundriss des Fahrzeugs. Der Komponententyp definiert die Abmessung des Grundrisses. In der Auflösung 1:20 entspricht die Standard-Abmessung den Maßen 1000 × 300 mm.

Die Geschwindigkeit, mit der sich diese Fahrzeuge in der Simulation im Schienennetz bewegen, wird in der Regel von den einzelnen Schienenstrecken vorgegeben. Die Geschwindigkeit kann aber auch am nicht sichtbaren analogen Eingang *Speed* der Komponente als prozentualer Wert der parametrierbaren Nennfördergeschwindigkeit (Parameter *NominalSpeed*) vorgegeben werden. Die beiden Geschwindigkeitsvorgaben für eine Komponente werden addiert.

Eine Komponente dieses Typs kann auch so parametriert werden, dass das Fahrzeug erst verzögert wieder anfährt, wenn es vorher durch Auffahren angehalten worden ist. Diese Funktion ist nur wirksam, wenn die Geschwindigkeit über die Förderstrecke vorgegeben ist.

Zusätzlich ist für Komponenten dieses Typs ein Datenspeicher mit einer einstellbaren Größe (Parameter *SizeOfStorage*) definiert. Dieser Speicher bildet den mobilen Datenspeicher eines Fahrzeugs. Er wird als Zustandsvektor *MDS* mit dem Datentyp *Byte* angelegt und er kann maximal 256 Byte groß sein. Komponententypen zum Schreiben und Lesen dieses Speichers finden Sie in der Bibliothek *SENSORS* im Verzeichnis *RFID* (Seite 896).

Parameter

Das Verhalten der Komponente ist über Parameter einstellbar:

- *NominalSpeed*
  Nenngeschwindigkeit; online änderbar
- *StartUpDelay*
  Anfahrverzögerung
- *SensorRange*
  Erfassungsbereich für Sensorik
- *SizeOfStorage*
  Größe des mobilen Datenspeichers

Die Parameter mit ihren Einheiten und Vorbelegungen sind in folgender Abbildung zu sehen.

Name		Wert
NominalSpeed	[m/s]	2.0
StartUpDelay	[s]	0.0
SensorRange	[%]	100.0
SizeOfStorage		1

Zusatzparameter

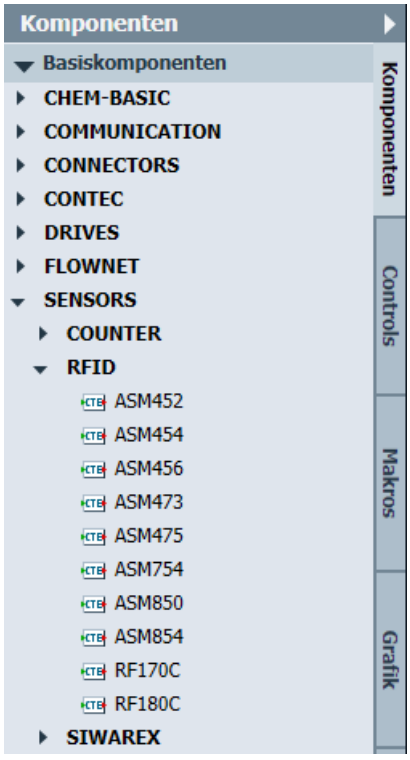
Die Komponente *VehicleDS256* enthält die Zusatzparameter *TEMPLATE und HIERARCHY*, über die die Generierung der Antriebsebene gesteuert werden kann. Weitere Informationen hierzu finden Sie im Kapitel: Verwendung von Vorlagen (Seite 834).

Name	Wert
TEMPLATE	VehicleDS256
HIERARCHY	VEHICLES

8.3.3.5 Komponenten zur Simulation von Identifikationssystemen

Im Verzeichnis *RFID* der Bibliothek *SENSORS* stehen Komponententypen zur Verfügung, mit denen Identifikationssysteme simuliert werden können, die von der Steuerung über den FC/FB45 im Normalbetrieb angesteuert werden. Andere Betriebsmodi werden nicht simuliert. Mit Komponenten dieser Typen ist es möglich, Daten zwischen der Steuerung und den entsprechenden Simulationskomponenten von Fördergütern auszutauschen. Weitere Informationen hierzu finden Sie im Kapitel: Komponententypen zur Simulation von Fördergütern (Seite 889).

Es können sowohl feste Daten wie beispielsweise ein Barcode als auch Schreib-/Lesedaten im Falle eines Moby ausgetauscht werden.



Alle Komponententypen im Verzeichnis *RFID* sind strukturell gleich aufgebaut, unterscheiden sich aber in der Anzahl der zur Verfügung gestellten Kanäle und in den unterstützen Befehlen.

Da die Kommunikation zwischen Anschaltmodulen (ASM) und der Steuerung über zyklische E/A-Daten und azyklische Datensätze erfolgt, können die Komponententypen im Verzeichnis *RFID* nur mit Kopplungen eingesetzt werden, die beide Kommunikationsarten unterstützen. Folglich können diese Komponenten nur mit der PROFIBUS DP- und der PROFINET IO-Kopplung eingesetzt werden. In der ersten Tabelle sind die für PROFIBUS DP unterstützten Anschaltmodule aufgeführt, in der zweiten Tabelle die für PROFINET IO unterstützten.

Tabelle 8-57 Anschaltmodule für PROFIBUS DP

Anschaltmodul	MLFB	Bei modularen Slaves unterstützte Kopfstation
ASM 452	6GT2002-0EB20	
ASM 454	6GT2002-2EE00	

Anschaltmodul	MLFB	Bei modularen Slaves unterstützte Kopfstation
ASM 456	6GT2002-0ED00	
ASM 473	6GT2002-0HA00 6GT2002-0HA10	ET200X 6ES7 141-1BF00-0AB0 (80D2) ET200X 6ES7 141-1BF40-0AB0 (80D3) ET200X 6ES7 141-1BF11-0XB0 (803D) ET200X 6ES7 141-1BF12-0XB0 (803D) ET200X 6ES7 142-1BD21-0XB0 (803C) ET200X 6ES7 142-1BD22-0XB0 (803C) ET200X 6ES7 143-1BF00-0AB0 (809A) ET200X 6ES7 143-1BF00-0XB0 (809A)
ASM 475	6GT2002-0GA10	ET200M 6ES7 153-2BA00-0XB0 (801E) ET200M 6ES7 153-2BB00-0XB0 (8071)
ASM 754	6GT2302-2EE00	
ASM 850	6GT2402-2EA00	
ASM 854	6GT2402-2BB00	
RF170C	6GT2002-0HD00	ET200pro 6ES7 154-1AA00-0AB0 (8118) ET200pro 6ES7 154-1AA01-0AB0 (8118) ET200pro 6ES7 154-2AA00-0AB0 (8119) ET200pro 6ES7 154-2AA01-0AB0 (8119)

Tabelle 8-58 Anschaltmodule für PROFINET IO

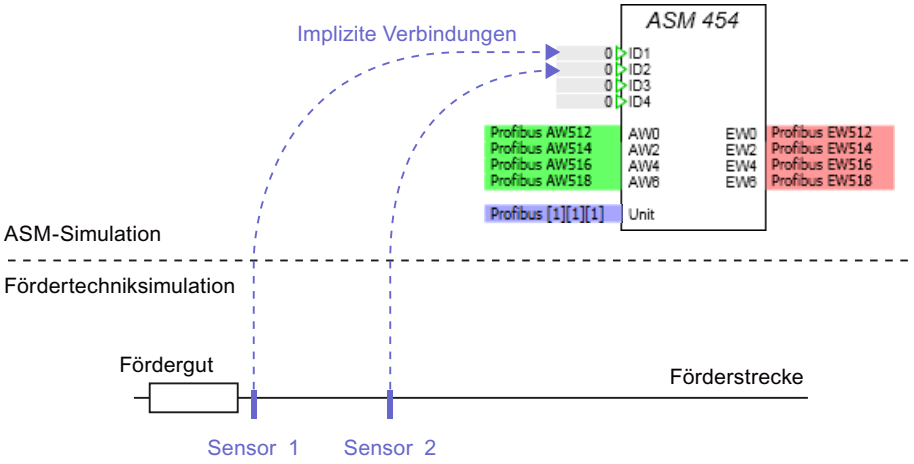
Anschaltmodul	MLFB	Bei modularen Slaves unterstützte Kopfstation
ASM 475	6GT2002-0GA10	ET200M 6ES7 153-4BA00-0XB0 ET200M 6ES7 153-4AA00-0XB0
RF170C	6GT2002-0HD00	ET200pro 6ES7 154-4AB10-0AB0 ET200pro 6ES7 154-6AB00-0AB0 ET200pro 6ES7 154-6AB50-0AB0

Funktionsprinzip

Simulierte Fördergüter werden in SIMIT als Fördergutkomponenten in Materiallisten verwaltet. Jeder Fördergutkomponente wird automatisch mit dem Start der Simulation eine eindeutige Kennnummer (ID) zugeordnet, die über die simulierten Sensoren einer Förderstreckenkomponente zugänglich ist. Zusätzlich kann in Fördergutkomponenten der Inhalt eines mobilen Datenspeichers (MDS) abgebildet sein. Die beiden Komponententypen *CBoxDS256* und *VehicleDS256* im Verzeichnis *MATERIAL* der Bibliothek *CONTEC* sind entsprechend mit einem Datenspeicher angelegt. Um die Inhalte eines simulierten Datenspeichers mit der Steuerung austauschen zu können, werden Komponenten in der Simulation benötigt, die funktional das entsprechende Anschaltmodul nachbilden.

In der realen Anlage ist ein Anschaltmodul mit einem Schreib/Lesegerät (SLG) verbunden, das an einer bestimmten Stelle in der Förderanlage positioniert ist. Dort vorbei transportiertes Fördergut wird vom SLG detektiert und das SLG kann gesteuert den Datenspeicher des Förderguts lesen und gegebenenfalls auch beschreiben.

In SIMIT werden die konkreten an das ASM angeschlossenen Schreib/Lese-Geräte (SLG) und deren Verhalten nicht nachgebildet, es wird lediglich ihre Funktionalität in Bezug auf das Umsetzen der Daten der mobilen Datenspeicher in den ASM-Komponententypen nachgebildet. Wie in folgender Abbildung schematisch dargestellt, werden dazu die Kennnummern (ID) der Fördergüter an die ASM-Komponenten übertragen, die von den simulierten Sensoren an der Förderstrecke geliefert werden. Die entsprechenden Eingänge der ASM-Komponenten sind dazu implizit mit den simulierten Sensoren zu verknüpfen.



Anhand dieser ID wird der Steuerung über das zyklische Eingangssignal mitgeteilt, dass ein Objekt detektiert worden ist. Entsprechend wird, wenn die ASM-Komponente einen azyklischen Befehl empfängt, dieser mit den Daten des MDS des detektierten Förderguts ausgeführt. Der Zugriff auf den Datenspeicher erfolgt dabei einfach über die ID.

Alle hier in der Bibliothek *CONTEC* enthaltenen RFID-Komponententypen enthalten eine Nachbildung der "SIMATIC Sensors – RFID-Systeme" für die Kommunikation mit der STEP 7-Funktion FB45 im Normalbetrieb. Andere Betriebsarten der RFID/Moby-Systeme sind nicht nachgebildet und werden folglich in der Simulation nicht unterstützt.

Die RFID-Komponenten unterstützen die in der folgenden Abbildung aufgeführten Befehle des FB45 Kommandos.

Tabelle 8-59 Unterstützte Befehle des FB45

Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhandenen MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS Status aus einem vorhandenen MDS

Hinweis

Nicht jeder RFID-Komponententyp unterstützt alle in der Tabelle aufgeführten Befehle. Eine Beschreibung der von einem RFID-Komponententyp unterstützten Befehle finden Sie in der Beschreibung des jeweiligen Komponententyps.

Ein mobiler Datenspeicher (MDS) wird in SIMIT in Form eines Zustands-Arrays vom Typ *Byte* nachgebildet, das innerhalb der einzelnen Förderobjekte angelegt ist. Der Zugriff auf diese Daten erfolgt dann anhand der an die ASM-Komponente gemeldeten ID des detektierten Förderobjektes und des in der ASM-Komponente parametrierten Parameters "*MDS*". Dieser Parameter gibt den Namen des Zustandsarrays an, auf das zugegriffen werden soll. Damit ist es möglich, mehrere und auch unterschiedliche MDS für ein Förderobjekt zu definieren, es muss in der ASM-Komponente nur hinterlegt werden, welcher MDS jeweils gelesen bzw. beschrieben werden soll. Für die hier in der CONTEC-Bibliothek zur Verfügung gestellten Komponententypen von Fördergütern mit MDS (Komponententypen *CBoxDS256*, *VehicleDS256*) ist allerding nur ein MDS modelliert. Da es den Namen "*MDS*" hat, ist der Zugriff durch die passende Vorbelegung der Parameter von ASM-Komponenten sichergestellt.

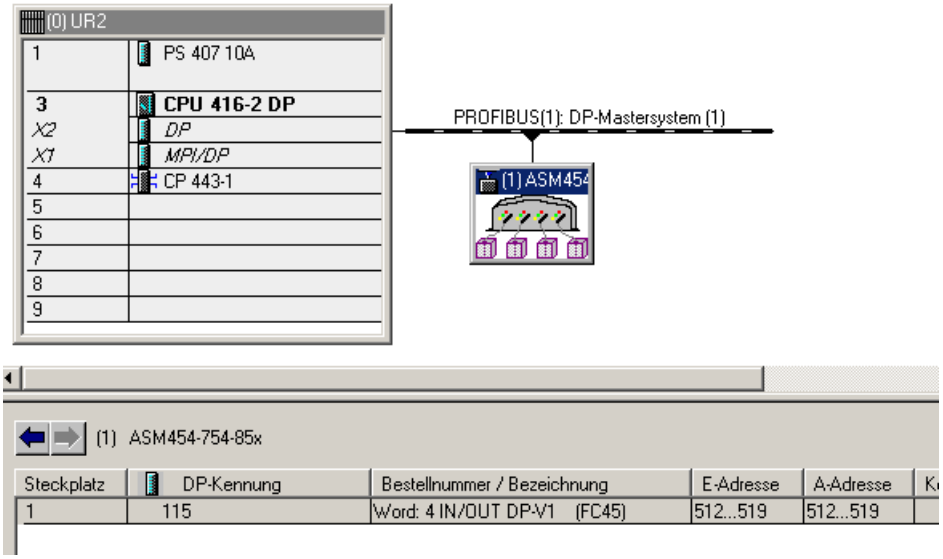
Ein fehlender MDS, fehlende Zustandsarrays oder Bereichsüberschreitungen werden mit den entsprechenden Fehlermeldungen von einer *RFID*-Komponente an die Steuerung gemeldet.

Tabelle 8-60 Mögliche Fehlercodes der RFID-Komponenten

Kommando	Code	Bedeutung
Status OK:	0x00	kein Fehler (Standardwert)
Anwesenheitsfehler	0x01	kein MDS (ID==0) vor dem SLG
Unbekannter Befehl:	0x05	Die SIMIT-Komponente unterstützt den an das ASM gesendeten Befehl nicht.
Adressfehler:	0x0	• Fehler beim Zugriff auf die Daten im MDS • Bereichsüberschreitung bei MDS-Zugriff
Antennenfehler:	0x1C	• Schreib/Lese-Befehle bei ausgeschalteter Antenne • Falsche Antennenbefehle (2x An / 2x Aus)

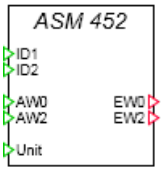
Der Signalaustausch der RFID-Komponenten mit der Steuerung erfolgt sowohl über Ein- und Ausgangsworte der Kopplung für die zyklische Kommunikation, als auch azyklisch über entsprechend im Komponententyp realisierte Kommunikationsfunktionen. Zur azyklischen Kommunikation ist der Unit-Eingang der ASM-Komponente mit dem Unit-Konnektor zu verbinden. Der Unit-Konnektor beinhaltet die Subsystem-Nummer, die Slave-Nummer und die Slot-Nummer des ASM aus der Hardware-Konfiguration. Er kann per Drag & Drop aus der Hardware-Ansicht im Eigenschaftsfenster der SIMIT Unit-Kopplung in das Diagramm übernommen und mit der ASM-Komponente verbunden werden. Wenn Sie den Unit-Konnektor manuell parametrieren, verwenden Sie die Subsystem-Nummer, die Slave-Nummer und die Slot-Nummer in der Form "[#][#][#]".

In folgender Abbildung sehen Sie beispielhaft eine zur Darstellung in der Abbildung oben passende Hardware-Konfiguration mit einer vierkanaligen Anschaltung:



ASM452 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM452* bildet eine zweikanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bzw. *AW2*, *EW2* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* und *ID2* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-61 Unterstützte Befehle des *ASM452*

Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhandenen MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS

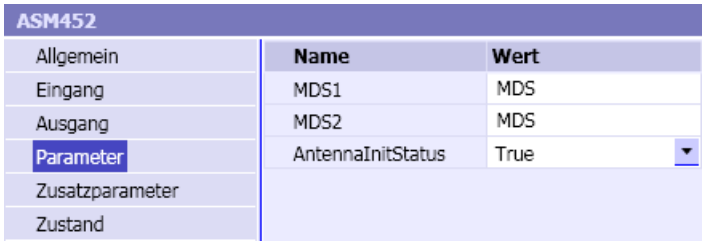
Kommando	Code	Bedeutung
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS-Status aus einem vorhandenen MDS

Parameter

Die Parameter *MDS1* und *MDS2* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bzw. *ID2* zugegriffen werden soll.

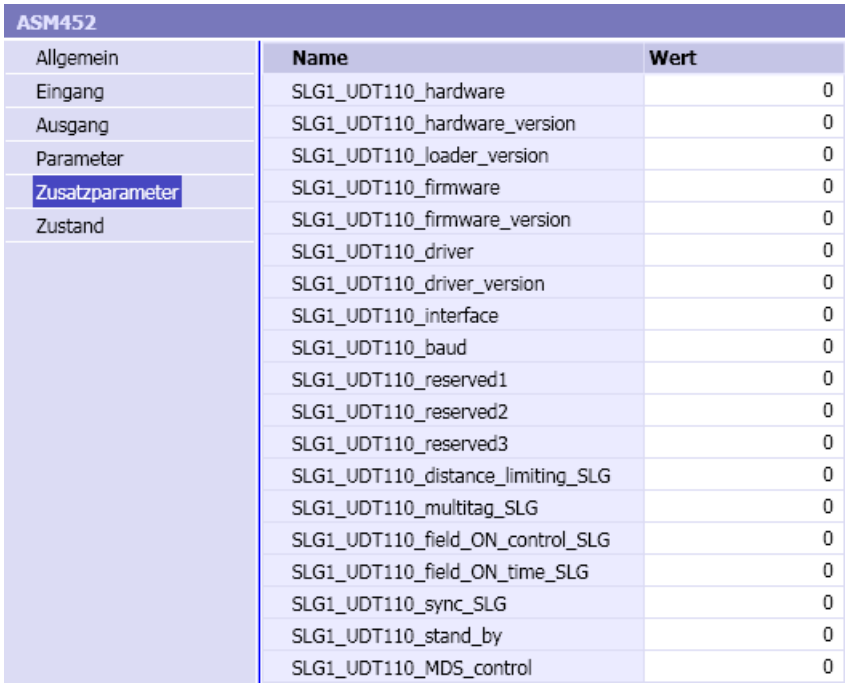
Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



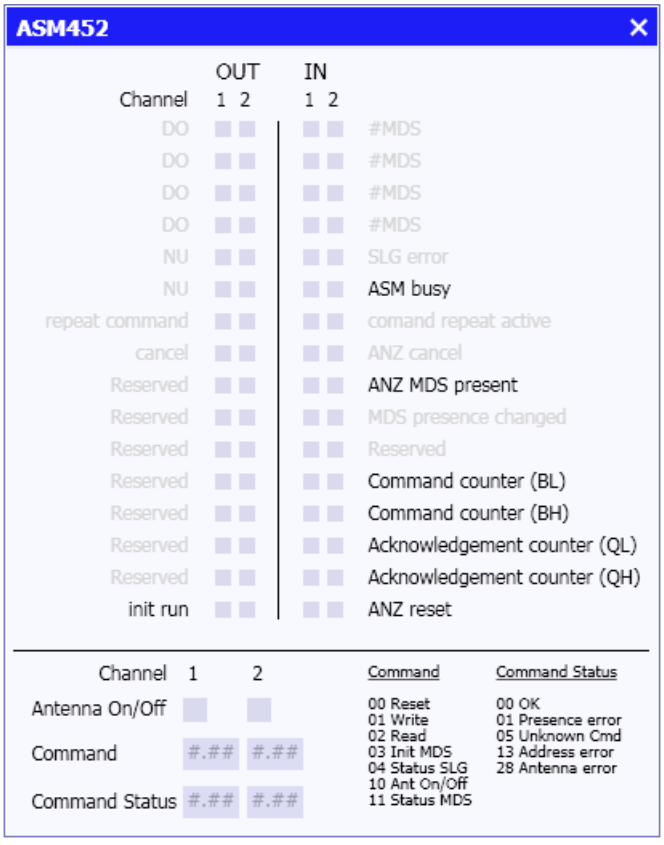
Zusatzparameter

Als Zusatzparameter sind für jedes der beiden simulierten Schreib/Lesegeräte (SLG) die Statuswerte entsprechend dem UDT110 als online-änderbare Parameter vorhanden.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der beiden Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der beiden Kanäle angezeigt.

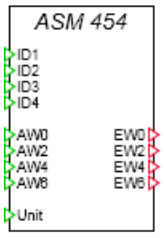


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM454 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM454* bildet eine vierkanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

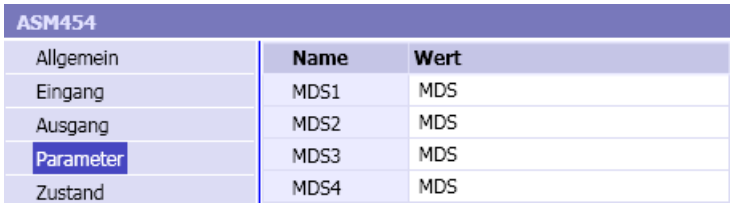
Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bis *AW6*, *EW6* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* bis *ID4* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-62 Unterstützte Befehle des AMS454

Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS

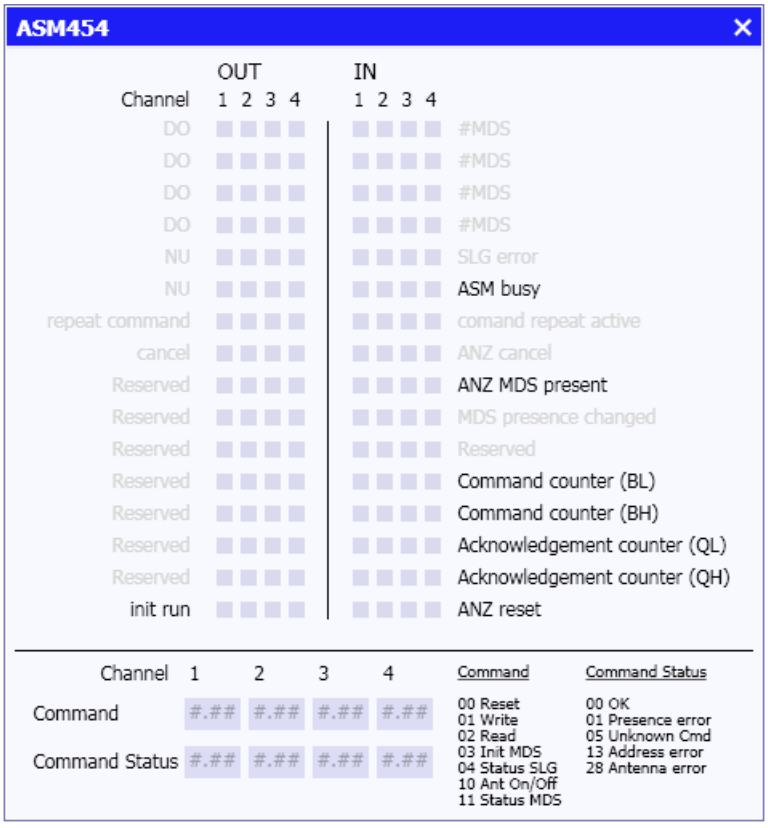
Parameter

Die Parameter *MDS1* bis *MDS4* geben an, auf welchen Bereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bis *ID4* zugegriffen werden soll. In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der vier Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der vier Kanäle angezeigt.

