



SIEMENS



Lern-/Lehrunterlage

Siemens Automation Cooperates with Education
(SCE) | Ab Version V9 SP1

PA Modul P01-04
SIMATIC PCS 7 – Einzelsteuerfunktionen

[siemens.de/sce](https://www.siemens.de/sce)

SIEMENS

Global Industry
Partner of
WorldSkills
International



Passende SCE Trainer Pakete zu dieser Lern-/Lehrunterlage

- **SIMATIC PCS 7 Software 3er Paket V9.0**
Bestellnr.: 6ES7650-0XX58-0YS5
- **SIMATIC PCS 7 Software 6er Paket V9.0**
Bestellnr.: 6ES7650-0XX58-2YS5
- **SIMATIC PCS 7 Software Upgrade Pakete 3er**
Bestellnr.: 6ES7650-0XX58-0YE5 (V8.x→ V9.0)
- **SIMIT Simulation Platform mit Dongle V10**
(beinhaltet SIMIT S & CTE, FLOWNET, CONTEC Bibliotheken) – 2500-Simulation-Tags
Bestellnr.: 6DL8913-0AK00-0AS5
- **Upgrade SIMIT Simulation Platform V10**
(beinhaltet SIMIT S & CTE, FLOWNET, CONTEC Bibliotheken) von V8.x/V9.x
Bestellnr.: 6DL8913-0AK00-0AS6
- **Demo-Version SIMIT Simulation Platform V10**
[Download](#)
- **SIMATIC PCS 7 AS RTX Box (PROFIBUS) nur in Kombination mit ET 200M für RTX –**
Bestellnr.: 6ES7654-0UE23-0XS1
- **ET 200M für RTX Box (PROFIBUS) nur in Kombination mit PCS 7 AS RTX Box –**
Bestellnr.: 6ES7153-2BA10-4AB1

Bitte beachten Sie, dass diese Trainer Pakete ggf. durch Nachfolge-Pakete ersetzt werden.
Eine Übersicht über die aktuell verfügbaren SCE Pakete finden Sie unter: [siemens.de/sce/tp](https://www.siemens.de/sce/tp)

Fortbildungen

Für regionale Siemens SCE Fortbildungen kontaktieren Sie Ihren regionalen SCE Kontaktpartner:
[siemens.de/sce/contact](https://www.siemens.de/sce/contact)

Weitere Informationen rund um SCE

[siemens.de/sce](https://www.siemens.de/sce)

Verwendungshinweis

Die SCE Lern-/Lehrunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (TIA) wurde für das Programm "Siemens Automation Cooperates with Education (SCE)" speziell zu Ausbildungszwecken für öffentliche Bildungs- und F&E-Einrichtungen erstellt. Siemens übernimmt bezüglich des Inhalts keine Gewähr.

Diese Unterlage darf nur für die Erstausbildung an Siemens Produkten/Systemen verwendet werden. D. h. Sie kann ganz oder teilweise kopiert und an die Studierenden zur Nutzung im Rahmen deren Studiums ausgehändigt werden. Die Weitergabe sowie Vervielfältigung dieser Unterlage und Mitteilung Ihres Inhalts ist innerhalb öffentlicher Aus- und Weiterbildungsstätten für Zwecke im Rahmen des Studiums gestattet.

Ausnahmen bedürfen der schriftlichen Genehmigung durch Siemens. Alle Anfragen hierzu an scsupportfinder.i-ia@siemens.com.

Zuwendungen verpflichten zu Schadensersatz. Alle Rechte auch der Übersetzung sind vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

Der Einsatz für Industriekunden-Kurse ist explizit nicht erlaubt. Einer kommerziellen Nutzung der Unterlagen stimmen wir nicht zu.

Wir danken der TU Dresden, besonders Prof. Dr.-Ing. Leon Urbas und der Fa. Michael Dziallas Engineering und allen weiteren Beteiligten für die Unterstützung bei der Erstellung dieser SCE Lehrunterlage.