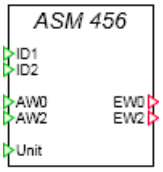


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM456 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM456* bildet eine zweikanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bzw. *AW2*, *EW2* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* und *ID2* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-63 Unterstützte Befehle des ASM456

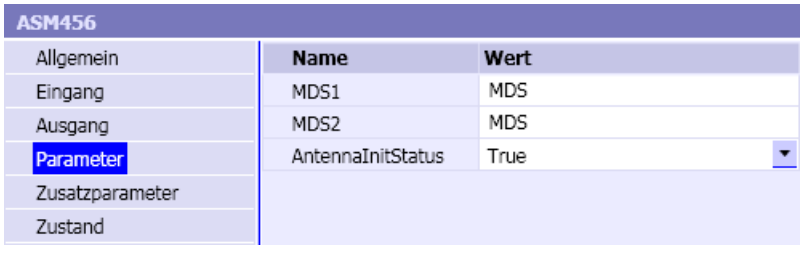
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS-Status aus einem vorhanden MDS

Parameter

Die Parameter *MDS1* und *MDS2* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bzw. *ID2* zugegriffen werden soll.

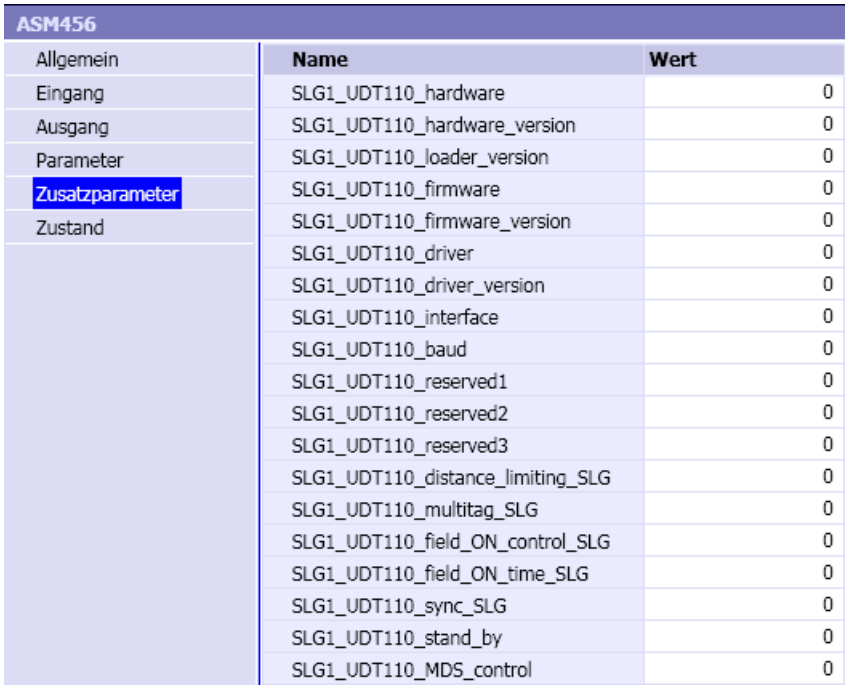
Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



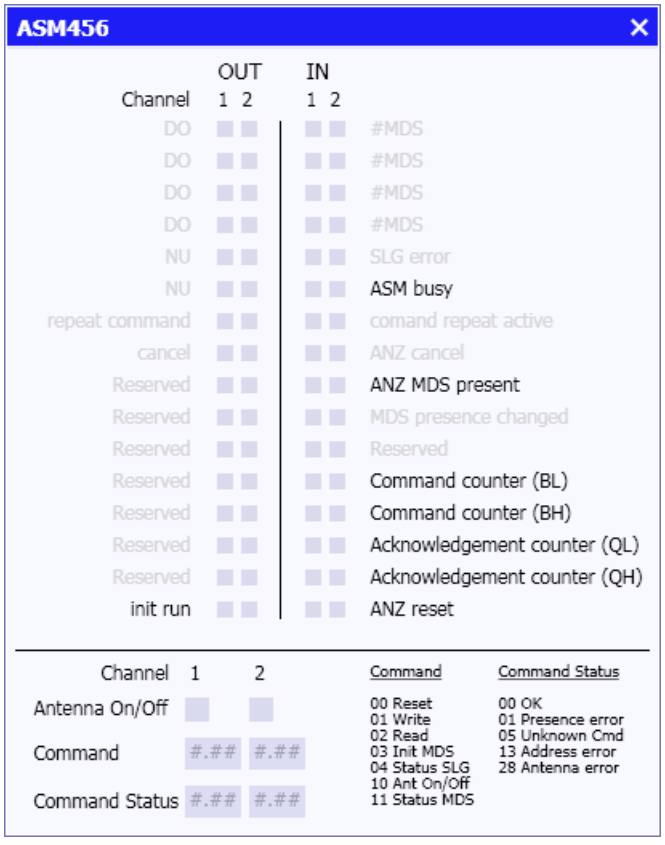
Zusatzparameter

Als Zusatzparameter sind für jedes der beiden simulierten Schreib/Lesegeräte (SLG) die Statuswerte entsprechend dem UDT110 als online-änderbare Parameter vorhanden.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der beiden Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der beiden Kanäle angezeigt.

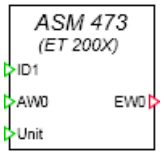


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM473 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM473* bildet eine einkanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über den Ein- und Ausgang *AW0*, *EW0* werden die zyklischen Daten mit der Steuerung ausgetauscht. Sie sind mit dem entsprechenden Ein-/Ausgangssignal der SIMIT Unit-Kopplung zu verknüpfen. Der Eingang *ID1* ist implizit mit dem Sensor der Fördertechniksimulation zu verbinden.

Tabelle 8-64 Unterstützte Befehle des *ASM473*

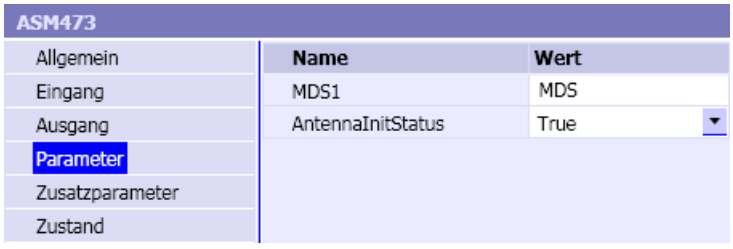
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS-Status aus einem vorhanden MDS

Parameter

Der Parameter *MDS1* gibt an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den Sensor *ID1* zugegriffen werden soll.

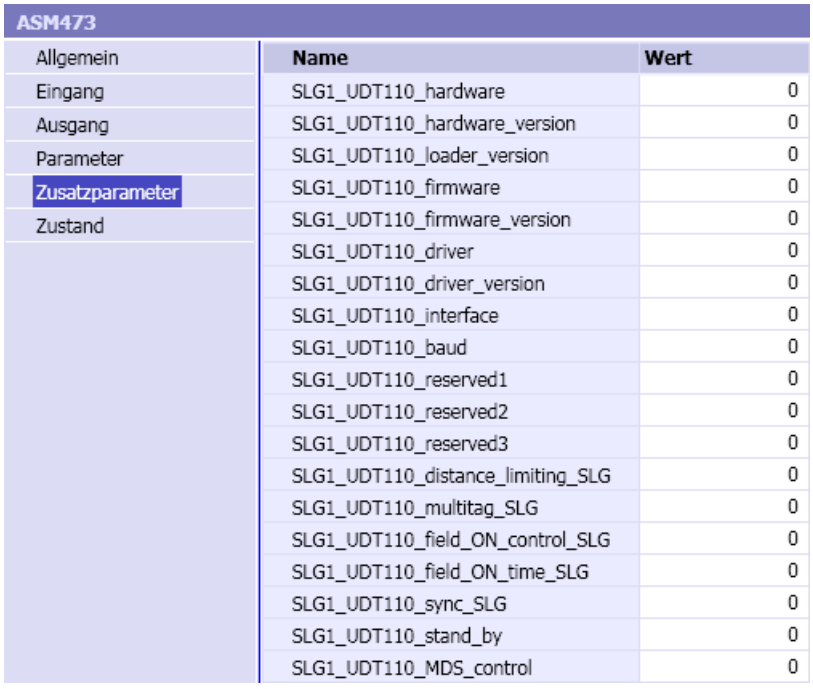
Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Zusatzparameter

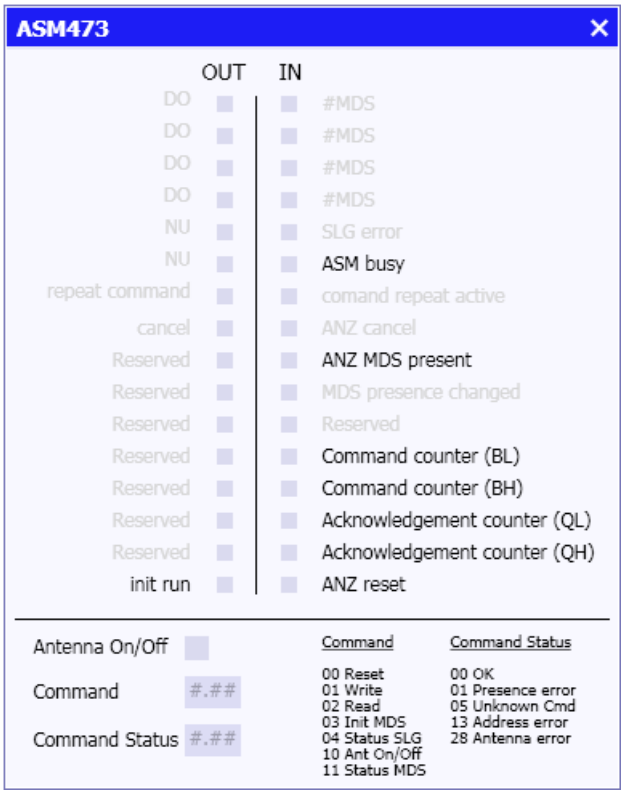
Als Zusatzparameter sind für das simulierte Schreib/Lesegerät (SLG) die Statuswerte entsprechend dem UDT110 als online-änderbare Parameter vorhanden.



Name	Wert
SLG1_UDT110_hardware	0
SLG1_UDT110_hardware_version	0
SLG1_UDT110_loader_version	0
SLG1_UDT110_firmware	0
SLG1_UDT110_firmware_version	0
SLG1_UDT110_driver	0
SLG1_UDT110_driver_version	0
SLG1_UDT110_interface	0
SLG1_UDT110_baud	0
SLG1_UDT110_reserved1	0
SLG1_UDT110_reserved2	0
SLG1_UDT110_reserved3	0
SLG1_UDT110_distance_limiting_SLG	0
SLG1_UDT110_multitag_SLG	0
SLG1_UDT110_field_ON_control_SLG	0
SLG1_UDT110_field_ON_time_SLG	0
SLG1_UDT110_sync_SLG	0
SLG1_UDT110_stand_by	0
SLG1_UDT110_MDS_control	0

Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände des Ein- und Ausgangsworts, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus angezeigt.

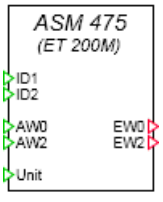


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM475 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM475* bildet eine zweikanalige Anschaltung für Identifikationssysteme am PROFIBUS DP oder PROFINET IO nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bzw. *AW2*, *EW2* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* und *ID2* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-65 Unterstützte Befehle des *ASM475*

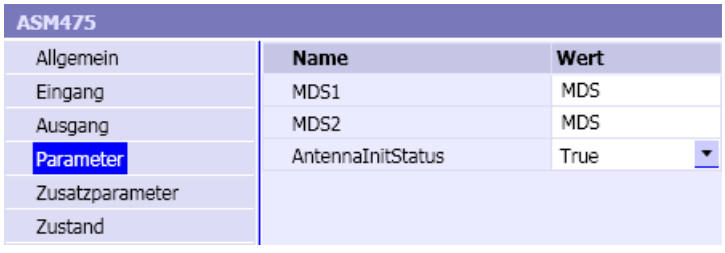
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS-Status aus einem vorhanden MDS

Parameter

Die Parameter *MDS1* und *MDS2* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bzw. *ID2* zugegriffen werden soll.

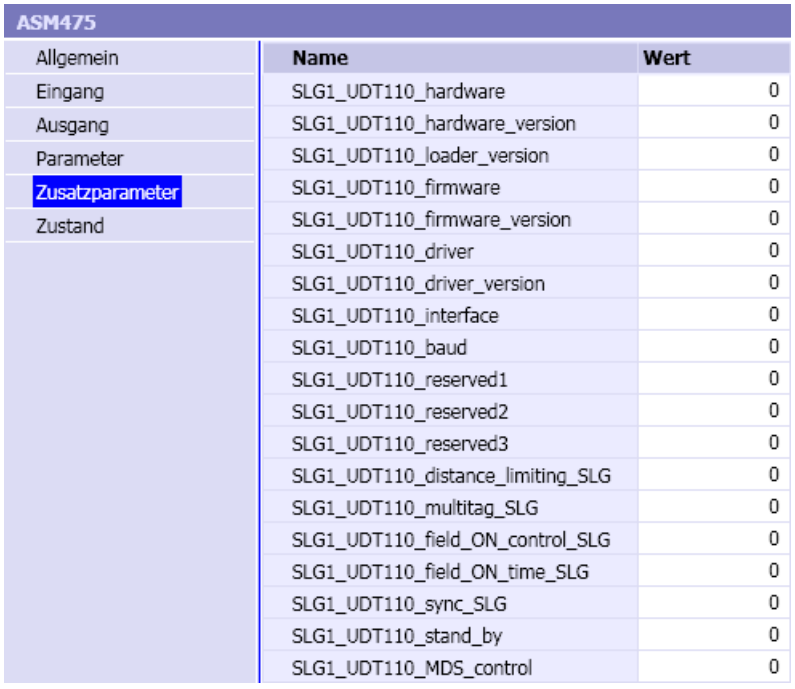
Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



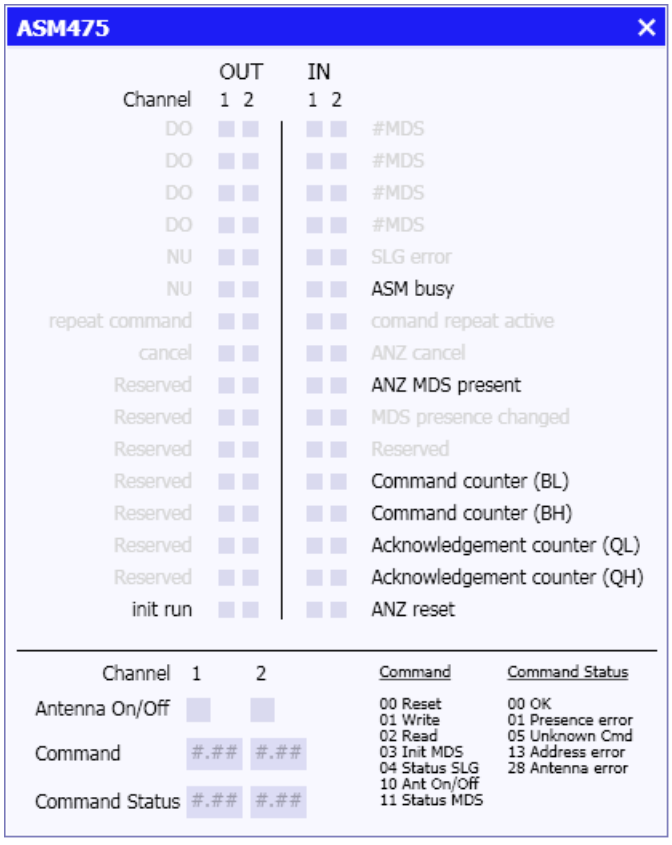
Zusatzparameter

Als Zusatzparameter sind für jedes der beiden simulierten Schreib/Lesegeräte (SLG) die Statuswerte entsprechend dem UDT110 als online-änderbare Parameter vorhanden.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der beiden Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der beiden Kanäle angezeigt.

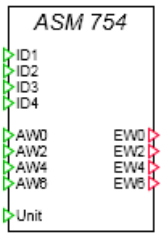


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM754 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM754* bildet eine vierkanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

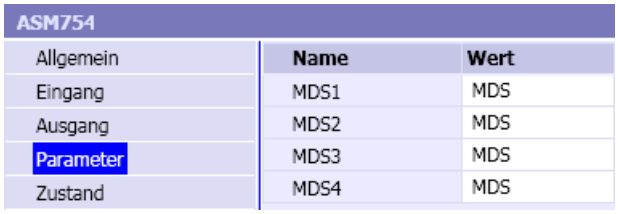
Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bis *AW6*, *EW6* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* bis *ID4* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-66 Unterstützte Befehle des ASM754

Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS

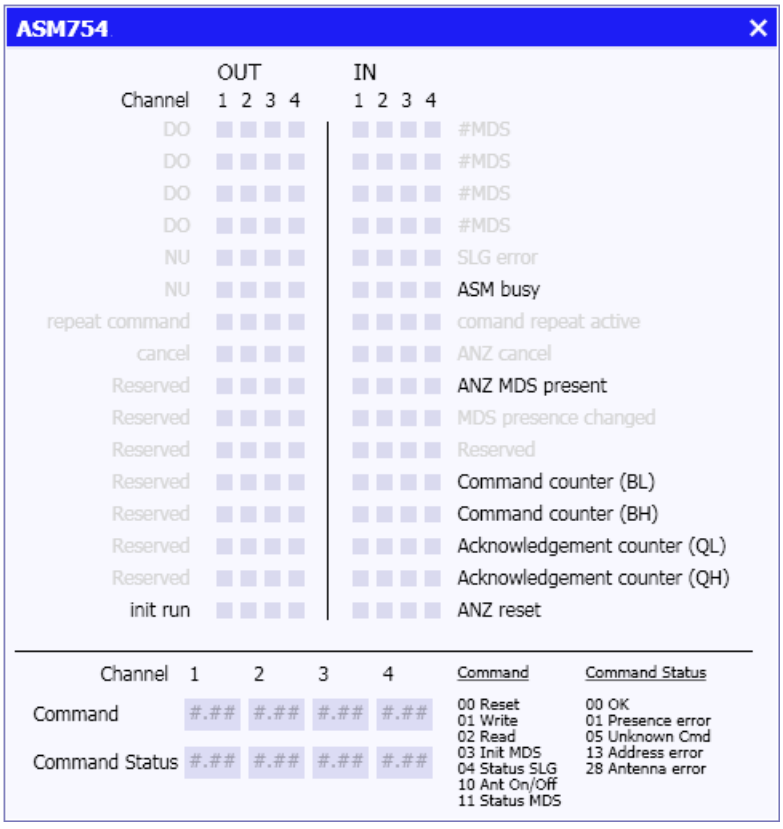
Parameter

Die Parameter *MDS1* bis *MDS4* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bis *ID4* zugegriffen werden soll. In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der vier Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der vier Kanäle angezeigt.



Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM850 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM850* bildet eine einkanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über den Ein- und Ausgang *AW0*, *EW0* werden die zyklischen Daten mit der Steuerung ausgetauscht. Sie sind mit dem entsprechenden Ein-/Ausgangssignal der SIMIT Unit-Kopplung zu verknüpfen. Der Eingang *ID1* ist implizit mit dem Sensor der Fördertechniksimulation zu verbinden.

Tabelle 8-67 Unterstützte Befehle des ASM850

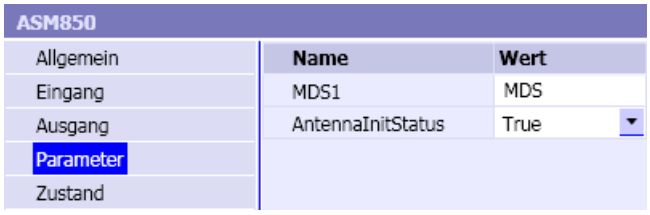
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)

Parameter

Der Parameter *MDS1* gibt an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den Sensor *ID1* zugegriffen werden soll.

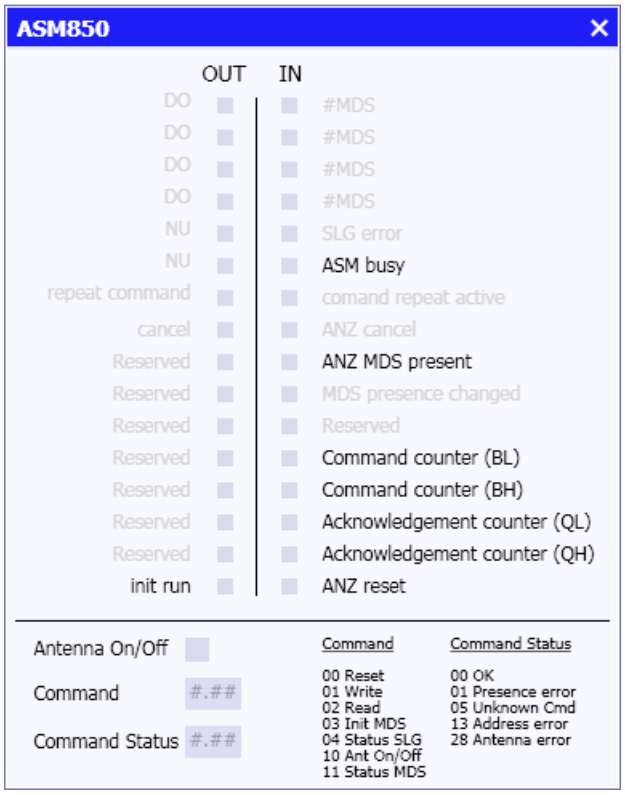
Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände des Ein- und Ausgangswortes, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus angezeigt.

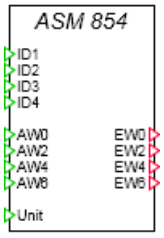


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

ASM854 – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *ASM854* bildet eine vierkanalige Anschaltung für Identifikationssysteme am PROFIBUS DP nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bis *AW6*, *EW6* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* bis *ID4* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-68 Unterstützte Befehle des *ASM854*

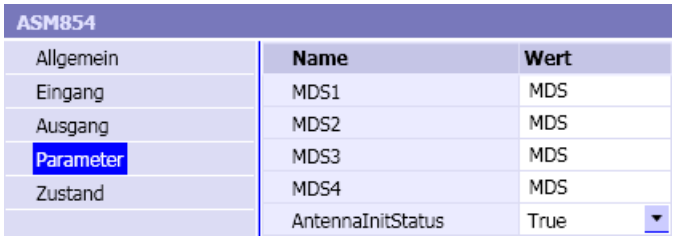
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)

Parameter

Die Parameter *MDS1* bis *MDS4* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bis *ID4* zugegriffen werden soll.

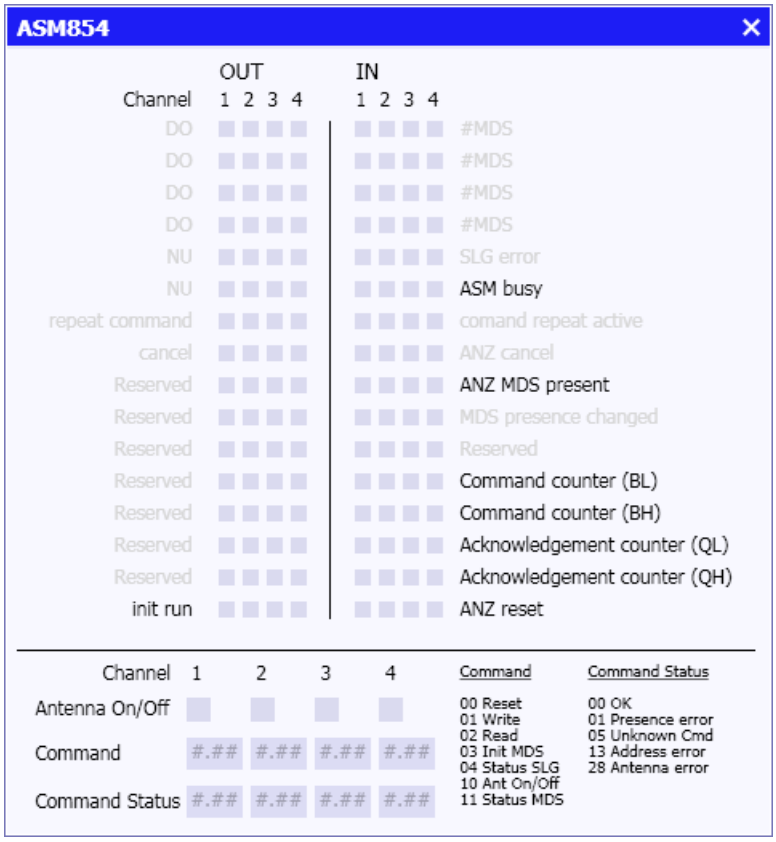
Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der vier Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der vier Kanäle angezeigt.

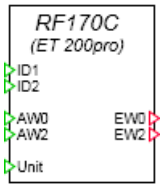


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

RF170C – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *RF170C* bildet eine zweikanalige Anschaltung für Identifikationssysteme am PROFIBUS DP oder PROFINET IO nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bzw. *AW2*, *EW2* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* und *ID2* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-69 Unterstützte Befehle des *RF170C*

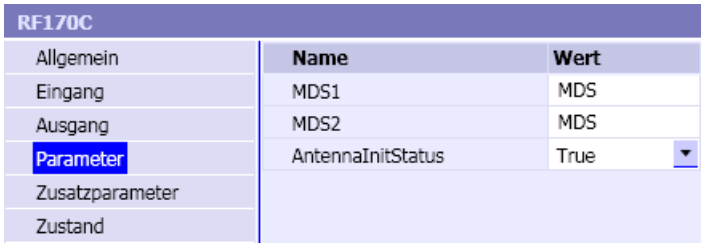
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS-Status aus einem vorhanden MDS

Parameter

Die Parameter *MDS1* und *MDS2* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bzw. *ID2* zugegriffen werden soll.

Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Zusatzparameter

Als Zusatzparameter sind für jedes der beiden simulierten Schreib/Lesegeräte (SLG) die Statuswerte entsprechend dem UDT110 als online-änderbare Parameter vorhanden.

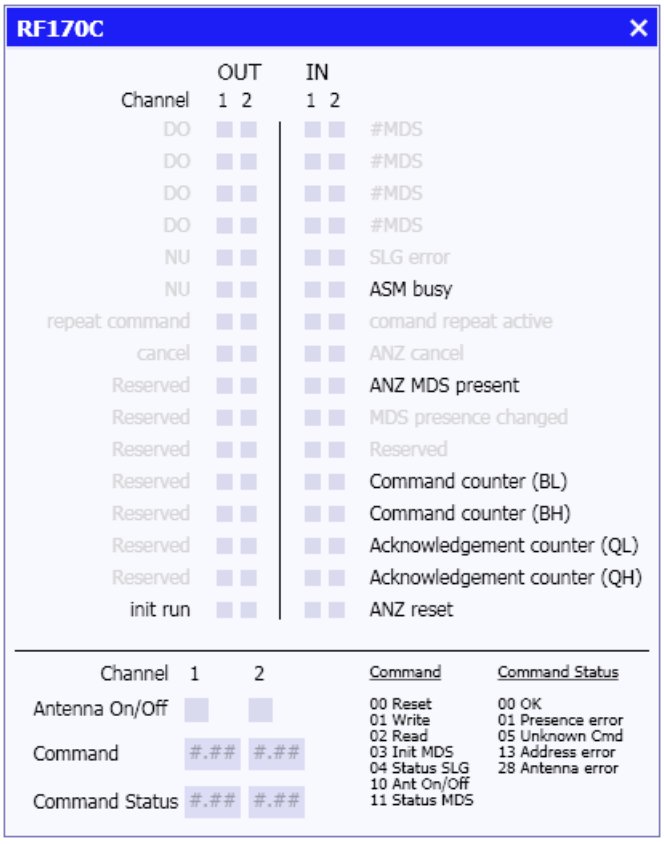
RF170C

- Allgemein
- Eingang
- Ausgang
- Parameter
- **Zusatzparameter**
- Zustand

Name	Wert
SLG1_UDT110_hardware	0
SLG1_UDT110_hardware_version	0
SLG1_UDT110_loader_version	0
SLG1_UDT110_firmware	0
SLG1_UDT110_firmware_version	0
SLG1_UDT110_driver	0
SLG1_UDT110_driver_version	0
SLG1_UDT110_interface	0
SLG1_UDT110_baud	0
SLG1_UDT110_reserved1	0
SLG1_UDT110_reserved2	0
SLG1_UDT110_reserved3	0
SLG1_UDT110_distance_limiting_SLG	0
SLG1_UDT110_multitag_SLG	0
SLG1_UDT110_field_ON_control_SLG	0
SLG1_UDT110_field_ON_time_SLG	0
SLG1_UDT110_sync_SLG	0
SLG1_UDT110_stand_by	0
SLG1_UDT110_MDS_control	0

Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der beiden Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der beiden Kanäle angezeigt.

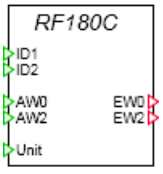


Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

RF180C – Anschaltung für Identifikationssysteme

Symbol



Funktion

Der Komponententyp *RF180C* bildet eine zweikanalige Anschaltung für Identifikationssysteme am PROFINET IO nach. Unterstützt werden die in folgender Tabelle gelisteten Kommandos.

Alle Ein- und Ausgänge des Komponententyps sind ganzzahlig (Integer). Am Eingang *Unit* ist die Slave- und Moduladresse des ASM mit Hilfe des Unit-Konnektors vorzugeben. Über die Ein- und Ausgänge *AW0*, *EW0* bzw. *AW2*, *EW2* werden die zyklischen Daten für jeden Kanal mit der Steuerung ausgetauscht. Sie sind mit den entsprechenden Ein-/Ausgangssignalen der SIMIT Unit-Kopplung zu verknüpfen. Die Eingänge *ID1* und *ID2* sind implizit mit den Sensoren der Fördertechniksimulation zu verbinden.

Tabelle 8-70 Unterstützt Befehle des *RF180C*

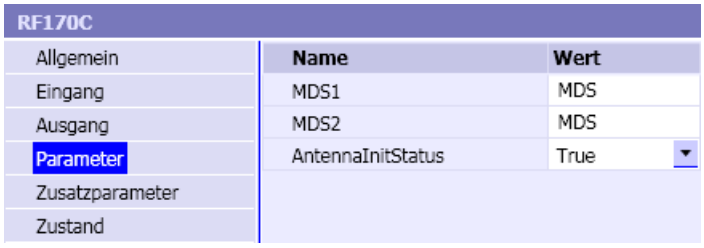
Kommando	Code	Bedeutung
Reset	0x00	Reset des SLG
Write	0x01	Schreiben von Daten in einen vorhandenen MDS
Read	0x02	Lesen von Daten aus einem vorhanden MDS
Init MDS	0x03	Initialisieren des Datenbereiches eines MDS
Status SLG	0x04	Auslesen des Status des SLG
Antenna On/Off	0x0A	An/Ausschalten der Antenne (falls vorhanden)
MDS Status	0x0B	Lesen des MDS-Status aus einem vorhanden MDS

Parameter

Die Parameter *MDS1* und *MDS2* geben an, auf welchen Speicherbereich des MDS von Fördergutkomponenten über den jeweiligen Sensor *ID1* bzw. *ID2* zugegriffen werden soll.

Der Parameter *AntennaInitStatus* gibt an, mit welchem Anfangsantennenstatus das SLG im ASM beim Start der Simulation initialisiert werden soll: True := Antenne ein, False := Antenne aus.

In folgender Abbildung sehen Sie die Parameter mit ihren Vorbelegungen.



Zusatzparameter

Als Zusatzparameter sind für jedes der beiden simulierten Schreib/Lesegeräte (SLG) die Statuswerte entsprechend dem UDT110 als online-änderbare Parameter vorhanden.

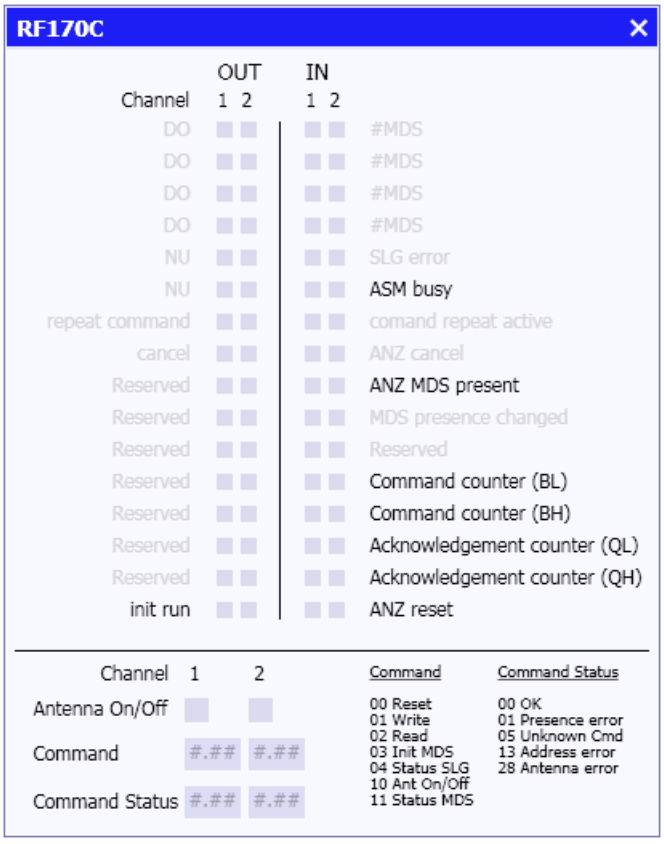
RF170C

- Allgemein
- Eingang
- Ausgang
- Parameter
- Zusatzparameter
- Zustand

Name	Wert
SLG1_UDT110_hardware	0
SLG1_UDT110_hardware_version	0
SLG1_UDT110_loader_version	0
SLG1_UDT110_firmware	0
SLG1_UDT110_firmware_version	0
SLG1_UDT110_driver	0
SLG1_UDT110_driver_version	0
SLG1_UDT110_interface	0
SLG1_UDT110_baud	0
SLG1_UDT110_reserved1	0
SLG1_UDT110_reserved2	0
SLG1_UDT110_reserved3	0
SLG1_UDT110_distance_limiting_SLG	0
SLG1_UDT110_multitag_SLG	0
SLG1_UDT110_field_ON_control_SLG	0
SLG1_UDT110_field_ON_time_SLG	0
SLG1_UDT110_sync_SLG	0
SLG1_UDT110_stand_by	0
SLG1_UDT110_MDS_control	0

Bedienfenster

Im Bedienfenster sehen Sie die aktuellen Zustände der Ein- und Ausgangsworte der beiden Kanäle, die vom FB45 angesprochen werden. Die grau dargestellten Bits werden von einer ASM-Komponente nicht ausgewertet. Im erweiterten Bedienfenster werden der Antennenstatus, der zuletzt ausgeführte Befehl und der zugehörige Befehlsstatus für jeden der beiden Kanäle angezeigt.



Siehe auch

Zugriff auf einen Datensatz oder Speicherbereich (Seite 194)

8.3.4 Komponententypen für Fördertechniksimulation selbst erstellen

Mit dem Komponententypeditor *CTE* von SIMIT sind Sie in der Lage, eigene Komponententypen zu erstellen, welche die Mechanismen des Lösungsverfahrens der Fördertechniksimulation nutzen. Sie können so die mit der Bibliothek *CONTEC* gelieferten Komponenten funktional erweitern, also beispielsweise komplexere Transportvorgänge realisieren, die in den gelieferten Bibliothekskomponenten nicht berücksichtigt sind und so die Tiefe Ihrer Simulationen von Förderanlagen erhöhen. Sie können auch völlig eigene Komponententypen erstellen und so die Bibliothek *Fördertechnik* erweitern und den bei Ihnen vorliegenden Besonderheiten anpassen.

Bei der Erstellung von Komponententypen sind drei Aspekte zu berücksichtigen:

- die topologischen Aspekte,
- die Anbindung an das Lösungsverfahren und
- das spezifische Verhalten der Komponente.

Der toplogische Aspekt wird durch entsprechende Ergänzungen der Komponententypdefinition abgedeckt. Für die Anbindung an das Lösungsverfahren werden spezielle Verbindungstypen bereitgestellt.

Im Übrigen bilden die allgemein im Handbuch "SIMIT – Component Type Editor" beschriebenen Eigenschaften von Komponenten die Grundlage der Erstellung von Komponententypen zur Fördertechniksimulation. Die allgemeinen Eigenschaften sind ohne Einschränkung auch für diese Komponententypen gültig.

8.3.4.1 Topologische Eigenschaften

Die Topologie der modellierten Förderanlage wird beim Übersetzen des Simulationsprojekts aus der Zusammenschaltung der Fördermittelkomponenten automatisch ermittelt. Somit muss jede Komponente die sie betreffende topologische Information, also die Information, wie sie topologisch im System zu betrachten ist, bereitstellen. Unter topologischen Gesichtspunkten ist es erforderlich, zu wissen, wie für eine Komponente die Bezugsrichtungen für Größen des fördertechnischen Systems definiert sind und wie der Datenaustausch der Komponente mit dem Lösungsverfahren gerichtet ist.

Elemente des fördertechnischen Systems mit topologischer Information sind:

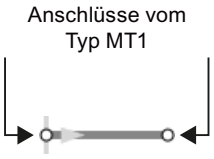
- Abschnitte von Förderstrecken
- Verzweigungen
- Ränder

Für jedes dieser Elemente sind seine Beziehungen zu den topologischen Anschlüssen der Fördertechnikkomponente zu definieren. Zur Definition öffnen Sie einfach den Topologie-Editor mit einem Doppelklick auf das Element *Topologie* in der Navigation des Komponententyps.



Topologische Anschlüsse (Verbindungstyp MT1)

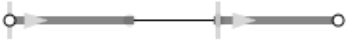
Der Verbindungstyp *MT1* kennzeichnet die Anschlüsse von Fördermittelkomponenten über die sie miteinander verbunden werden. Aus den miteinander verbundenen Anschlüssen dieses Typs und der topologischen Information der einzelnen Fördermittelkomponenten wird die Topologie der Förderanlage gewonnen. Der Typ *MT1* ist damit ein rein topologischer Verbindungstyp. Er trägt folglich auch keine Signale. Anschlüsse dieses Typs werden daher im Folgenden kurz "topologische Anschlüsse" genannt. Topologische Anschlüsse werden durch Kreise dargestellt.



Die Anschlüsse können durch Aufeinanderlegen verbunden werden. Beide Anschlüsse werden nach dem Verbinden ausgeblendet.



Werden topologische Anschlüsse im Diagrammeditor durch eine Verbindungslinie miteinander verbunden, dann werden die verbundenen Anschlüsse ausgeblendet. Sie sind nur noch als schwacher Schatten sichtbar. Anschlüsse des Typs MT1 können nicht mehrfach verbunden werden, sie stellen immer nur eine 1:1-Verbindung her.



Topologie eines Streckenabschnitts

Mit der Topologiebeschreibung eines Streckenabschnitts wird seine Bezugsrichtung festgelegt. Beispielsweise wird mit der Definition

```
FROM a TO b;
```

der Streckenabschnitt mit einer Bezugsrichtung vom Anfangspunkt `a` zum Endpunkt `b` festgelegt. Der Anfangs- bzw. Endpunkt kann im Komponententyp als Anschluss vom Typ *MT1* oder als Verzweigung oder Rand definiert sein.

In Bezugsrichtung werden Geschwindigkeiten positiv gezählt. Gleichzeitig ist das auch die Vorzugsrichtung, die das Lösungsverfahren zur Ermittlung der Berechnungsreihenfolge auswertet.

Topologie der Verzweigung

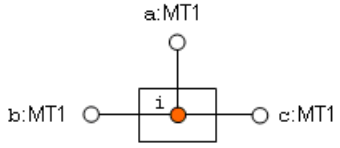
Eine Verzweigung wird als interner Knoten in einem Komponententyp in der Topologiebeschreibung wie folgt definiert:

```
INTERNAL_NODE i;
```

Zusätzlich sind dieser Verzweigung topologische Anschlüsse der Komponente zuzuordnen. Drei Anschlüsse können in der Topologiebeschreibung beispielsweise wie folgt zugeordnet werden:

```
FROM i TO a;
FROM i TO b;
FROM i TO c;
```

Die drei Anschlüsse a, b und c sind im Komponententyp als Anschlüsse vom Typ *MT1* zu definieren. Das Schema der damit definierten topologischen Struktur des Komponententyps zeigt folgende Abbildung:



Eine Verzweigung kann auch dadurch definiert sein, dass mehrere Streckenabschnitte einen gemeinsamen Anfangs- oder Endpunkt haben, der als topologischer Anschluss vom Typ *MT1* definiert ist.

Topologie des Randes

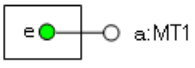
In der topologischen Beschreibung eines Komponententyps wird ein Randpunkt als externer Knoten `e` wie folgt definiert:

```
EXTERNAL_NODE e;
```

Weiter ist dieser externe Knoten mit mindestens einem topologischen Anschluss der Komponente zu verbinden. Die Topologiebeschreibung ist beispielsweise gemäß

```
FROM e TO a;
```

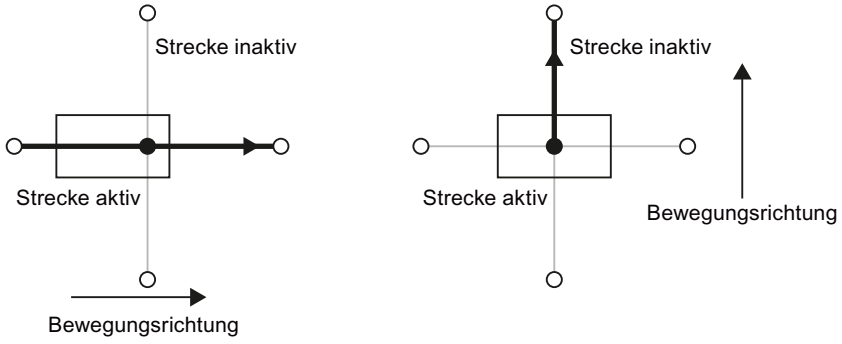
zu ergänzen. Der topologische Anschluss a ist im Komponententyp als Anschluss vom Typ *MT1* zu definieren. Das Schema der damit definierten topologischen Struktur des Komponententyps zeigt folgende Abbildung.



Bestimmung der Folgestrecke

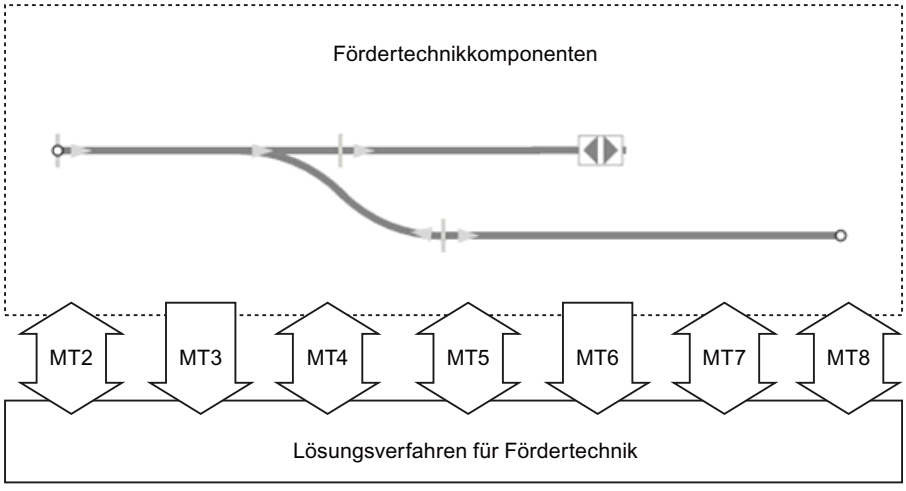
Ein Fördergut ist immer nur genau einem Streckenabschnitt im Streckennetz zugeordnet. Ein Transport kann nur stattfinden, wenn der Übergang auf den folgenden Streckenabschnitt eindeutig definiert ist. An Verzweigungsstellen darf daher nur genau eine mögliche Folgestrecke als aktiv gekennzeichnet sein, alle anderen Alternativstrecken müssen inaktiv sein.

Ein Sonderfall ergibt sich an einer Verzweigung als interner Knoten einer Komponente, sofern ein Fördergut mit seinem Umriss die Verzweigung überdeckt und genau einer der an die Verzweigung angeschlossenen Streckenabschnitte aktiv ist. In diesem Fall wird das Fördergut diesem einen aktiven Streckenabschnitt zugeordnet.



8.3.4.2 Anbindung an das Lösungsverfahren

Die Streckenabschnitte, Verzweigungen und Randpunkte einer Komponente werden über spezielle Anschlüsse datentechnisch mit dem Lösungsverfahren für die Fördertechnik verbunden. Zu diesem Zweck stehen sieben verschiedene Verbindungstypen *MT2* bis *MT8* zur Verfügung. Ihr Einsatz in Fördertechnikkomponenten ist in den folgenden Abschnitten erläutert.



Anschlüsse der Verbindungstypen *MT2* bis *MT8* stellen eine Verbindung zum Lösungsverfahren her und sind somit als **nicht sichtbar** im Symbol der Komponente zu setzen. In den Eigenschaften des Anschlusses ist dazu die Verwendung "*Nur im Eigenschaftsfenster*" oder "*Nur im CTE*" zu setzen.

Eigenschaft	Wert
Verwendung	Nur im Eigenschaftsfenster
Voreinstellung Sichtbarkeit	

Anschlüsse vom Verbindungstyp MT2 bis MT7 sind in Richtung OUT zu definieren, ein Anschluss vom Verbindungstyp MT8 in Richtung IN.

Siehe auch

Verbindungstyp MT2 für Streckenabschnitte (Seite 930)

Verbindungstyp MT3 für Blocker (Seite 931)

Verbindungstyp MT4 für Sensoren (Seite 932)

Verbindungstyp MT5 zum Aufsetzen von Fördergütern (Seite 932)

Verbindungstyp MT6 für Positionen (Seite 933)

Verbindungstyp MT7 zur Parametrierung von Fördergütern (Seite 934)

Verbindungstyp MT8 zur Übergabe von Fördergütern an Randpunkten (Seite 936)

Verbindungstyp MT2 für Streckenabschnitte

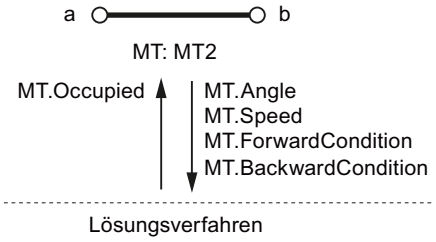
Über einen Anschluss vom Typ *MT2* kann ein Streckenabschnitt Größen mit dem Lösungsverfahren austauschen. Für einen Anschluss mit dem Namen *MT* wird die Topologiebeschreibung des Streckenabschnitts dazu wie folgt ergänzt:

```
FROM a TO b: MT;
```

Der Anschluss ist immer in Richtung OUT zu definieren. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene, den Streckenabschnitt betreffende Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT2	OUT	1

Folgende Abbildung zeigt die Ein- und Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



Die Ausgangssignale werden für den Streckenabschnitt in der Komponente festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **Angle** (integer)
   Der Winkel des Kreissegmentes in Grad, das diese Teilstrecke beschreibt. Positive Winkel drehen im Uhrzeigersinn, negative Winkel gegen den Uhrzeigersinn. Bei einer geraden Verbindung ist hier null anzugeben.
   Diese Angabe wird ausschließlich während der Initialisierung der Simulation ausgewertet, eine Veränderung des Wertes während der zyklischen Berechnung ist ohne Auswirkung.
2. **Speed** (analog)
   Die Geschwindigkeit in m/s, mit der Fördergüter über diesen Streckenabschnitt bewegt werden sollen. Positive Werte bewegen in der Vorzugsrichtung, negative Werte entgegen der Vorzugsrichtung.

3. **ForwardCondition** (integer)
   Legt fest, ob diese Teilstrecke in Vorzugsrichtung aktiv ist (0) oder nicht (−1).
4. **BackwardCondition** (integer)
   Legt fest, ob diese Teilstrecke gegen die Vorzugsrichtung aktiv ist (0) oder nicht (−1).

Über das Eingangssignal erhält die Komponente vom Lösungsverfahren Informationen über die Förderstrecke:

1. **Occupied** (bool)
   Gibt an, ob diese Teilstrecke vom Umriss eines oder mehrerer Fördergüter berührt wird.

Der Verbindungstyp *MT2* wird beispielsweise in den Bibliothekskomponenten *Rail-S4* und *Conveyor-S4* verwendet.

Verbindungstyp MT3 für Blocker

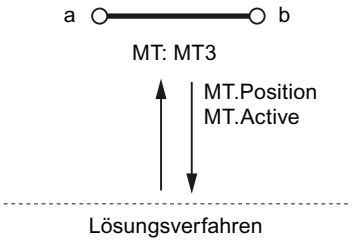
Über einen Anschluss vom Typ *MT3* kann ein Streckenabschnitt Größen mit dem Lösungsverfahren austauschen. Für einen Anschluss mit dem Namen *MT* wird die Topologiebeschreibung des Streckenabschnitts dazu wie folgt ergänzt:

```
FROM a TO b: MT;
```

Der Anschluss ist immer in Richtung OUT zu definieren. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT3	OUT	1

Folgende Abbildung zeigt die Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



Die Ausgangssignale werden für den Streckenabschnitt in der Komponente festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **Position** (analog)
   Die Position des Blockers auf dem Streckenabschnitt als prozentualer Wert der Gesamtlänge des Streckenabschnitts.
   Diese Angabe wird ausschließlich während der Initialisierung der Simulation ausgewertet, eine Veränderung des Wertes während der zyklischen Berechnung ist ohne Auswirkung.
2. **Active** (binary)
   Gibt an, ob der Blocker wirksam ist (True) oder unwirksam (False).

Der Verbindungstyp *MT3* wird beispielsweise in der Bibliothekskomponente *Conveyor-S4-Stopper* verwendet.

Verbindungstyp MT4 für Sensoren

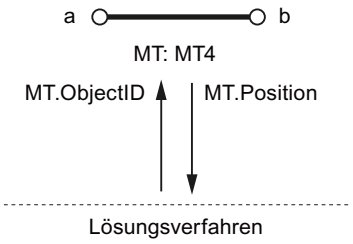
Über einen Anschluss vom Typ *MT4* kann ein Streckenabschnitt Größen mit dem Lösungsverfahren austauschen. Für einen Anschluss mit dem Namen *MT* wird die Topologiebeschreibung des Streckenabschnitts dazu wie folgt ergänzt:

```
FROM a TO b: MT;
```

Der Anschluss ist immer in Richtung OUT zu definieren. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT4	OUT	1

Folgende Abbildung zeigt die Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



Die Ausgangssignale werden für den Streckenabschnitt in der Komponente festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **Position** (analog)
   Die Position des Sensors auf dem Streckenabschnitt als prozentualer Wert der Gesamtlänge des Streckenabschnitts.
   Diese Angabe wird ausschließlich während der Initialisierung der Simulation ausgewertet, eine Veränderung des Wertes während der zyklischen Berechnung ist ohne Auswirkung.

Über das Eingangssignal erhält die Komponente vom Lösungsverfahren Informationen über den Streckenabschnitt:

1. **ObjectID** (integer)
   Die ID des Fördergutes, das über den Sensor aktuell erfasst wird. Falls kein Fördergut erfasst wird, ist dieser Wert null.

Der Verbindungstyp *MT4* wird beispielsweise in den Bibliothekskomponenten *Rail-S4* und *Conveyor-S4* verwendet.

Verbindungstyp MT5 zum Aufsetzen von Fördergütern

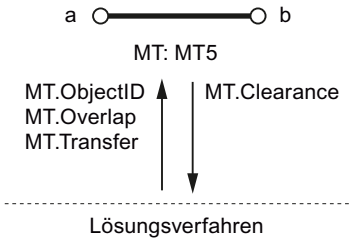
Über einen Anschluss vom Verbindungstyp *MT5* können Fördergüter auf einen Streckenabschnitt aufgesetzt werden. Die Topologiedefinition des Streckenabschnitts ist entsprechend zu ergänzen, für einen Anschluss mit Namen *MT* gemäß

```
FROM a TO b: MT;
```

Der Anschluss ist immer in Richtung OUT definiert. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT5	OUT	1

Folgende Abbildung zeigt die Ein- und Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



Die Ausgangssignale werden in der Komponente für den Streckenabschnitt festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **ObjectID** (integer)
   Die ID des Objektes, das auf diesen Streckenabschnitt aufgesetzt werden soll. Die Komponente erhält diese ID in der Regel mit Hilfe geeigneter Funktionsaufrufen. Weitere Informationen hierzu finden Sie in den folgenden Kapiteln:
   - _MT.GetObjectByName (Seite 939)
   - _MT.GetObjectByType (Seite 939)
2. **Overlap** (analog)
   Verschiebung in Millimetern in Richtung der Vorzugsrichtung. Damit kann der Aufsetzpunkt relativ verschoben werden. Allerdings muss sich der Aufsetzpunkt immer auf dem Streckenabschnitt befinden.
3. **Transfer** (bool)
   Wenn dieses Signal auf *True* gesetzt wird, startet das Lösungsverfahren den Aufsetzvorgang. Die Komponente darf das Signal nur genau einen Zyklus lang anstehen lassen und muss es dann wieder auf *False* zurücknehmen.

Über die Eingangssignale erhält die Komponente vom Lösungsverfahren Informationen über den Streckenabschnitt:

1. **Clearance** (analog)
   Die zur Verfügung stehende freie Strecke des Streckenabschnitts in Millimetern in Vorzugsrichtung gesehen.

Der Verbindungstyp *MT5* wird beispielsweise in den Bibliothekskomponenten *Rail-S4* und *Conveyor-S4* verwendet.

Verbindungstyp MT6 für Positionen

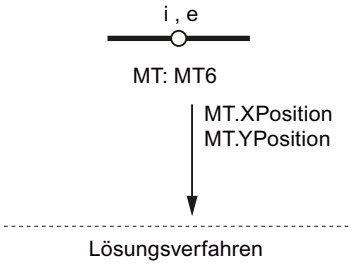
Die Positionen von Verzweigungen als interner Knoten wie auch von Randpunkten als externe Knoten müssen individuell parametriert werden. Die Topologiedefinition eines Knotens ist dafür zu ergänzen durch einen Anschluss über den die Position an das Lösungsverfahren übermittelt wird, für einen Anschluss *MT* wie folgt:

```
EXTERNAL_NODE E: MT;
INTERNAL_NODE I: MT;
```

Der Anschluss ist immer in Richtung OUT zu definieren. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene, den Knoten betreffende Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT6	OUT	1

Folgende Abbildung zeigt die Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



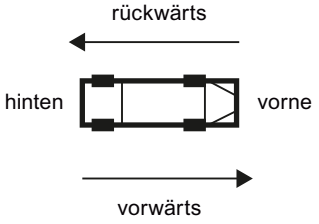
Die Ausgangssignale werden für den Knoten in der Komponente festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **XPosition** (analog)
   Die X-Position des Knotens in Millimetern bezogen auf die linke obere Ecke der Komponente.
   Diese Angabe wird ausschließlich während der Initialisierung der Simulation ausgewertet, eine Veränderung des Wertes während der zyklischen Berechnung ist ohne Auswirkung.
2. **YPosition** (analog)
   Die Y-Position des Knotens in Millimetern bezogen auf die linke obere Ecke der Komponente.
   Diese Angabe wird ausschließlich während der Initialisierung der Simulation ausgewertet, eine Veränderung des Wertes während der zyklischen Berechnung ist ohne Auswirkung.

Verbindungstyp MT7 zur Parametrierung von Fördergütern

Über einen Anschluss vom Typ *MT7* kann ein Fördergut Größen mit dem Lösungsverfahren austauschen. Eine Topologiebeschreibung ist dafür nicht notwendig.

Die Orientierung des Förderguts hängt von seiner Projektierung ab und entspricht der Darstellung des Grundsymbols im CTE.

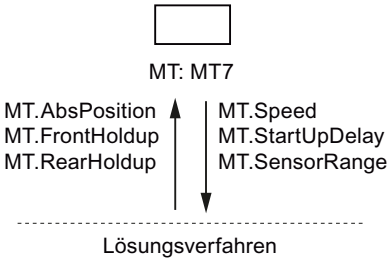


Der Anschluss vom Typ *MT7* ist immer in Richtung OUT zu definieren. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene, das Fördergut betreffende Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Eine Komponente darf höchstens einen Anschluss vom Typ MT7 besitzen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT7	OUT	1

Folgende Abbildung zeigt die Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



Die Ausgangssignale werden für das Fördergut in der Komponente festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **Speed** (analog)
   Die Geschwindigkeit des Fördergutes aus eigenem Antrieb in m/s. Eine positive Geschwindigkeit bewegt das Fahrzeug vorwärts, eine negative rückwärts.
   Diese Geschwindigkeit überlagert sich mit einer eventuell von der Schiene vorgegebenen Geschwindigkeit.
2. **StartUpDelay** (analog)
   Anfahrverzögerung in Sekunden. Wenn das Fördergut vom Lösungsverfahren aufgestaut wurde, wird diese Wartezeit eingehalten, bevor es nach dem Auflösen des Staus weiterfährt.
3. **SensorRange** (analog)
   Der prozentuale Anteil der Fördergutlänge, der von einem Sensor erfasst wird. Dieser Wert muss zwischen 0 und 100 % liegen.

Über die Eingangssignale erhält die Komponente vom Lösungsverfahren Informationen über das Fördergut:

1. **AbsPosition** (analog)
   Reserviert für spätere Erweiterung.
2. **FrontHoldup** (binary)
   Wenn dieses Signal auf *True* steht, befindet sich unmittelbar vor dem Fördergut ein Blocker oder ein anderes Fördergut.
3. **RearHoldup** (binary)
   Wenn dieses Signal auf *True* steht, befindet sich unmittelbar hinter dem Fördergut ein Blocker oder ein anderes Fördergut.

Der Verbindungstyp *MT7* wird beispielsweise in der Bibliothekskomponente *Vehicle* (Seite 891) verwendet.

Verbindungstyp MT8 zur Übergabe von Fördergütern an Randpunkten

An Randpunkten als externe Knoten können Fördergüter vom Lösungsverfahren an eine Komponente übergeben werden und umgekehrt. Sobald nicht mehr das Lösungsverfahren, sondern eine Komponente die Kontrolle über das Fördergut hat, wird die Position des Fördergutes vom Lösungsverfahren nicht mehr verändert. Das Fördergut bleibt aber auf dem Diagramm sichtbar und kann nun von der Komponente über entsprechende Systemfunktionen positioniert werden. Weitere Informationen hierzu finden Sie im Kapitel: _MT.SetPosition (Seite 941).

Auf diese Weise können komplexere Transportvorgänge in Fördermittelkomponenten realisiert werden, die mit dem auf einem Bahnmodell basierenden Lösungsverfahren nicht simuliert werden können.

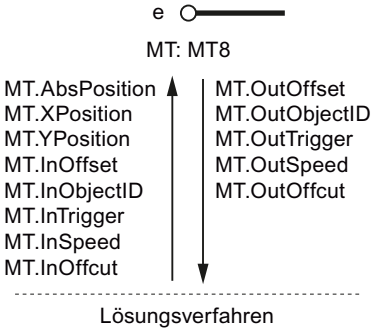
Über einen Anschluss vom Typ *MT8* kann ein Randpunkt als externer Knoten Größen mit dem Lösungsverfahren austauschen. Für einen Anschluss mit dem Namen *MT* wird die Topologiebeschreibung des externen Knotens dazu wie folgt ergänzt:

```
EXTERNAL_NODE E: MT;
```

Der Anschluss ist immer in Richtung IN zu definieren. Er schließt die Komponente an das Lösungsverfahren an, um verschiedene Größen zwischen dem Lösungsverfahren und der Komponente auszutauschen.

Name	Verbindungstyp	Richtung	Anzahl
MT	MT8	IN	1

Folgende Abbildung zeigt die Ausgangssignale mit der Richtung des Datenaustauschs zwischen der Komponente und dem Lösungsverfahren.



Die Ausgangssignale werden für den externen Knoten in der Komponente festgelegt bzw. berechnet und an das Lösungsverfahren gegeben:

1. **OutOffset** (analog)
   Die Länge der freien Strecke, vom Anschluss in die Komponente hinein gesehen. Ein negativer Wert bedeutet, dass von der Komponente aus gesehen ein Fördergut auf die vom Lösungsverfahren kontrollierte Nachbarstrecke hinausragt, in diesem Fall ist der Betrag dieses Wertes die Länge des überstehenden Teils des Fördergutes. Alle Angaben erfolgen in Millimetern.
2. **OutObjectID** (integer)
   Die Objekt-ID des Fördergutes, das von der Komponente aus gesehen auf die vom Lösungsverfahren kontrollierte Nachbarstrecke hinausragt. Null, falls kein Fördergut übersteht.

3. **OutTrigger** (binary)
   Die Komponente legt dieses Signal für genau einen Zyklus lang auf *True*, um anzuzeigen, dass die vom Lösungsverfahren kontrollierte Nachbarstrecke das Fördergut übernehmen muss.
4. **Out.Speed** (analog)
   Die Geschwindigkeit in m/s, mit der das Fördergut übergeben wird.
5. **OutOffcut** (analog)
   Die Zeit, die vom aktuellen Zyklus nicht mehr berücksichtigt werden konnte, weil das Fördergut den Übergabeknoten erreicht hat, gemessen in Sekunden. Das Lösungsverfahren muss im folgenden Zyklus diese Zeit berücksichtigen, um eine zusätzliche Bewegung zu berechnen.

Über die Eingangssignale erhält die Komponente vom Lösungsverfahren Informationen:

1. **AbsPosition** (analog)
   Reserviert für spätere Erweiterungen.
2. **XPosition** (analog)
   Die relative X-Position des mit dem Knoten verbundenen Anschlusses bezogen auf die linke obere Ecke der Komponente in Millimetern.
3. **YPosition** (binary)
   Die relative Y-Position des mit dem Knoten verbundenen Anschlusses bezogen auf die linke obere Ecke der Komponente in Millimetern.
4. **InOffset** (analog)
   Die Länge der freien Strecke, vom Anschluss der Komponente aus gesehen in Richtung der anliegenden Strecke außerhalb der Komponente. Ein negativer Wert bedeutet, dass von der vom Lösungsverfahren kontrollierten Nachbarstrecke aus gesehen ein Fördergut in diese Komponente hineinragt, in diesem Fall ist der Betrag dieses Wertes die Länge des hineinragenden Teils des Fördergutes. Alle Angaben erfolgen in Millimetern.
5. **InObjectID** (integer)
   Die Objekt-ID des Fördergutes, das von der vom Lösungsverfahren kontrollierten Strecke aus gesehen in die Komponente hineinragt. Null, falls kein Fördergut hineinragt.
6. **InTrigger** (binary)
   Das Lösungsverfahren legt dieses Signal für genau einen Zyklus lang auf *True*, um anzuzeigen, dass die Komponente das Fördergut übernehmen muss.
7. **InSpeed** (analog)
   Die Geschwindigkeit in m/s, mit der das Fördergut vom Lösungsverfahren zuletzt gefahren wurde.
8. **InOffcut** (analog)
   Die Zeit, die vom aktuellen Zyklus nicht mehr berücksichtigt werden konnte, weil das Fördergut den Übergabeknoten erreicht hat, gemessen in Sekunden. Die Komponente muss im folgenden Zyklus diese Zeit berücksichtigen, um eine zusätzliche Bewegung zu berechnen.

Der Verbindungstyp *MT8* wird beispielsweise in der Bibliothekskomponente *TransferCarriage* verwendet.

Fehler im Komponenten-Code

Falls in Fördertechnikkomponenten Schnittstellen zum Lösungsverfahren angelegt sind, aber nicht alle notwendigen Signale zur Verfügung gestellt werden, wird das Starten der Simulation mit einer entsprechenden Fehlermeldung verweigert.



8.3.4.3 Systemfunktionen

Beim Erstellen von eigenen Komponententypen können Sie verschiedene Systemfunktionen nutzen, um in der Simulation auf Fördergüter zuzugreifen.

Alle im Folgenden beschriebenen Funktionen liefern einen Fehlercode als Rückgabewert. Damit eine Funktion in einer übergebenen Variablen Rückgaben liefern kann, muss diese Variable als Feld (Vektor), in diesem Fall mit der Dimension eins, definiert werden. Die Bedeutung dieses Rückgabewertes entnehmen Sie folgender Tabelle.

Tabelle 8-71 Rückgabewerte der Fördertechnik Systemfunktionen

Wert	Bedeutung
0	Kein Fehler
−2	Das Fördergut kann nicht direkt positioniert werden, da es aktuell vom Lösungsverfahren verwaltet wird.
−3	Das Fördergut kann weder positioniert noch gedreht werden, weil es derzeit keiner Förderstrecke zugeordnet ist.
−10	Es existiert kein Fördergut mit der angegebenen ID.
−11	Das Fördergut mit der angegebenen ID steht nicht zur Verfügung.[1)]
−12	Es existiert kein Fördergut mit dem angegebenen Namen.
−13	Das Fördergut mit dem angegebenen Namen steht nicht zur Verfügung.
−14	Es existiert kein Fördergut vom angegebenen Typ.
−15	Fördergut vom angegebenen Typ steht nicht zur Verfügung.
−20	Die Komponente besitzt kein Signal mit dem angegebenen Namen.
−21	Der angegebene Signaltyp kann in diesem Zusammenhang nicht benutzt werden.
−22	Der Adressbereich des zu lesenden oder schreibenden Byte-Arrays wird überschritten.
−1000	Allgemeiner Fehler

[1] Ein Fördergut steht nicht zur Verfügung, wenn es bereits reserviert oder einer Förderstrecke zugeordnet worden ist.

_MT.GetObjectByName

Die Funktion *_MT.GetObjectByName* liefert die ID des Förderguts mit dem Instanznamen *name*, falls dieses als Fördergut zur Verfügung steht, ansonsten −1. Der Rückgabewert ist vom Typ *long*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- name (string)
  Der Instanzname des Förderobjektes.

_MT.GetObjectByType

Die Funktion *_MT.GetObjectByType* liefert die ID eines Fördergutes vom Komponententyp *type*, falls dieses als Fördergut zur Verfügung steht, ansonsten −1. Der Rückgabewert ist vom Typ *long*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- type (string)
  Der Typ des gesuchten Förderobjektes.
- store (string)
  Der Name der Materialliste, in dem sich das Objekt befindet. Wird hier ein Leerstring übergeben, werden alle vorhandenen Listen durchsucht.

_MT.Restore

Die Funktion *_MT.Restore* gibt ein Fördergut frei. Der Rückgabewert ist *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des freizugebenden Förderobjektes.
- delay (bool)
  Gibt an, ob die Rückgabe um einen Zyklus verzögert werden soll, damit das Fördergut zur sofortigen Wiederverwendung zur Verfügung steht.

_MT.GetWidth

Die Funktion *_MT.GetWidth* liefert die Breite eines Förderguts in Millimetern. Der Rückgabewert ist vom Typ *double*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.

_MT.GetHeight

Die Funktion *_MT.GetHeight* liefert die Höhe eines Förderguts in Millimetern. Der Rückgabewert ist vom Typ *double*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.

_MT.GetDepth

Die Funktion *_MT.GetDepth* liefert die Tiefe eines Förderguts in Millimetern. Der Rückgabewert ist vom Typ *double*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.

_MT.GetAngle

Die Funktion *_MT.GetAngle* liefert den Winkel eines Förderguts relativ zur Tangente des Streckenabschnitts, auf dem er sich befindet in Grad (−180° .. +180°). Der Rückgabewert ist vom Typ *integer*.

Beachten Sie, dass diese Funktion nur für Förderobjekte aufgerufen werden darf, die einer Förderstrecke zugeordnet sind.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.

_MT.SetPosition

Die Funktion *_MT.SetPosition* setzt die Position eines Förderguts relativ zur Position der Förderstreckenkomponente, der dieses Fördergut zugeordnet ist. Die Position wird in Millimetern bezogen auf die linke obere Ecke der Komponente (in ihrer Ausgangslage) angegeben. Der Rückgabewert ist vom Typ *void*.

Beachten Sie, dass diese Funktion nur für Förderobjekte aufgerufen werden darf, die einer Förderstrecke zugeordnet sind, sich aber nicht unter der Kontrolle des Lösungsverfahrens befinden.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu verändernden Förderguts.
- x (analog)
  X-Position in Millimetern relativ zur Förderstreckenkomponente.
- y (analog)
  Y-Position in Millimetern relativ zur Förderstreckenkomponente.

_MT.SetAngle

Die Funktion *_MT.SetAngle* setzt den Winkel eines Förderguts relativ zur Tangente des Streckenabschnitts, auf dem er sich befindet in Grad (−180° .. +180°). Der Rückgabewert ist vom Typ *void*.

Beachten Sie, dass diese Funktion nur für Förderobjekte aufgerufen werden darf, die einer Förderstrecke zugeordnet sind.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu verändernden Förderguts.
- angle (integer)
  Winkel in Grad relativ zur Tangente des Streckenabschnittes.

_MT.AddAngle

Die Funktion *_MT.AddAngle* erhöht den Winkel eines Förderguts relativ zur Tangente des Streckenabschnitts, auf dem er sich befindet in Grad (−180° .. +180°). Der Rückgabewert ist vom Typ *void*.

Beachten Sie, dass diese Funktion nur für Förderobjekte aufgerufen werden darf, die einer Förderstrecke zugeordnet sind.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu verändernden Förderguts.
- angle (integer)
  Winkel in Grad relativ zur Tangente des Streckenabschnittes.

_MT.SetHoldup

Die Funktion *_MT.SetHoldup* überträgt an ein Fördergut die Information, ob es sich vorne bzw. hinten unmittelbar an einem Blocker oder einem anderen Fördergut befindet. Der Rückgabewert ist vom Typ *void*.

Beachten Sie, dass diese Funktion nur für Förderobjekte aufgerufen werden darf, die einer Förderstrecke zugeordnet sind, sich aber nicht unter der Kontrolle des Lösungsverfahrens befinden.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu verändernden Förderguts.
- front (binary)
  Angabe, ob das Fördergut vorne blockiert ist.
- rear (binary)
  Angabe, ob das Fördergut hinten blockiert ist.

_MT.GetBinaryValue

Die Funktion *_MT.GetBinaryValue* ermöglicht die Abfrage einer binären Größe (Zustand, Eingang, Ausgang) eines Fördergutes. Der Rückgabewert ist vom Typ *bool*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.
- property (string)
  Der Name der abzufragenden Größe

_MT.GetAnalogValue

Die Funktion *_MT.GetAnalogValue* ermöglicht die Abfrage einer analogen Größe (diskreter Zustand, Eingang, Ausgang) eines Fördergutes. Der Rückgabewert ist vom Typ *double*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.
- property (string)
  Der Name der abzufragenden Größe.

_MT.GetIntegerValue

Die Funktion *_MT.GetIntegerValue* ermöglicht die Abfrage einer ganzzahligen Größe (Zustand, Eingang, Ausgang) eines Fördergutes. Der Rückgabewert ist vom Typ *long*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.
- property (string)
  Der Name der abzufragenden Größe.

_MT.GetByteValue

Die Funktion *_MT.GetByteValue* ermöglicht die Abfrage eines Bytes (Zustand) eines Fördergutes. Der Rückgabewert ist vom Typ *byte*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.
- property (string)
  Der Name des abzufragenden Zustands.

_MT.GetByteArray

Die Funktion *_MT.GetByteArray* ermöglicht die Abfrage eines Byte-Arrays (Zustand) eines Fördergutes. Der Rückgabewert ist vom Typ *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des abzufragenden Förderguts.
- property (string)
  Der Name des auszulesenden Zustandsvektors.
- stateoffset (long)
  Der Offset in Bytes, ab dem der Zustandsvektor ausgelesen werden soll.
- buffer (byte[])
  Byte-Array, in das die gelesenen Bytes eingetragen werden sollen. Falls Sie einen Zustandsvektor der Komponente angeben, übergeben Sie ihn hier in der Schreibweise für die neuen Zustandswerte, also mit vorangestelltem "@".
- bufferoffset (long)
  Der Offset in Bytes, ab dem die gelesenen Bytes eingetragen werden sollen.
- count (long)
  Anzahl Byte, die gelesen werden sollen.

Hinweis

Alle Arrays müssen groß genug sein, um diese Operation auszuführen, da ansonsten unvorhersehbare Fehler im Simulationsablauf auftreten können.

_MT.SetBinaryValue

Die Funktion *_MT.SetBinaryValue* ermöglicht das Schreiben einer binären Größe (Zustand, Eingang) eines Fördergutes. Der Rückgabewert ist vom Typ *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu beschreibenden Förderguts.
- property (string)
  Der Name des zu schreibenden Zustandes.
- value (bool)
  Der zu schreibende Wert.

_MT.SetAnalogValue

Die Funktion *_MT.SetAnalogValue* ermöglicht das Schreiben einer analogen Größe (diskreter Zustand, Eingang) eines Fördergutes. Der Rückgabewert ist vom Typ *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu beschreibenden Förderguts.
- property (string)
  Der Name der zu schreibenden Größe.
- value (double)
  Der zu schreibende Wert.

_MT.SetIntegerValue

Die Funktion *_MT.SetIntegerValue* ermöglicht das Schreiben einer ganzzahligen Größe (Zustand, Eingang) eines Fördergutes. Der Rückgabewert ist vom Typ *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu beschreibenden Förderguts.
- property (string)
  Der Name der zu schreibenden Größe.
- value (long)
  Der zu schreibende Wert.

_MT.SetByteValue

Die Funktion *_MT.SetByteValue* ermöglicht das Schreiben eines Bytes (Zustand) eines Fördergutes. Der Rückgabewert ist vom Typ *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu beschreibenden Förderguts.
- property (string)
  Der Name des zu schreibenden Zustands.
- value (byte)
  Der zu schreibende Wert.

_MT.SetByteArray

Die Funktion *_MT.SetByteArray* ermöglicht das Schreiben eines Byte-Arrays (Zustand) eines Fördergutes. Der Rückgabewert ist vom Typ *void*.

Geben Sie folgende Übergabeparameter an:

- error (long[])
  Fehler-Code
  Informationen zum Fehler-Code finden Sie in der Tabelle im Abschnitt: Systemfunktionen (Seite 938).
- id (long)
  Die ID des zu beschreibenden Förderguts.
- property (string)
  Der Name des zu beschreibenden Zustandsvektors des Fördergutes.
- stateoffset (long)
  Der Offset in Bytes, ab dem der Zustandsvektor beschrieben werden soll.
- buffer (byte[])
  Byte-Array, das die zu schreibenden Bytes enthält.
- bufferoffset (long)
  Der Offset in Bytes, ab dem das Byte-Array übertragen werden soll.
- count (long)
  Anzahl Bytes, die geschrieben werden sollen.

Hinweis

Alle Arrays müssen groß genug sein, um diese Operation auszuführen, da ansonsten unvorhersehbare Fehler im Simulationsablauf auftreten können.

8.3.4.4 Systemvariablen

Für die Modellierung von Fördertechnikkomponenten stehen die in folgender Tabelle gelisteten Systemvariablen zur Verfügung, auf die in der Verhaltensbeschreibung der Komponente ausschließlich lesend zugegriffen werden darf.

Tabelle 8-72 Systemvariablen

Variablenname	Bedeutung
_WIDTH	Breite der Komponente in Pixel
_HEIGHT	Höhe der Komponente in Pixel
_SCALEX	Horizontale Skalierung der Komponente. Faktor eins entspricht der Default-größe.
_SCALEY	Vertikale Skalierung der Komponente. Faktor eins entspricht der Defaultgröße.
_TECHSCALE	Maßstab des Planes, auf dem die Komponente liegt. Es werden die Anzahl Millimeter angegeben, die einem Pixel entsprechen.

Menüs und Dialogfelder

9

9.1 Menüs

9.1.1 Portalansicht > Start

Die Schaltfläche "Start" öffnet den Zugriff auf folgende Funktionen:

- **Bestehendes Projekt öffnen**
  Hier öffnen Sie ein bereits bestehendes Projekt. Weitere Informationen hierzu finden Sie im Abschnitt: Projekt > Öffnen... (Seite 953)
- **Neues Projekt erstellen**
  Hier erstellen Sie ein neues Projekt. Weitere Informationen hierzu finden Sie im Abschnitt: Projekt > Neues Projekt... (Seite 952)
- **Projekt dearchivieren**
  Hier dearchivieren Sie ein Projekt. Weitere Informationen hierzu finden Sie im Abschnitt: Projekt > Dearchivieren (Seite 955)
- **Beispielprojekt dearchivieren**
  Hier dearchivieren Sie eines der mitgelieferten Beispielprojekte. Wählen Sie das gewünschte Beispielprojekt aus der Klappliste aus. Der Speicherort für das dearchivierte Beispielprojekt ist frei wählbar.
- **Projekt schließen**
  Das aktuelle Projekt wird geschlossen.
- **Erste Schritte**
  Hier werden Ihnen die ersten Schritte mit SIMIT erklärt.
- **Installierte Software**
  Hier wird Ihnen die aktuell installierte SIMIT-Software angezeigt. Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Info" (Seite 978)
- **Hilfe**
  Hier rufen Sie die Online-Hilfe zu SIMIT auf.
- **Größenvariante**
  Hier ändern Sie die Größenvariante, für die Sie eine Lizenz erworben haben. Die Änderung wird erst nach einem Neustart von SIMIT wirksam.
- **Oberflächensprache**
  Hier ändern Sie die Oberflächensprache von SIMIT. Die Änderung wird erst nach einem Neustart von SIMIT wirksam.

9.1.2 Portalansicht > Kopplungen

Die Schaltfläche "Kopplungen" öffnet den Zugriff auf folgende Funktionen:

- **Neue Kopplung hinzufügen**
  Wählen Sie eine Kommunikationsverbindung, über die SIMIT Daten austauschen soll.
- **Kopplungssignale zuweisen**
  Wenn sich die Signale der Konnektoren eindeutig einer Kopplung zuordnen lassen, können sie mit dieser Funktion aktualisiert werden. Das ist z. B. bei Namensänderungen nötig.
- **SU-Verwaltung**
  Wenn als Kopplung "SIMIT Unit" gewählt ist, tragen Sie hier einen Namen und die IP-Adresse der SIMIT Unit ein.
- **Hilfe**
  Hier rufen Sie die Online-Hilfe zu SIMIT auf.

Siehe auch

Extras > Kopplungssignale zuweisen (Seite 959)

9.1.3 Portalansicht > Simulationsmodell

Die Schaltfläche "Simulationsmodell" öffnet den Zugriff auf folgende Funktionen:

- **Neues Diagramm hinzufügen**
  Fügen Sie ein neues Diagramm in Ihr Projekt ein. SIMIT wechselt zur Projektansicht in den Diagrammeditor.
- **Neues Makro erstellen**
  Erstellen Sie ein neues Makro. SIMIT wechselt zur Projektansicht in den Makroeditor.
- **Neuen Komponententyp erstellen**
  Erstellen Sie einen eigenen Komponententyp. SIMIT wechselt zum Component Type Editor.
- **Hilfe**
  Hier rufen Sie die Online-Hilfe zu SIMIT auf.

9.1.4 Portalansicht > Automatische Modellerstellung

Die Schaltfläche "Automatische Modellerstellung" öffnet den Zugriff auf folgende Funktionen:

- **Vorlagen instanziieren**
  Erzeugen Sie Instanzen von Vorlagen aus Import-Dateien (*.xls, *.xlsx, *.txt, *.iea, *.xml) oder aus dem Simulationsmodell. Weitere Informationen hierzu finden Sie im Abschnitt: Instanziierung von Vorlagen aus Dateien oder dem Simulationsmodell (Seite 259).
- **Neue Vorlage erstellen**
  Erstellen Sie eine neue Vorlage. SIMIT wechselt zur Projektansicht in den Vorlageneditor. Weitere Informationen hierzu finden Sie im Kapitel: Vorlagen (Seite 252).

- **Automatisierter Import**
  Importiert Modelle aus folgenden Quellen:
  - XML (Seite 271)
  - ZIP (Seite 278)

  Weitere Informationen hierzu finden Sie im Abschnitt Automatisierter Import (Seite 270).
- **Massenbearbeitung Import**
  Übertragen Sie Parameter und Eingangsvorbelegungen aus einer Tabelle in bestehende Diagramme. Weitere Informationen hierzu finden Sie im Abschnitt: Massenbearbeitung (Seite 280).
- **Hilfe**
  Hier rufen Sie die Online-Hilfe zu SIMIT auf.

Siehe auch

Tabellenimport (Seite 261)

IEA-Import (Seite 264)

CMT-Import (Seite 266)

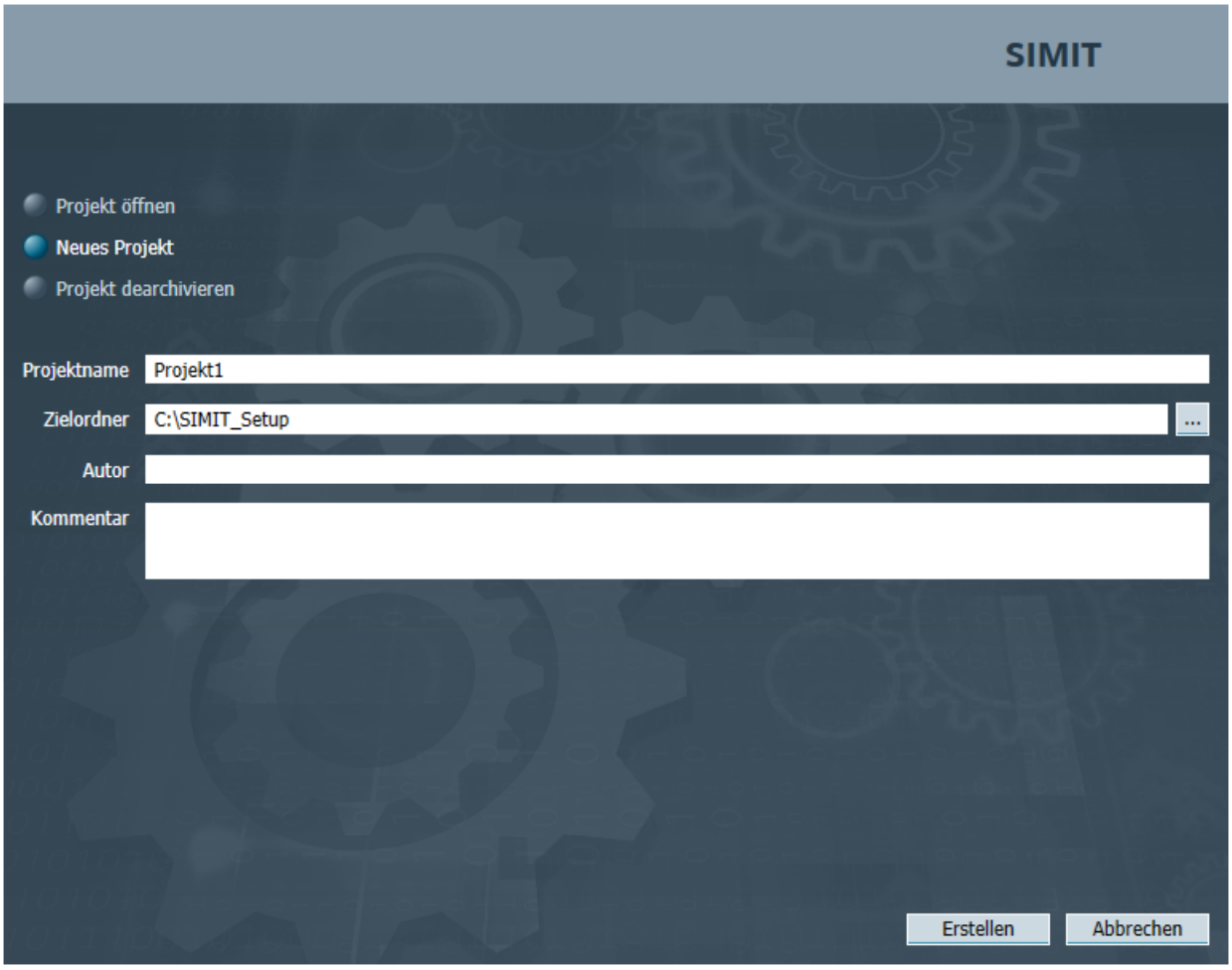
9.1.5 Portalansicht > Diagnose & Visualisierung

Die Schaltfläche "Diagnose & Visualisierung" öffnet den Zugriff auf folgende Funktionen:

- **Konsistenzprüfung**
  Überprüfen Sie Ihr Projekt auf formale Fehler. Das Ergebnis der Konsistenzprüfung wird in der Projektansicht angezeigt. Weitere Informationen hierzu finden Sie im Abschnitt: Konsistenzprüfung (Seite 304).
- **Suchen & Ersetzen**
  Führen Sie die Funktion "Suchen & Ersetzen" im aktuellen Projekt aus. Weitere Informationen hierzu finden Sie im Kapitel: Suchen & Ersetzen (Seite 299).
- **Neues Kurvenbild hinzufügen**
  Erstellen Sie ein neues Kurvenbild. SIMIT wechselt in die Projektansicht in den Kurvenbildeditor. Weitere Informationen hierzu finden Sie im Kapitel: Kurvenbilder (Seite 290).
- **Archiv bearbeiten**
  Erstellen Sie ein Archiv. Aus einem Archiv kann ein Kurvenbild erstellt werden. SIMIT wechselt in die Projektansicht in den Archiveditor. Weitere Informationen hierzu finden Sie im Kapitel: Archiv (Seite 288).
- **Hilfe**
  Rufen Sie die Online-Hilfe zu SIMIT auf.

9.1.6 Projekt > Neues Projekt...

Mit diesem Menübefehl öffnet sich das folgende Dialogfeld:



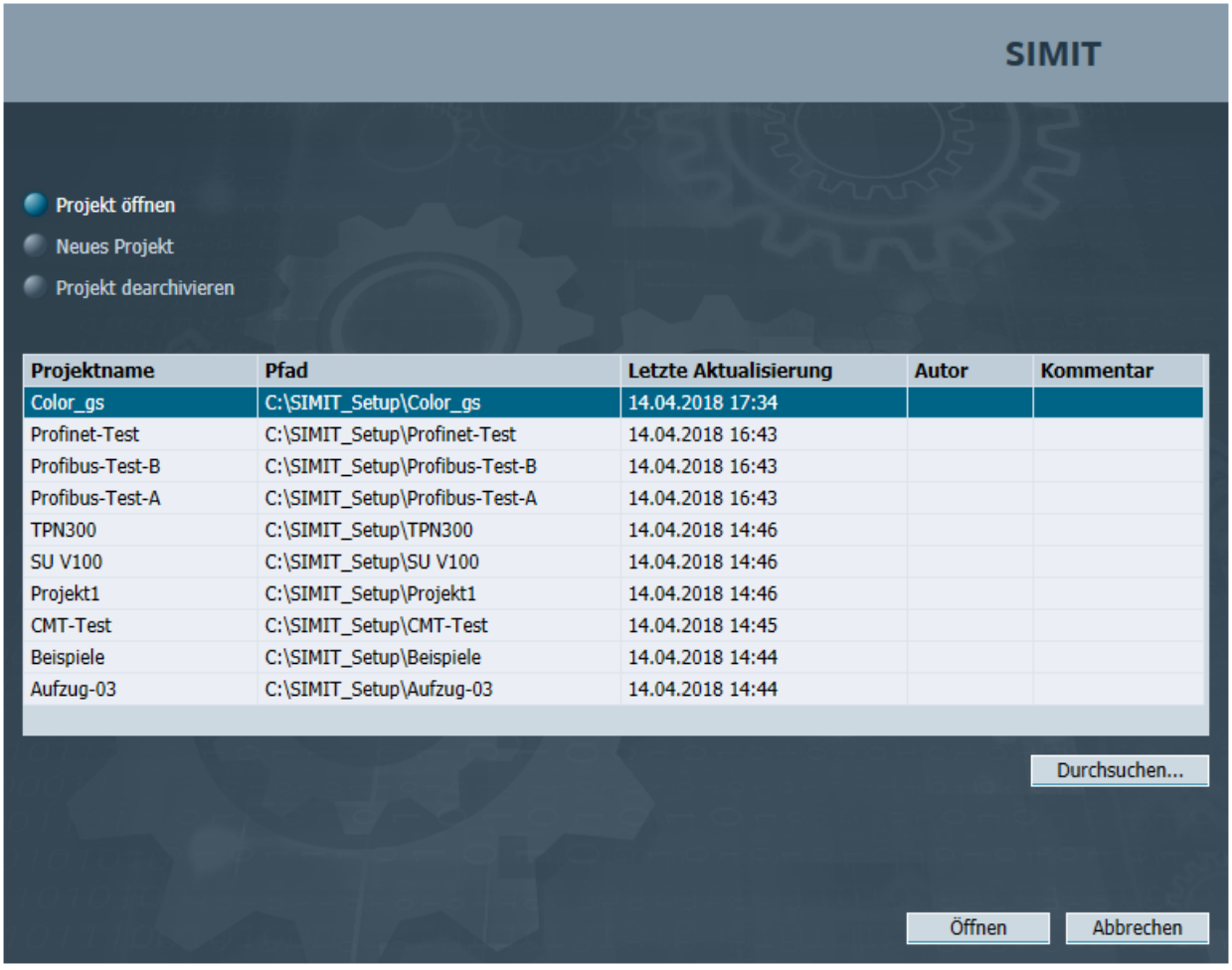
Hier können Sie ein neues Projekt erstellen. In den Eingabefeldern tragen Sie die dafür notwendigen Daten ein und nehmen Einstellungen vor.

- **Projektname**
  Tragen Sie einen beliebigen Namen für Ihr Projekt ein oder übernehmen Sie den voreingestellten Namen.
- **Zielordner**
  Der Speicherort des neuen Projekts ist bereits voreingestellt. Um einen anderen Speicherort auszuwählen, klicken Sie auf die Schaltfläche "...".
- **Autor**
  Tragen Sie hier den Namen des Autors ein. Dieser Eintrag ist optional.
- **Kommentar**
  Tragen Sie hier einen beliebigen Kommentar zum Projekt ein. Dieser Eintrag ist optional.

Klicken Sie auf die Schaltfläche "Erstellen" um das neue Projekt zu erstellen und das Dialogfeld zu schließen.

9.1.7 Projekt > Öffnen...

Mit diesem Menübefehl öffnet sich das folgende Dialogfeld:



Zuletzt geöffnete Projekte sowie das Beispielprojekt werden hier aufgelistet. Zu jedem Projekt werden Projektname, Speicherort und das Datum der letzten Aktualisierung des Projekts mit angezeigt. Diese Daten können hier nicht verändert werden.

Wird das gewünschte Projekt hier nicht angezeigt, suchen Sie es über die Schaltfläche "Durchsuchen...".

Öffnen Sie das Projekt, indem Sie es markieren und auf die Schaltfläche "Öffnen" klicken.

9.1.8 Projekt > Schließen

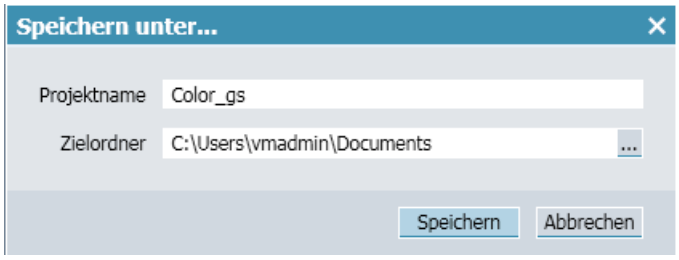
Mit diesem Menübefehl wird das aktuelle Projekt geschlossen.

9.1.9 Projekt > Alles speichern

Mit diesem Menübefehl wird das aktuelle Projekt gespeichert.

9.1.10 Projekt > Speichern unter...

Mit diesem Menübefehl öffnet sich das folgende Dialogfeld:



Projektname und Zielordner sind voreingestellt.

- **Projektname**
  Übernehmen Sie den voreingestellten Namen oder tragen Sie hier einen beliebigen Namen für Ihr Projekt ein.
- **Zielordner**
  Übernehmen Sie den voreingestellten Speicherort für Ihr Projekt, oder wählen Sie einen anderen indem Sie auf die Schaltfläche "..." klicken.

Klicken Sie auf die Schaltfläche "Speichern", um das Projekt zu speichern und das Dialogfeld zu schließen.

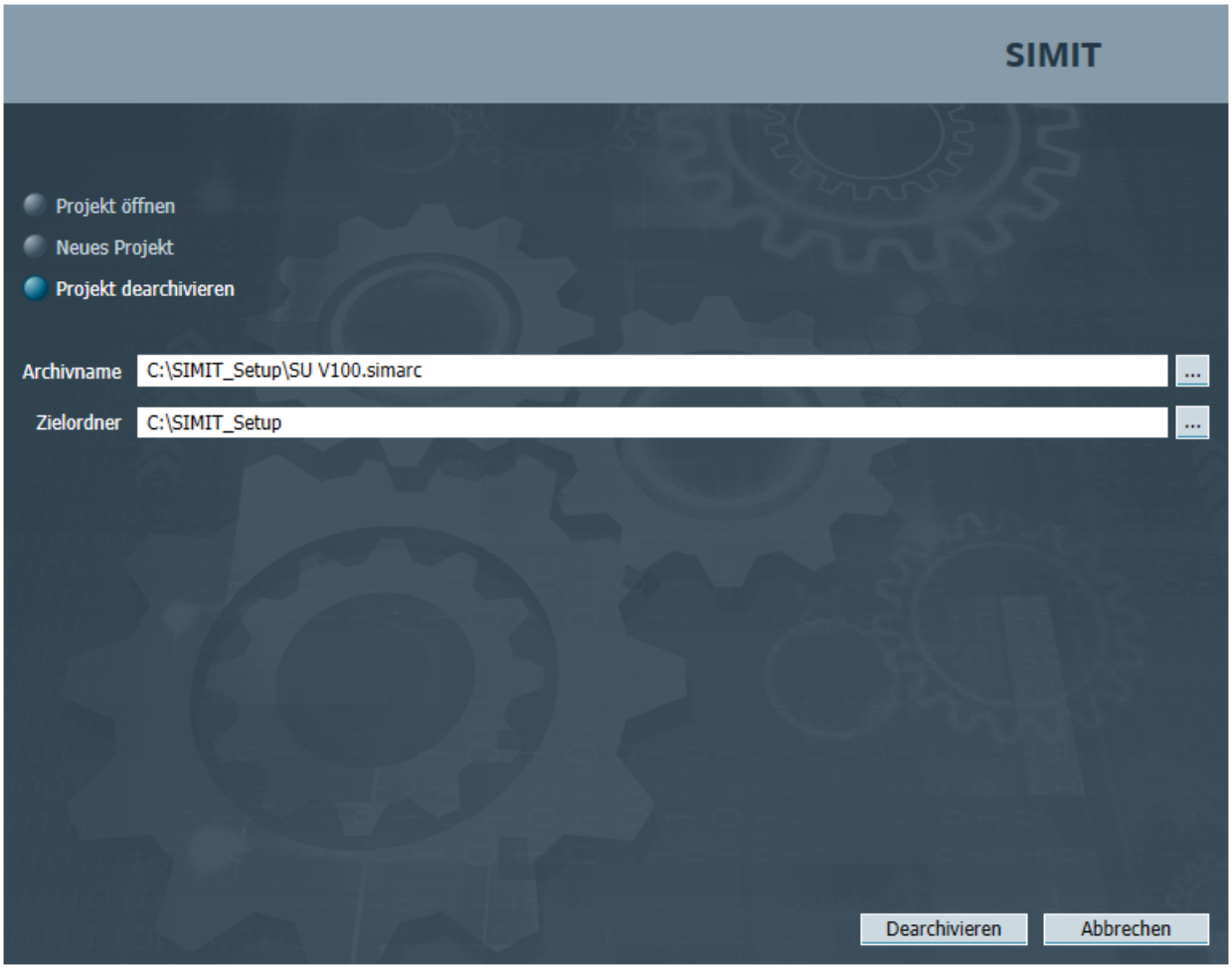
9.1.11 Projekt > Archivieren

Mit diesem Menübefehl öffnet sich das Dialogfeld "Speichern unter", wo Sie einen Namen und einen Speicherort für das aktuelle Projekt eingeben. Klicken Sie anschließend auf die Schaltfläche "Speichern", um die Archivierung des Projekts durchzuführen.

Das archivierte Projekt erhält das Dateikürzel "*.simarc".

9.1.12 Projekt > Dearchivieren

Mit diesem Menübefehl öffnet sich das folgende Dialogfeld:



Hier können Sie ein archiviertes Projekt wieder dearchivieren.

- **Archivname**
  Klicken Sie auf die Schaltfläche "..." um ein archiviertes SIMIT-Projekt auszuwählen. Archivierte SIMIT-Projekte haben die Dateiendung ".simarc".
- **Zielordner**
  Klicken Sie auf die Schaltfläche "..." um den Ordner auszuwählen, in den das Projekt nach dem Dearchivieren abgelegt werden soll.

Klicken Sie auf die Schaltfläche "Dearchivieren", um das ausgewählte Projekt zu dearchivieren.

Hinweis

Ein archiviertes SIMIT-Projekt kann ausführbaren Code enthalten. Dearchivieren Sie nur Projekte, die aus vertrauenswürdiger Quelle stammen.

9.1.13 Projekt > Analyse

Mit diesem Menübefehl öffnet sich ein Dialogfeld, in dem Sie eine statistische Auswertung aus den Daten eines SIMIT-Projekts speichern.

Die Datei wird im Format "*.xslx" gespeichert.

9.1.14 Projekt > Beenden

Mit diesem Menübefehl schließen Sie das aktuelle Projekt.

Wenn im Projekt Änderungen noch nicht gespeichert sind, öffnet sich ein Dialogfeld in dem abgefragt wird, ob Sie vor dem Beenden speichern möchten.

9.1.15 Bearbeiten > Ausschneiden

Mit diesem Menübefehl entfernen Sie das markierte Objekt.

9.1.16 Bearbeiten > Kopieren

Mit diesem Menübefehl kopieren Sie das markierte Objekt in die Zwischenablage.

9.1.17 Bearbeiten > Einfügen

Mit diesem Menübefehl fügen Sie das kopierte Objekt aus der Zwischenablage in den aktuellen Editor ein.

9.1.18 Simulation > Initialisieren

Mit diesem Menübefehl starten Sie die Konsistenzprüfung. Das Ergebnis der Konsistenzprüfung wird angezeigt.

Hinweis

Die Initialisierung startet nicht die Simulation.

9.1.19 Simulation > Starten

Mit diesem Menübefehl starten Sie die Simulation.

Weitere Informationen hierzu finden Sie im Kapitel: Aktionen bei laufender Simulation (Seite 39).

Bevor die Simulation gestartet wird, werden Sie gefragt, ob Sie die Änderungen im Projekt speichern möchten und die Konsistenzprüfung wird durchgeführt.

9.1.20 Simulation > Anhalten

Mit diesem Menübefehl wird die Simulation angehalten.

9.1.21 Simulation > Einzelschnitt

Mit diesem Menübefehl führen Sie einen Einzelschritt in der Simulation aus.

Weitere Informationen hierzu finden Sie im Kapitel: Ausführen eines Einzelschritts (Seite 318).

9.1.22 Simulation > Beenden

Mit diesem Menübefehl wird die laufende Simulation beendet und zur Projektansicht gewechselt.

9.1.23 Simulation > Schnappschuss

Mit diesem Menübefehl wird ein Schnappschuss der Simulation angelegt.

Weitere Informationen hierzu finden Sie im Kapitel: Aktionen bei laufender Simulation (Seite 39).

9.1.24 Fenster > Horizontal teilen

Mit diesem Menübefehl teilen Sie das markierte Editorfenster horizontal auf.

9.1.25 Fenster > Vertikal teilen

Mit diesem Menübefehl teilen Sie das markierte Editorfenster vertikal auf.

9.1.26 Fenster > Teilung aufheben

Mit diesem Menübefehl wird eine horizontale oder vertikale Fensterteilung wieder aufgehoben.

9.1.27 Fenster > Alle schließen

Mit diesem Menübefehl werden alle aktuell geöffneten Fenster geschlossen.

9.1.28 Automatische Modellerstellung > Vorlagen instanziieren

Öffnet das Dialogfeld "Instanziiere Vorlagen".

Weitere Informationen hierzu finden Sie im Abschnitt Instanziierung von Vorlagen aus Dateien oder dem Simulationsmodell (Seite 259).

9.1.29 Automatische Modellerstellung > Automatisierter Import

Öffnet den Dateiauswahldialog zum Import von Modellen aus folgenden Quellen:

- XML (Seite 271)
- ZIP (Seite 278)

Weitere Informationen hierzu finden Sie im Abschnitt: Diagramme aus ZIP- oder XML-Dateien importieren (Seite 270).

9.1.30 Automatische Modellerstellung > Massenbearbeitung Import

Mit diesem Menübefehl öffnet sich das Dialogfeld "Massenbearbeitung Import".

Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Massenbearbeitung Import" (Seite 976).

9.1.31 Extras > Zoom

Mit diesem Menübefehl können Sie das markierte Objekt zoomen. Verwenden Sie hierzu den Schieberegler.

9.1.32 Extras > SU-Verwaltung

Mit diesem Menübefehl öffnet sich das Dialogfeld "SU-Verwaltung":

Weitere Informationen hierzu finden Sie unter:

- Dialogfeld "SU-Verwaltung" (Seite 973)
- SIMIT Unit konfigurieren (Seite 87)
- Gerätebeschreibungsdatei in SIMIT importieren (Seite 88)
- Firmware der SIMIT Unit aktualisieren (Seite 88)

9.1.33 Extras > Kopplungssignale zuweisen

Mit diesem Menübefehl werden die Einträge in Input- bzw. Output-Konnektoren auf den Diagrammen des gesamten Projekts aktualisiert, wenn sich das im Konnektor eingetragene Signal eindeutig einer Kopplung zuordnen lässt.

Anwendungsbeispiele:

- eine Kopplung wurde nachträglich umbenannt
- Kopplungssignale haben nachträglich symbolische Namen erhalten

Siehe auch

Signal symbolisch adressieren (Seite 214)

9.1.34 Hilfe > Hilfe anzeigen

Mit diesem Menübefehl rufen Sie die Online-Hilfe zu SIMIT auf.

9.1.35 Hilfe > Info

Mit diesem Menübefehl werden Informationen über die SIMIT-Installation angezeigt.

Weitere Informationen hierzu finden Sie im Abschnitt: Dialogfeld "Info" (Seite 978).

9.1.36 CTE > Komponente > Neue Komponente

Mit diesem Menübefehl erstellen Sie eine neue Komponente.

9.1.37 CTE > Komponente > Öffnen

Mit diesem Menübefehl öffnen Sie eine bereits vorhandene Komponente.

9.1.38 CTE > Komponente > Schließen

Mit diesem Menübefehl schließen Sie die geöffnete Komponente.

9.1.39 CTE > Komponente > Speichern

Mit diesem Menübefehl speichern Sie die aktuelle Komponente.

9.1.40 CTE > Komponente > Speichern unter

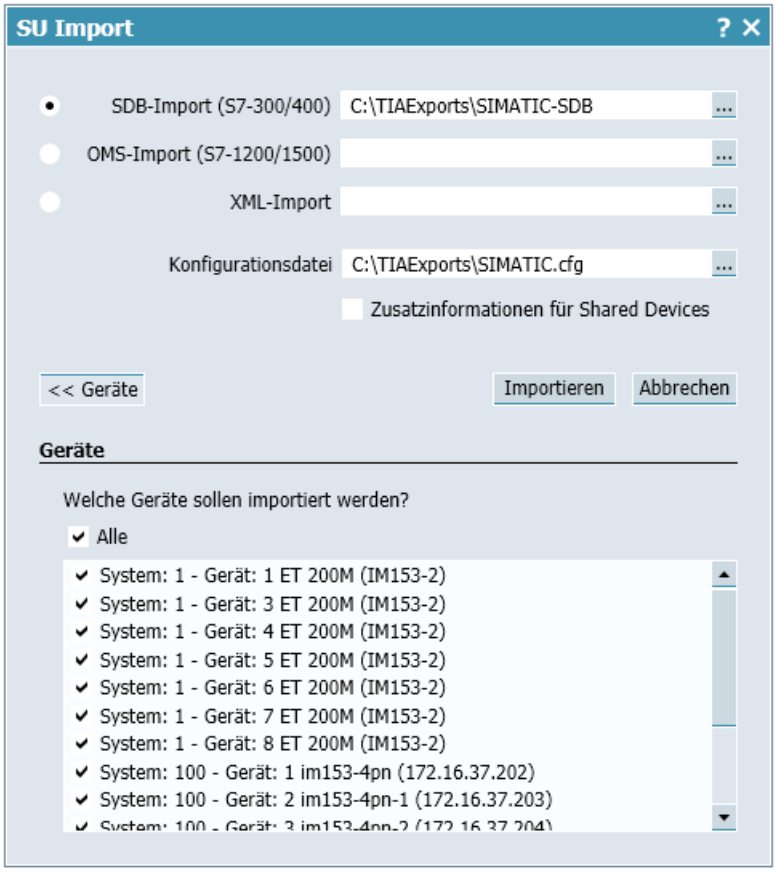
Mit diesem Menübefehl wählen Sie einen verfügbaren Speicherort aus und speichern dort die aktuelle Komponente.

9.1.41 CTE > Komponente > Beenden

Mit diesem Menübefehl beenden Sie den Component Type Editor.

9.2 Dialogfelder

9.2.1 Dialogfeld "SU Import"



Mit diesem Dialogfeld nehmen Sie folgende Einstellungen zum Import der Hardwarekonfiguration vor:

- "SDB-Import"
  Legt den Ordner fest, in dem die Systemdatenbausteine für den Import der Hardware-Konfiguration einer S7-300/S7-400-Station liegen.
- "OMS-Import"
  Legt den Ordner fest, in dem die Systemdatenbausteine für den Import der Hardware-Konfiguration einer S7-1200/S7-1500-Station liegen.
  Der OMS-Import wird ab STEP 7 V13 unterstützt.
- "XML-Import"
  Legt die Datei fest, aus der eine herstellerneutrale Hardware-Konfiguration mit PROFIBUS- und/oder PROFINET-Mastersystemen importiert wird.
  Informationen zum Aufbau dieser Datei finden Sie unter "XML-Importschnittstelle (Seite 103)".

- "Konfigurationsdatei"
  Legt den Ablageort der Konfigurationsdatei mit zusätzlichen Geräteinformationen fest. Nur für den "SDB-Import" relevant.
- "Zusatzinformationen für Shared Devices"
  Legt fest, die zu importierenden Systemdatenbausteine der bestehenden Hardware-Konfiguration einer Station zu vervollständigen. Nur bei Simulation eines Shared Device relevant.
- "Geräte"
  Blendet den Bereich zur Stationsauswahl aus den gewählten Systemdaten ein. Standardmäßig sind alle Stationen ausgewählt.
- "Importieren"
  Startet den Import der Stationen in das SIMIT-Projekt.
- "Abbrechen"
  Bricht den Import ab. In der Projektnavigation wird nur der Ordner "SIMIT Unit" angelegt.

9.2.2 Dialogfeld "PLCSIM Advanced-Import"

Mit diesem Dialogfeld nehmen Sie folgende Einstellungen zum Import der Hardwarekonfiguration eines STEP 7-Projekts vor:

- TIA Portal Projekt
  Legt das Projekt fest, aus dem die Hardwarekonfiguration importiert wird. Diese Optionen sind nur verfügbar, wenn auf dem PC STEP 7 V14 SP1 installiert ist.
- HWCNExport-Datei
  Legt die HWCNExport-Datei fest, aus der die Hardwarekonfiguration importiert wird.
- Bus-synchron
  Legt fest, dass die PLCSIM Advanced-Kopplung bus-synchron betrieben wird.

- Symbolik
  Legt fest, wie Symbolnamen für Ein- und Ausgänge behandelt werden. Die Auswahl ist nur wirksam, wenn im STEP 7-Projekt Symbolnamen vergeben wurden.

Symbol-Modus	Stationen der PLCSIM Advanced-Kopplung werden erneut importiert (aktualisiert)	Stationen der PLCSIM Advanced-Kopplung werden neu angelegt
Neu anlegen	Alle Symbole der zu importierenden Stationen werden gelöscht und aus dem Import neu übernommen.	Alle Symbole der zu importierenden Stationen werden aus dem Import übernommen.
Überschreiben	Alle Symbole der zu importierenden Stationen werden aus dem Import übernommen. Bereits vorhandene Symbole werden überschrieben. Neue Symbole werden übernommen.	
Hinzufügen	Es werden nur die Symbole der zu importierenden Stationen aus dem Import übernommen, für die bisher keine Symbole vorhanden sind.	
Ignorieren	Die Symbole der zu importierenden Stationen werden nicht übernommen. Bereits vorhandene Symbole bleiben unverändert.	Die Symbole der zu importierenden Stationen werden nicht übernommen.

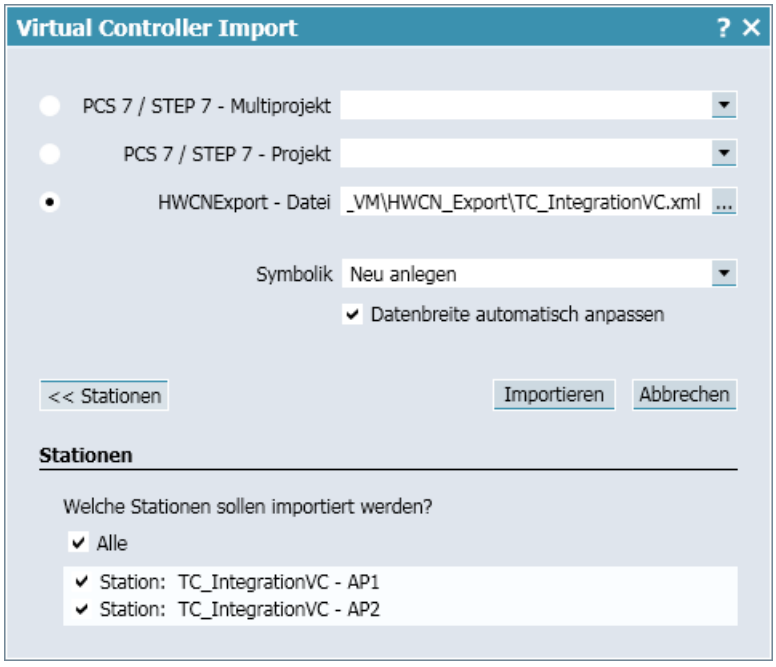
- Datenbreite automatisch anpassen
  Legt fest, dass die Datenbreite der importierten Signale an die Symbolik angepasst wird. Ohne diese Option werden nur Symbole übernommen, die mit dem Default-Datentyp Word verknüpft sind.
- Stationen **>>**
  Blendet den Bereich zur Stationsauswahl aus dem gewählten Projekt ein. Die Stationen werden aus dem ausgewählten Projekt eingelesen. Angezeigt werden nur S7-1500-Stationen, die von PLCSIM Advanced unterstützt werden.
  Standardmäßig sind alle unterstützten Stationen des STEP 7-Projekts ausgewählt.
- Importieren
  Startet den Import der Stationen in das SIMIT-Projekt.
- Abbrechen
  Bricht den Import ab. In der Projektnavigation wird nur der Ordner "PLCSIM Advanced" angelegt.

Siehe auch

Kopplung vom Typ "PLCSIM Advanced" anlegen (Seite 145)

Hardwarekonfiguration in Station importieren (Seite 146)

9.2.3 Dialogfeld "Virtual Controller Import"



Mit diesem Dialogfeld nehmen Sie folgende Einstellungen zum Import der Hardwarekonfiguration eines STEP 7-/PCS 7-Projekts vor:

- STEP 7-/PCS 7-Multiprojekt / STEP 7-/PCS 7-Projekt
  Legt das Projekt fest, aus dem die Hardwarekonfiguration importiert wird. Diese Optionen sind nur verfügbar, wenn der SIMATIC Manager installiert ist.
- HWCNExport-Datei
  Legt die HWCNExport-Datei fest, aus der die Hardwarekonfiguration importiert wird.

- Symbolik
  Legt fest, wie Symbolnamen für Ein- und Ausgänge behandelt werden. Die Auswahl ist nur wirksam, wenn im STEP 7-/PCS 7-Projekt Symbolnamen vergeben wurden.

Symbol-Modus	Virtual Controller wird erneut importiert (aktualisiert)	Virtual Controller wird neu angelegt
Neu anlegen	Alle Symbole der zu importierenden Virtual Controller werden gelöscht und aus dem Import neu übernommen.	Alle Symbole der zu importierenden Virtual Controller werden aus dem Import übernommen.
Überschreiben	Alle Symbole der zu importierenden Virtual Controller werden aus dem Import übernommen. Bereits vorhandene Symbole werden überschrieben. Neue Symbole werden übernommen.	
Hinzufügen	Es werden nur die Symbole der zu importierenden Virtual Controller aus dem Import übernommen, für die bisher keine Symbole vorhanden sind.	
Ignorieren	Die Symbole der zu importierenden Virtual Controller werden nicht übernommen. Bereits vorhandene Symbole bleiben unverändert.	Die Symbole der zu importierenden Virtual Controller werden nicht übernommen.

- Datenbreite automatisch anpassen
  Legt fest, dass die Datenbreite der importierten Signale an die Symbolik angepasst wird. Ohne diese Option werden nur Symbole übernommen, die mit dem Default-Datentyp Word verknüpft sind.
- Stationen >>
  Blendet den Bereich zur Stationsauswahl aus dem gewählten Projekt ein. Die Stationen werden aus dem ausgewählten Projekt eingelesen. Die Dauer dieses Vorgangs ist abhängig von der Größe des STEP 7-/PCS 7-Projekts. Ein Abbruch ist nicht möglich. Standardmäßig sind alle Stationen eines STEP 7-/PCS 7-Projekts ausgewählt.
- Importieren
  Startet den Import der Stationen in das SIMIT-Projekt.
- Abbrechen
  Bricht den Import ab. In der Projektnavigation wird nur der Ordner "Virtual Controller" angelegt.

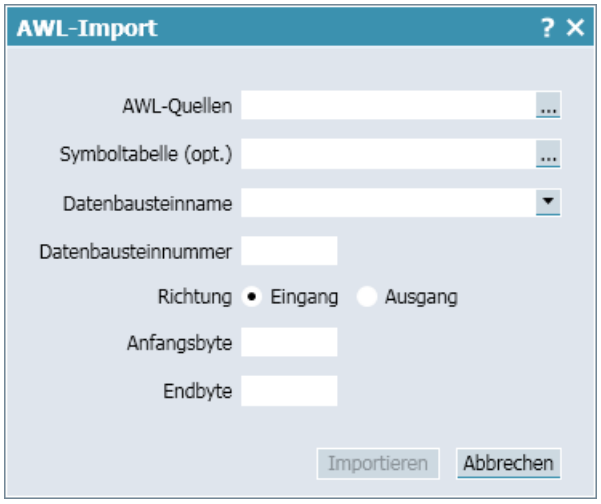
Siehe auch

HWCNExport-Datei erstellen (Seite 65)

STEP 7-/PCS 7-Projekt importieren (Seite 131)

Virtual Controller anlegen (Seite 130)

9.2.4 Dialogfeld "AWL-Import"



Öffnen Sie das Dialogfeld, indem Sie im Kopplungseditor der Virtual Controller-Kopplung in der Registerkarte "DB" auf die Schaltfläche "AWL-Import" klicken.

In diesem Dialogfeld nehmen Sie folgende Einstellungen vor:

- "AWL-Quellen"
  Legt die AWL-Quelle fest.
- "Symboltabelle (opt.)"
  Legt die Symboltabelle fest.
  Wenn die Datenbausteine in der AWL-Quelle symbolische Namen besitzen, können Sie die Datenbausteinnummer durch Angeben einer passenden Symboltabelle ermitteln lassen. Aus der Symboltabelle werden die Nummern der Datenbausteine eingelesen und unter "Datenbausteinnummer" im Dialogfeld angezeigt.
- "Datenbausteinname"
  Legt beim Import aus einer Symboltabelle den Datenbaustein fest, aus dem ein Speicherbereich importiert wird.
- "Datenbausteinnummer"
  Legt die Datenbausteinnummer fest, aus dem ein Speicherbereich importiert wird.
  Eine manuell eingetragene Datenbausteinnummer wird nicht validiert. Wenn Sie aus einer Symboltabelle importieren, ist die Datenbausteinnummer bereits eingetragen.
- "Richtung"
  Legt fest, ob Eingangs- oder Ausgangssignale importiert werden.
- "Anfangsbyte"
  Legt das Anfangsbyte fest, ab dem Signale importiert werden.

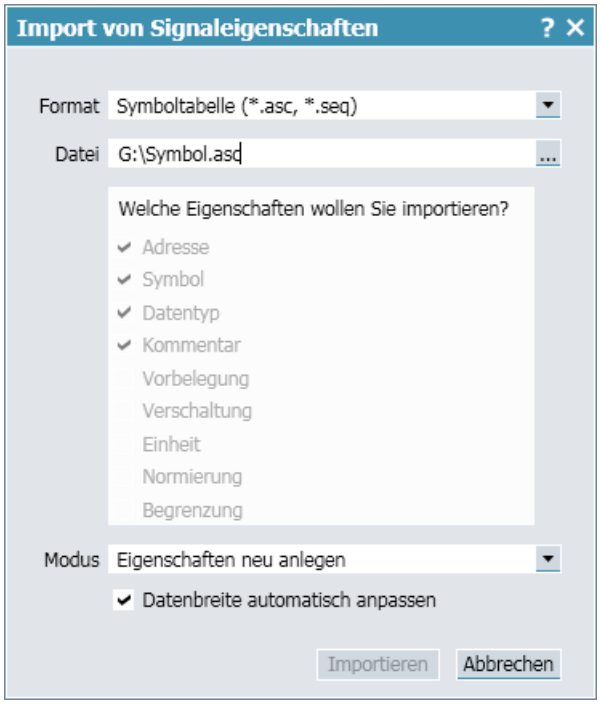
- "Endbyte"
  Legt das Endbyte fest, bis zu dem Signale importiert werden. Tragen Sie hier maximal das Ende des Datenbausteins ein.
- "Importieren"
  Startet den Import. Folgende Felder müssen gefüllt sein:
  - AWL-Quelle
  - Datenbausteinnummer
  - Anfangsbyte
  - Endbyte

  Wenn im angegebenen Speicherbereich von "Anfangsbyte" und "Endbyte" bereits Signale vorhanden sind, werden diese Signale überschrieben.

Siehe auch

DB-Daten aus AWL-Quellen importieren (Seite 143)

9.2.5 Dialogfeld "Import von Signaleigenschaften"



Mit diesem Dialogfeld nehmen Sie folgende Einstellungen zum Import der Signaleigenschaften vor:

- "Format"
  Welches Format eingelesen werden kann, ist vom aktuellen Kopplungstyp abhängig. Folgende Dateiformate werden eingelesen:
  - SIMATIC-Symboltabelle aus STEP 7 bzw. PCS 7
    Weitere Informationen hierzu finden Sie im Abschnitt: Symboltabelle (Seite 200).
  - PLC-Variablentabelle aus dem TIA-Portal
    Weitere Informationen hierzu finden Sie im Abschnitt: Variablentabelle (Seite 201).
  - Signaltabelle (txt-Datei)
    Weitere Informationen hierzu finden Sie im Abschnitt: Signaltabelle (Seite 202).

    Hinweis

    OPC-Kopplung

    Wurden Signaleigenschaften im txt-Format abgespeichert, enthält die Datei auch SIMATIC-Adressen und Angaben zur Normierung. Diese Informationen können von einer OPC-Kopplung aber nicht verarbeitet werden. Wenn eine txt-Datei in eine OPC-Kopplung importiert wird, werden deshalb nur diejenigen Informationen übernommen, die eine OPC-Kopplung auch verarbeiten kann.

  - INI-Datei
    Weitere Informationen hierzu finden Sie im Abschnitt: INI-Format der OPC-Kopplungen (Seite 203)
- "Datei"
  Legt den Speicherort der Import-Datei fest.
- "Bereich **"Welche Eigenschaften wollen Sie importieren?"**"
  Legt fest, welche Signaleigenschaften importiert werden. Importierbar sind nur in der Import-Datei enthaltene Signaleigenschaften. Wenn Sie eine Symboltabelle oder Variablentabelle importieren, werden folgende Signaleigenschaften standardmäßig importiert:
  - Adresse
  - Symbol
  - Datentyp
  - Kommentar

- "Modus"
  Der Modus legt fest, wie die Signale und deren Eigenschaften verarbeitet werden, die in der Kopplung bereits vorhanden sind. Welche Import-Modi auswählbar sind, ist vom aktuellen Kopplungstyp abhängig.

  Hinweis

  Die Import-Modi "Signale neu anlegen" und "Signale hinzufügen" werden von folgenden Kopplungstypen unterstützt:
  - SHM
  - OPC
  - PLCSIM
  - PRODAVE

  Die Import-Modi "Eigenschaften neu anlegen", "Eigenschaften überschreiben" und "Eigenschaften hinzufügen" werden von allen Kopplungstypen unterstützt.

  Es gibt folgende Import-Modi:

  – Signale neu anlegen
    Alle bereits vorhandenen Signale werden gelöscht und mit den importierten Informationen neu angelegt. Es werden nur die Eigenschaften importiert, die im Bereich "Welche Eigenschaften wollen Sie importieren?" aktiviert sind. Alle anderen Eigenschaften behalten ihre Voreinstellungen.

  – Signale hinzufügen
    Alle Signale aus der Import-Datei werden hinzugefügt. Es werden nur die Eigenschaften importiert, die im Bereich "Welche Eigenschaften wollen Sie importieren?" aktiviert sind. Alle anderen Eigenschaften behalten ihre Voreinstellungen. Bereits vorhandene Signale werden nicht verändert.

  – Eigenschaften neu anlegen
    Die im Bereich "Welche Eigenschaften wollen Sie importieren?" aktivierten Eigenschaften werden für alle in der Kopplung vorhandenen Signale auf ihre Voreinstellungen gesetzt und dann aus der Import-Datei übernommen.

  – Eigenschaften überschreiben
    Alle im Bereich "Welche Eigenschaften wollen Sie importieren?" aktivierten Eigenschaften werden aus der Import-Datei übernommen und auf die vorhandenen Signale übertragen.

  – Eigenschaften hinzufügen
    Alle im Bereich "Welche Eigenschaften wollen Sie importieren?" aktivierten Eigenschaften, die noch ihre Voreinstellungen besitzen, werden aus der Import-Datei übernommen und auf die vorhandenen Signale übertragen.

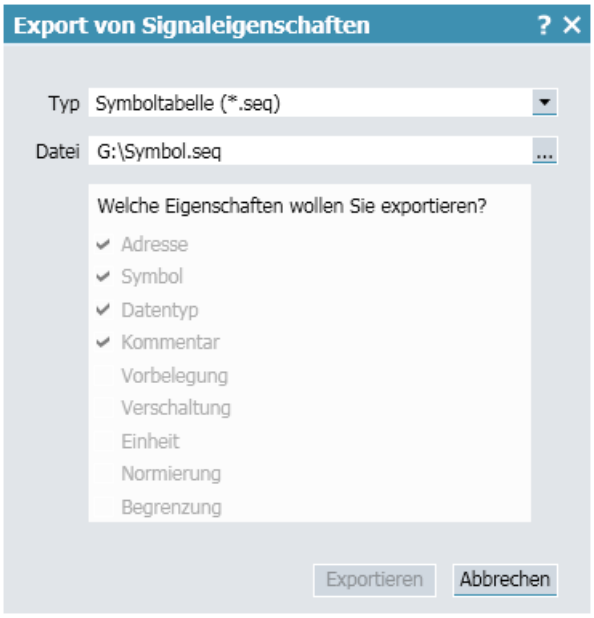
- "Datenbreite automatisch anpassen"

  Hinweis

  Dieses Optionskästchen ist nur bei folgenden Kopplungstypen verfügbar:
  - SIMIT Unit
  - Virtual Controller
  - PLCSIM Advanced

Legt fest, dass beim Import die Datenbreite von Signalen automatisch an die in der Import-Datei angegebene Datenbreite angepasst wird. Weitere Informationen finden Sie im Abschnitt: Umwandeln der Datenbreite von Signalen (Seite 196).
Wenn das Optionskästchen nicht aktiv ist, und die Datenbreite des zu importierenden Signals nicht zur vorhandenen passt, wird dieses Signal nicht importiert.

9.2.6 Dialogfeld "Export von Signaleigenschaften"



Mit diesem Dialogfeld nehmen Sie folgende Einstellungen zum Export der Signaleigenschaften vor:

- "Typ"
  Legt das Dateiformat der Export-Datei fest. Das Dateiformat ist abhängig vom Kopplungstyp, dessen Signaleigenschaften Sie exportieren.
  - SIMATIC-Symboltabelle aus STEP 7/PCS 7 (seq-Datei)
    Dieses Format enthält ausschließlich die Signale aus der Kopplung, für die ein symbolischer Name vergeben ist.
    Weitere Informationen hierzu finden Sie im Abschnitt: Symboltabelle (Seite 200).
  - Signaltabelle (txt-Datei)
    Dieses Format enthält alle Signale aus der Kopplung.
    Weitere Informationen hierzu finden Sie im Abschnitt: Signaltabelle (Seite 202).
  - INI-Format

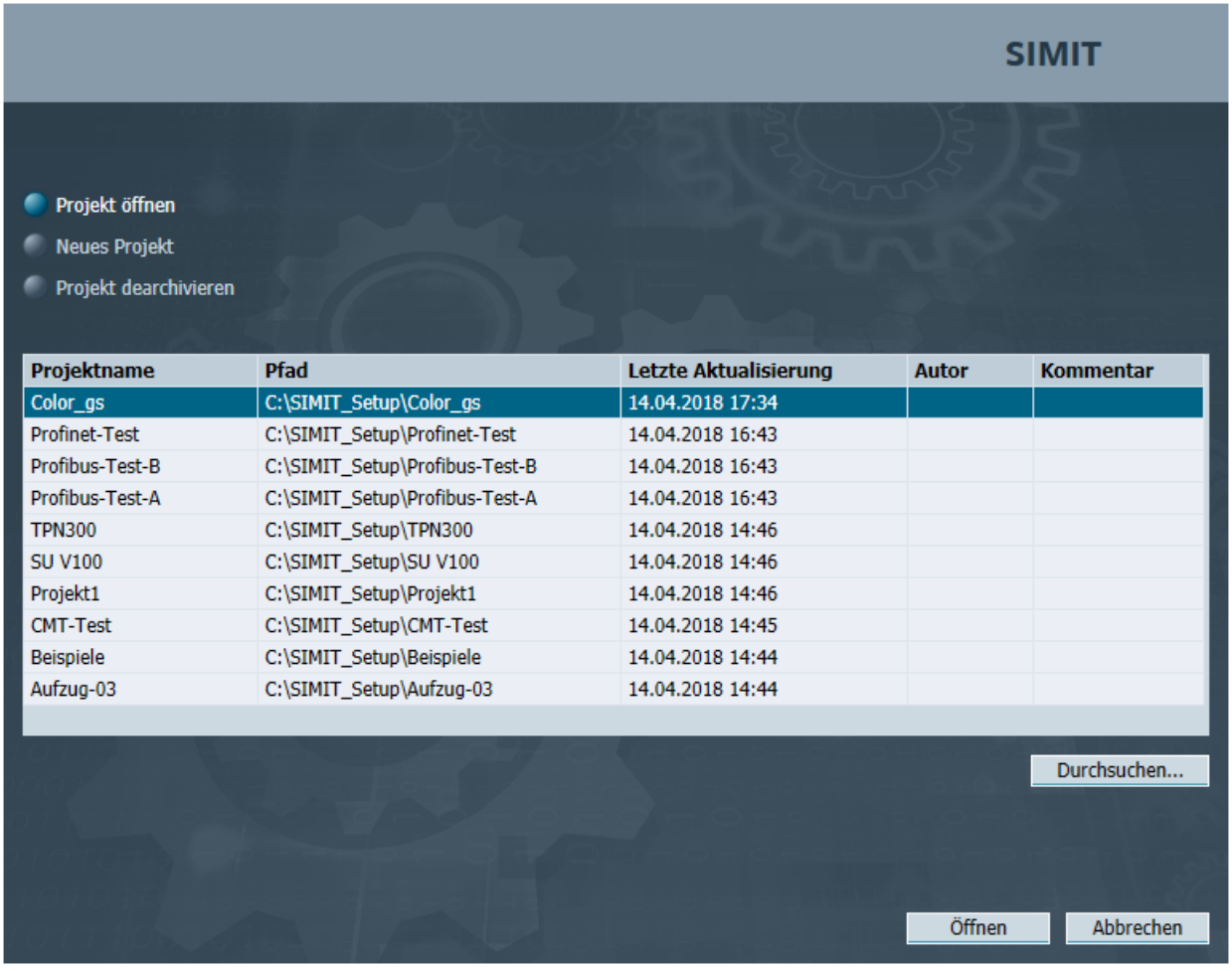
    Hinweis

    OPC-Kopplung

    Da OPC-Signale weder eine Adresse besitzen, noch als normierter Rohwert übertragen werden können, kann die Signaltabelle nur unvollständig ausgefüllt werden. Eine solche Signaltabelle kann in eine andere OPC-Kopplung importiert werden, es ist aber kein Import in Kopplungen möglich, die SIMATIC-Signale erwarten.

- "Datei"
  Legt den Speicherort für die Export-Datei fest.
- "Bereich "**Welche Eigenschaften wollen Sie exportieren?**""
  Legt fest, welche Signaleigenschaften exportiert werden. Die "Adresse" identifiziert ein Signal eindeutig und wird immer exportiert. Wenn Sie als Dateiformat "Symboltabelle" gewählt haben, werden zusätzlich folgende Signaleigenschaften exportiert:
  - Symbol
  - Datentyp
  - Kommentar
- "Exportieren"
  Startet den Export.

9.2.7 Dialogfeld "Projektauswahl"



Dieses Dialogfeld wird durch folgende Menübefehle geöffnet:

- "Projekt > Neues Projekt..."
- "Projekt > Öffnen"
- "Projekt > Dearchivieren"

Die Ansicht des Dialogfelds variiert je nach Auswahl.

In diesem Dialogfeld führen Sie folgende Funktionen aus:

- "Projekt öffnen"
  Öffnet ein bereits vorhandenes Projekt.
  Weitere Informationen hierzu finden Sie im Abschnitt: Projekt > Öffnen... (Seite 953). SIMIT-Projekte haben die Dateiendung ".simit".
- "Neues Projekt"
  Legt ein neues Projekt an.
  Weitere Informationen hierzu finden Sie im Abschnitt: Projekt > Neues Projekt... (Seite 952).
- "Projekt dearchivieren"
  Dearchiviert ein archiviertes Projekt.
  Weitere Informationen hierzu finden Sie im Abschnitt: Projekt > Dearchivieren (Seite 955).
  Archivierte SIMIT-Projekte haben die Dateiendung ".simarc".

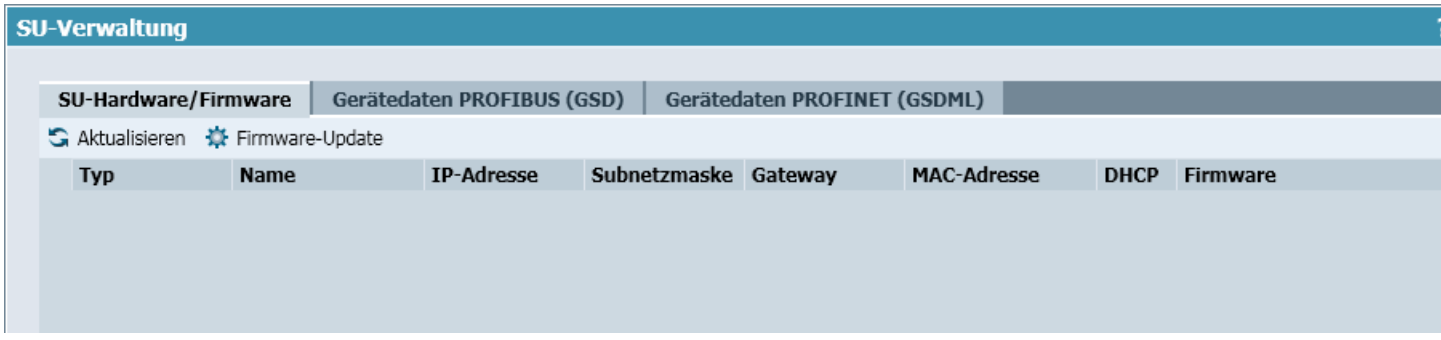
SIMIT-Projekte können in beliebigen Ordnern im Dateisystem abgelegt werden. Zur Auswahl eines Ordners klicken Sie auf die Schaltfläche "Durchsuchen...".

Hinweis

SIMIT verwaltet in diesem Ordner alle Informationen, die für das Projekt relevant sind. Beachten Sie Folgendes:

- Benennen Sie diesen Ordner nicht um.
- Ändern Sie keine Dateien in diesem Ordner.
- Legen Sie keine eigenen Dateien in diesem Ordner ab.
- Legen Sie keine Ordner in diesem Ordner an.

9.2.8 Dialogfeld "SU-Verwaltung"



In diesem Dialogfeld konfigurieren Sie die SIMIT Unit.

Registerkarte "SU-Hardware/Firmware"

Auf dieser Registerkarte werden alle im Netzwerk erreichbaren SIMIT Units aufgelistet. Weiterführende Informationen finden Sie unter SIMIT Unit konfigurieren (Seite 87).

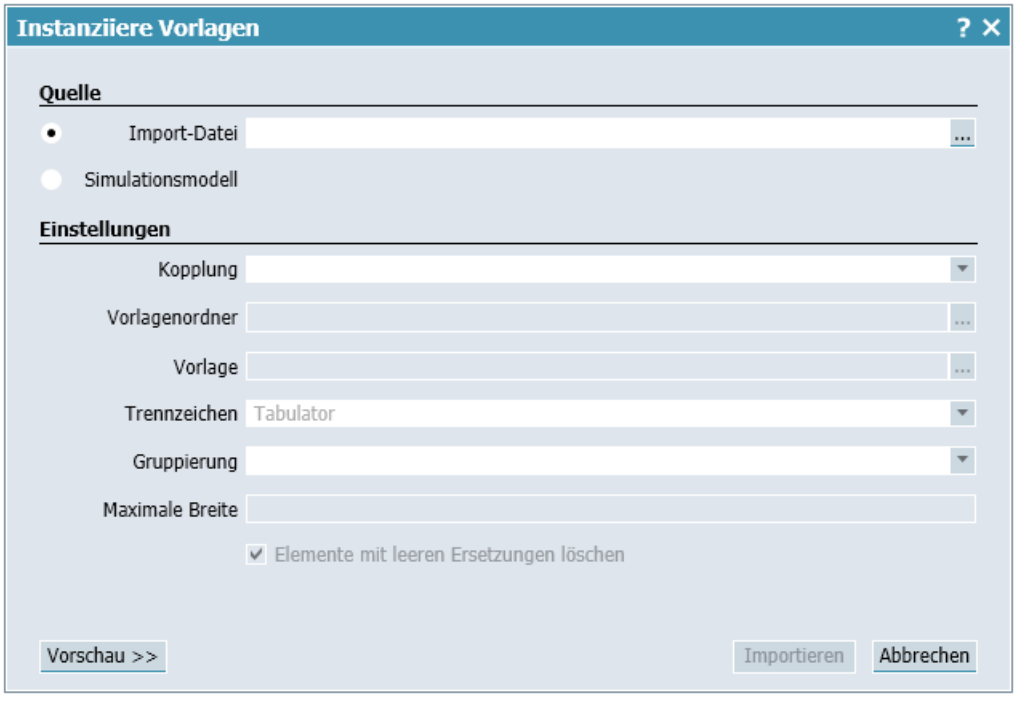
- **Aktualisieren**
  Aktualisiert die Ansicht.
- **Firmware-Update**
  Aktualisiert die Firmware der gewählten SIMIT Unit.

Registerkarten "Gerätedaten PROFIBUS (GSD)" und "Gerätedaten PROFINET (GSDML)"

Auf diesen Registerkarten werden die von der SIMIT Unit unterstützten Geräte aufgelistet. Über den Import von Gerätedaten erweitern Sie die Liste der unterstützten Geräte. Weiterführende Informationen finden Sie unter Gerätebeschreibungsdatei in SIMIT importieren (Seite 88).

- **Aktualisieren**
  Aktualisiert die Ansicht.
- **Import**
  Öffnet ein Dialogfeld zum Import von Gerätedaten aus einer Gerätebeschreibungsdatei.
- **Löschen**
  Entfernt ein aus einer Gerätebeschreibungsdatei importiertes Gerät aus der Liste.

9.2.9 Dialogfeld "Instanziiere Vorlagen"



In diesem Dialogfeld nehmen Sie folgende Einstellungen zum Instanziieren einer Vorlage vor:

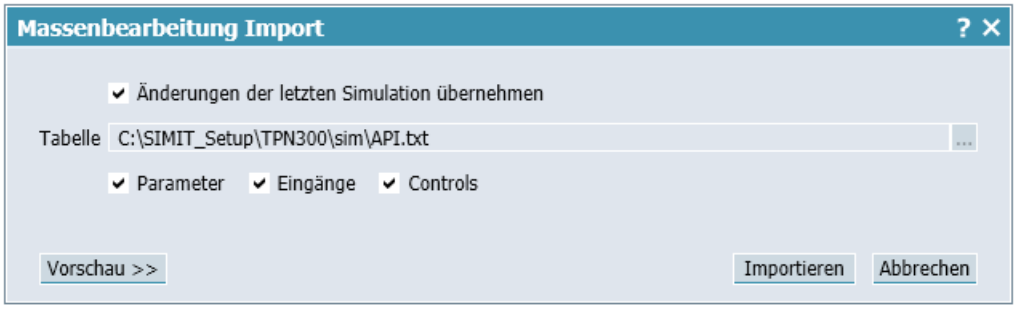
- "Import-Datei"
  Legt fest, dass die Vorlagen aus einer Datei instanziiert werden.
  Legt die Import-Datei fest. Folgende Dateitypen können importiert werden: *.txt, *.xls, *.xlsx, *.iea; *.xml.
- "Simulationsmodell"
  Legt fest, dass die Vorlagen aus dem Simulationsmodell instanziiert werden.
- "Kopplung"
  Legt die verwendete Kopplung fest.
- "Vorlagenordner"
  Legt den Ordner fest, in dem nach den in der CMT-Datei referenzierten SIMIT-Vorlagen gesucht wird.
  Der Name des in der CMT-Datei angegebenen CMT-Typs muss mit dem Namen der in SIMIT zu verwendenden Vorlage übereinstimmen.
- "Vorlage"
  Legt den Speicherort der Vorlage fest.
- "Trennzeichen"
  Legt das Trennzeichen fest.
- "Gruppierung"
  Legt die Gruppierung fest.
- "Maximale Breite"/"Maximale Höhe"
  Legt die Breite/Höhe in Pixeln fest.
- "Elemente mit leeren Ersetzungen löschen"
  Legt fest, ob leere Ersetzungen aus der Datei entfernt werden sollen.
- "Vorschau >>"
  Zeigt eine Vorschau des Imports an.
  In der Vorschau sehen Sie auf der linken Seite die durch den Import neu entstehenden Ordner und Diagramme. Einzelne Ordner und Diagramme sind abgewählbar. Bei Auswahl eines Diagramms auf der linken Seite, sehen Sie rechts die durchzuführenden Ersetzungen. Dabei ist rechts in der zweiten Kopfzeile die jeweils verwendete Vorlage eingetragen und vermerkt, ob eine Vorlage überhaupt gefunden wurde.
- "Importieren"
  Startet den Import.

Siehe auch

Vorlagen aus Dateien instanziieren (Seite 259)

Vorlagen aus dem Simulationsmodell instanziieren (Seite 269)

9.2.10 Dialogfeld "Massenbearbeitung Import"



In diesem Dialogfeld nehmen Sie folgende Einstellungen zur automatischen Parametrierung und zum Massendatenimport vor:

- "Änderungen der letzten Simulation übernehmen"
  Legt fest, ob die Änderungen der zuletzt aktiven Simulation für dieses Projekt automatisch übernommen werden sollen.
  Das Optionskästchen ist nur aktiv, wenn für das Projekt mindestens eine Simulation bereits gelaufen ist.
  Zum Import von Massendaten deaktivieren Sie diese Option.
- "Tabelle"
  Legt die Datei fest, aus der die Daten gelesen werden.
- "Parameter"
  Legt fest, ob die Parameter aus der Tabelle übernommen werden sollen.
- "Eingänge"
  Legt fest, ob die Eingangsvorbelegungen aus der Tabelle übernommen werden sollen.
- "Controls"
  Legt fest, ob die Voreinstellungen von Controls aus der Tabelle übernommen werden sollen.
- "Vorschau >>"
  Zeigt eine Vorschau der automatischen Parametrierung an.
- "Importieren"
  Führt die automatische Parametrierung durch / Importiert die geänderten Werte.

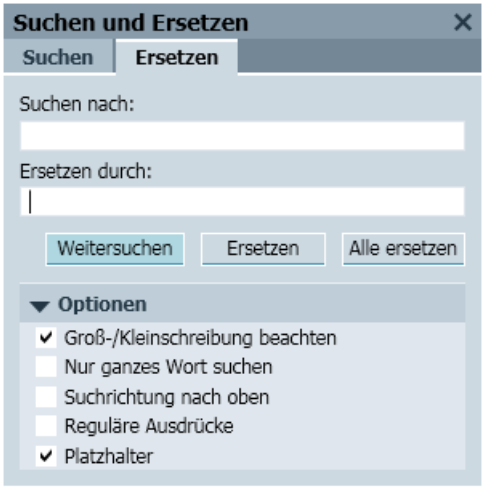
Weitere Informationen hierzu finden Sie im Abschnitt: Daten importieren (Seite 282).

9.2.11 Editor "Kennlinie"

Mit dem Kennlinieneditor erstellen Sie eine Kennlinie durch Vorgabe von Stützpunkten und der Interpolation zwischen diesen Stützpunkten.

Weitere Informationen finden Sie im Abschnitt: Characteristic – Kennlinie (Seite 418)

9.2.12 Dialogfeld "Suchen & Ersetzen"

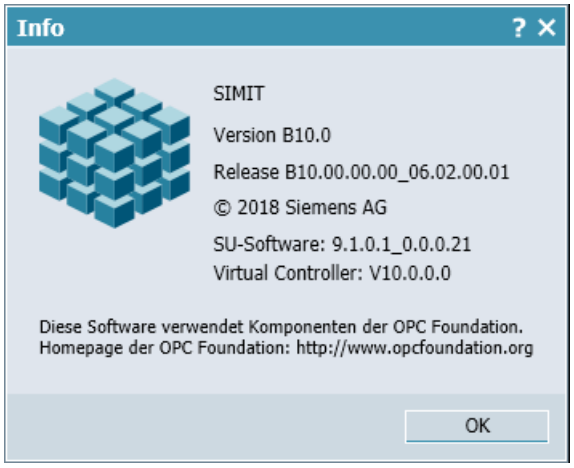


Öffnen Sie das Dialogfeld, indem Sie bei Texteingabe die Tastenkombination <Strg+F> für "Suchen" oder <Strg+H> für "Ersetzen" drücken. Gesucht wird nur in dem Bereich, in dem Sie das Dialogfeld aufgerufen haben.

In diesem Dialogfeld nehmen Sie folgende Einstellungen vor:

- "Suchen nach"
  Legt den Suchbegriff fest.
- "Weitersuchen"
  Sucht nach dem nächsten Auftreten des Suchbegriffs.
- "Ersetzen durch"
  Legt den Begriff fest, durch den der Suchbegriff ersetzt wird.
- "Ersetzen"
  Ersetzt den gefundenen Suchbegriff.
  Danach wird automatisch die Funktion "Weitersuchen" ausgeführt.
- "Alle ersetzen"
  Ersetzt alle gefundenen Suchbegriffe.
- "Optionen"
  Blendet einen Konfigurationsbereich ein, indem Sie zusätzliche Optionen aktivieren können.
  Unterstützt werden z. B. "reguläre Ausdrücke" oder "Platzhalter". Als Platzhalter können Sie z. B. folgende Zeichen unterstützt:
  - "?": Repräsentiert ein variables Zeichen. Der Suchbegriff "H?llo" findet z. B. "Hello" und "Hallo"
  - "*": Repräsentiert eine beliebige Anzahl variabler Zeichen

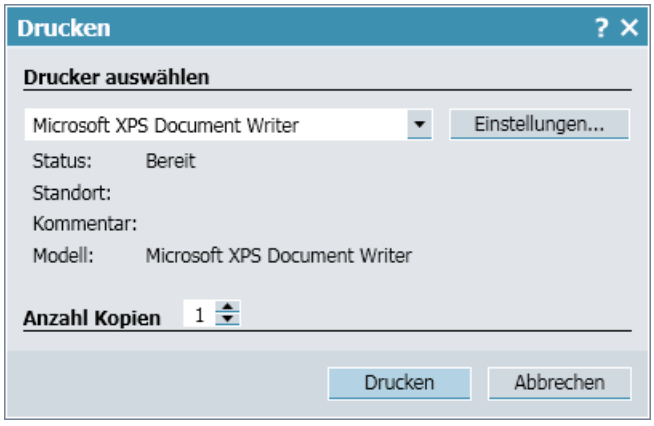
9.2.13 Dialogfeld "Info"



Mit diesem Menübefehl werden folgende Informationen über Ihre SIMIT-Installation angezeigt:

- Version
- Version der SU-Software
- Version von SIMIT Virtual Controller (VC) (wenn installiert)

9.2.14 Dialogfeld "Drucken"



Wählen Sie einen Drucker aus der Klappliste aus und stellen Sie die Anzahl der Ausdrucke ein.

Mit der Schaltfläche "Einstellungen..." können Sie die Einstellungen des ausgewählten Druckers ändern.

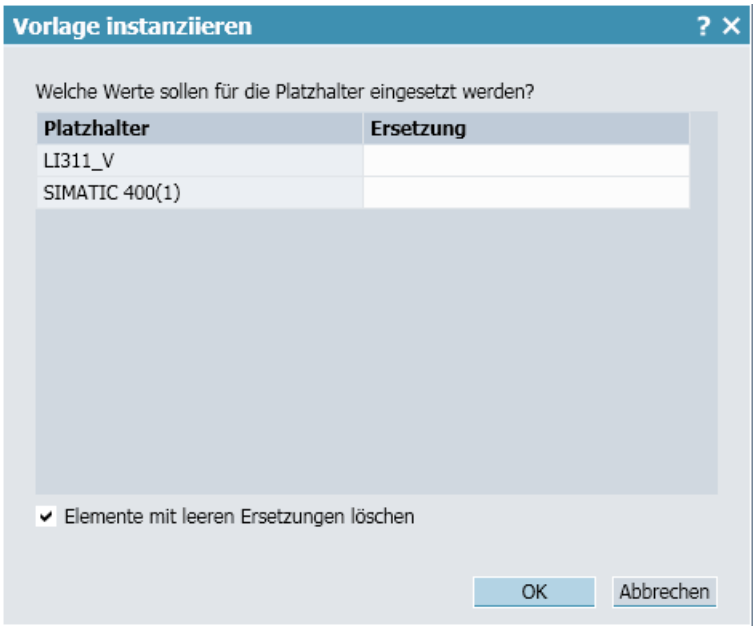
Klicken Sie auf die Schaltfläche "Drucken", um den Druckvorgang zu starten und das Dialogfeld zu schließen.

Siehe auch

Drucken von Diagrammen (Seite 231)

Drucken von Kurvenbildern (Seite 298)

9.2.15 Dialogfeld "Vorlage instanziieren"

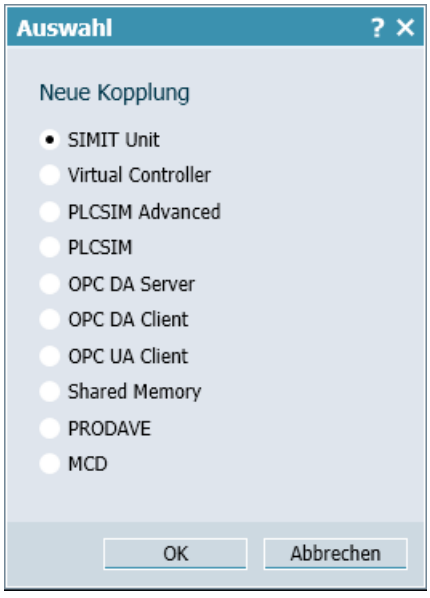


In diesem Dialogfeld werden die Parameter der ausgewählten Vorlage in der Spalte "Platzhalter" aufgelistet. In die Spalte "Ersetzung" können als Ersatz beliebige Werte eingetragen werden.

Aktivieren Sie das Optionskästchen "Elemente mit leeren Ersetzungen löschen", um die Parameter aus der Vorlage zu entfernen, für die Sie keinen Namen in die Spalte "Ersetzung" eingegeben haben.

Klicken Sie auf die Schaltfläche "OK", um die Änderungen zu übernehmen und das Dialogfeld zu schließen.

9.2.16 Dialogfeld "Auswahl"

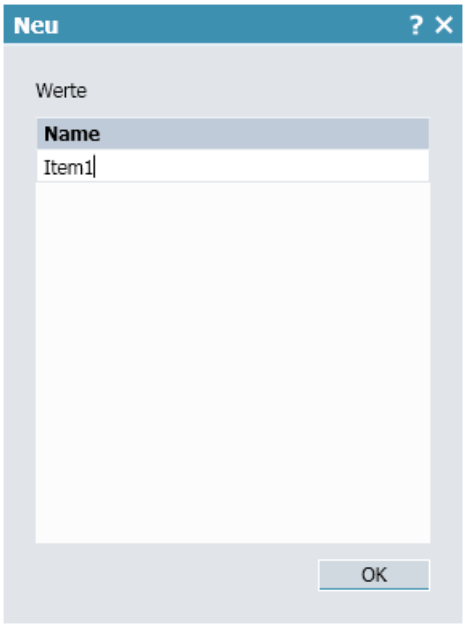


In diesem Dialogfeld wählen Sie eine neue Kopplung aus. Folgende Kopplungen stehen zur Auswahl:

- **SIMIT Unit**
  Weitere Informationen zur Kopplung finden Sie unter SIMIT Unit (Seite 67).
- **Virtual Controller**
  Weitere Informationen zur Kopplung finden Sie unter Virtual Controller (Seite 109).
- **PLCSIM Advanced**
  Weitere Informationen zur Kopplung finden Sie unter PLCSIM Advanced-Kopplung (Seite 145).
- **PLCSIM**
  Weitere Informationen zur Kopplung finden Sie unter PLCSIM-Kopplung (Seite 150).
- **OPC DA Server**
  Weitere Informationen zur Kopplung finden Sie unter OPC-Kopplung (Seite 152).
- **OPC DA Client**
  Weitere Informationen zur Kopplung finden Sie unter OPC-Kopplung (Seite 152).
- **OPC UA Client**
  Weitere Informationen zur Kopplung finden Sie unter OPC-Kopplung (Seite 152).
- **Shared Memory**
  Weitere Informationen zur Kopplung finden Sie unter Shared Memory-Kopplung (Seite 168).
- **PRODAVE**
  Weitere Informationen zur Kopplung finden Sie unter PRODAVE-Kopplung (Seite 177).
- **MCD**
  Weitere Informationen zur Kopplung finden Sie unter MCD-Kopplung (Seite 180).

Klicken Sie auf die Schaltfläche "OK", um die Auswahl zu bestätigen und das Dialogfeld zu schließen.

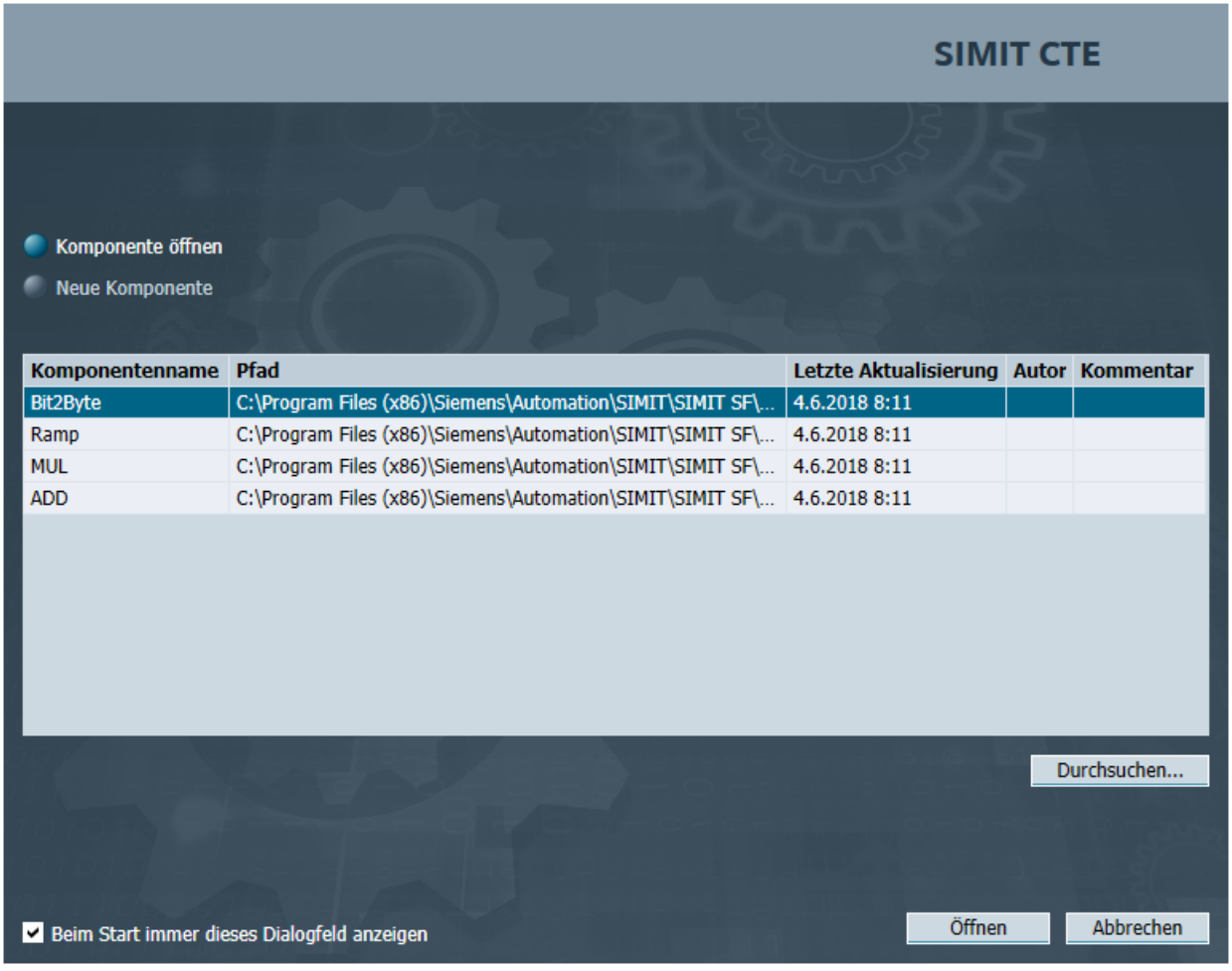
9.2.17 Dialogfeld "Aufzählung" (CTE)



In diesem Dialogfeld können Sie Namen für die Elemente eines eigenen Aufzählungstyps eingeben.

Weitere Informationen finden Sie im Abschnitt: Die Task-Card "Aufzählungstypen" (Seite 339).

9.2.18 Dialogfeld "Komponente verwalten" (CTE)



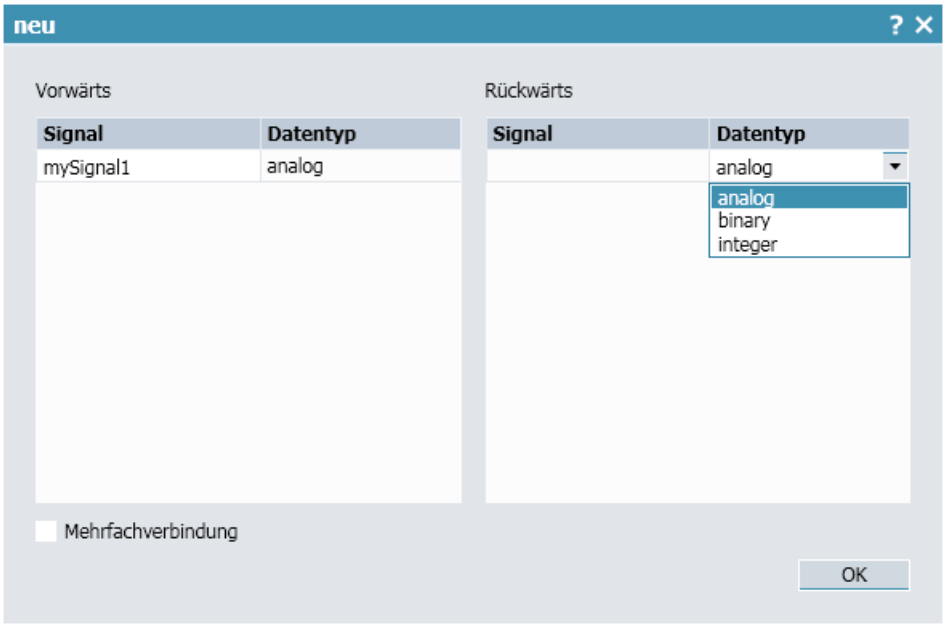
In diesem Dialogfeld können Sie eine neue Komponente erstellen oder eine bereits vorhandene öffnen. Wählen Sie dazu die gewünschte Funktion aus.

Klicken Sie auf die Schaltfläche "Durchsuchen...", um den Speicherort der Komponente auszuwählen.

Klicken Sie auf die Schaltfläche "Öffnen", um die Komponente im Editor zu öffnen.

Aktivieren Sie das Optionskästchen "Beim Start immer dieses Dialogfeld anzeigen", wenn beim Start des CTE automatisch immer dieses Dialogfeld geöffnet werden soll.

9.2.19 Dialogfeld "Verbindungstyp" (CTE)



In diesem Dialogfeld definieren Sie die Signale eines eigenen Verbindungstyps. Den Datentyp des Signals wählen Sie aus der Klappliste aus. Sie können beliebig viele Signale anlegen.

Aktivieren Sie das Optionskästchen "Mehrfachverbindung" wenn ein Ausgang mit mehr als einem Eingang verbunden werden soll.

Weitere Informationen finden Sie im Abschnitt: Die Task-Card "Verbindungstypen" (Seite 340).