Inhaltsverzeichnis

1	Zielstellung.....	5
2	Voraussetzung.....	5
3	Benötigte Hardware und Software	6
4	Theorie	7
4.1	Theorie in Kürze	7
4.2	Objektorientierte Softwarestrukturierung	9
4.3	Kanalfunktionen (Driver)	9
4.4	Einzelsteuerfunktionen	10
4.5	Schutzmaßnahmen.....	11
4.6	Betriebsarten	11
4.7	Funktionsbaustein-Typen in PCS 7.....	12
4.8	Literatur.....	15
5	Aufgabenstellung	16
6	Planung	16
7	Lernziel	17
8	Strukturierte Schritt-für-Schritt-Anleitung.....	18
8.1	Symboltabelle bearbeiten	18
8.2	Baustein Motor_Lean aus der PCS 7 AP Library V90 in Projekt-bibliothek einfügen	21
8.3	Motor_Lean zentral bearbeiten	23
8.4	Implementierung einer Instanz von Motor_Lean.....	28
8.5	Simulation mit PLCSIM vorbereiten.....	37
8.6	Objekte übersetzen und laden	40
8.7	Testen des Bausteins A1T2S003.....	47
8.8	Checkliste – Schritt-für-Schritt-Anleitung	53
9	Testen der Automatisierungslogik mit der Simulation (optional)	54
10	Übungen	59
10.1	Übungsaufgaben	59
10.2	Checkliste – Übung.....	62
11	Weiterführende Information	63

Einzelsteuerfunktionen

1 Zielstellung

Die Studierenden können nach der Bearbeitung dieses Moduls den Begriff der Einzelsteuerfunktion im Rahmen der objektorientierten Softwarestrukturierung definieren und einordnen. Sie verstehen das Konzept, den Aufbau sowie die Funktionsweise von Einzelsteuerfunktionen und kennen typische Einzelsteuerfunktionen sowie deren Umsetzung in **PCS 7**.

2 Voraussetzung

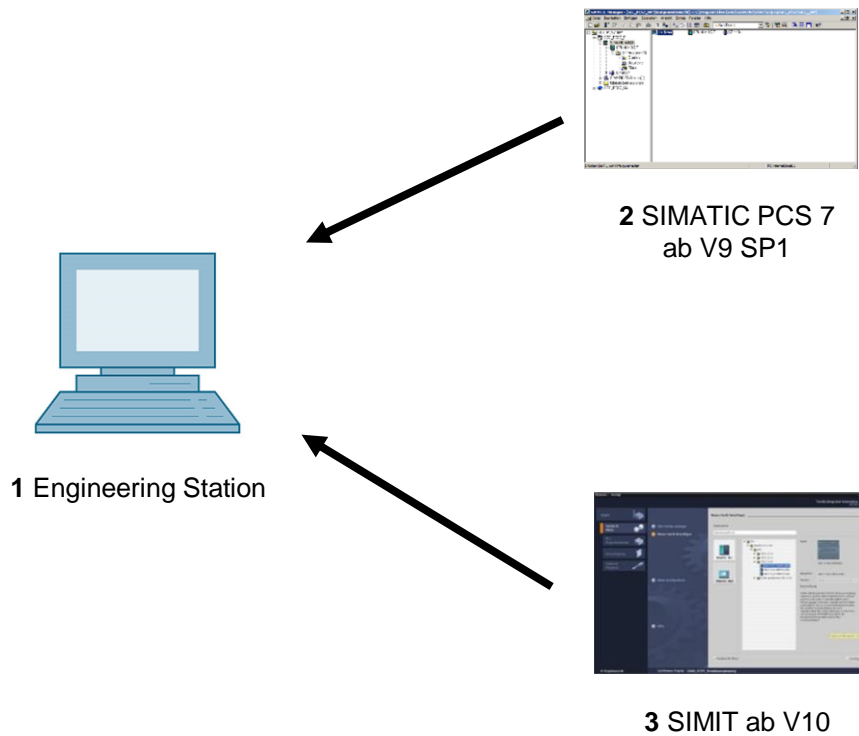
Dieses Kapitel baut auf das Kapitel ‚Technologische Hierarchie‘ auf. Zur Durchführung des Kapitels kann ein bereits bestehendes Projekt aus dem vorhergehenden Kapitel oder das durch SCE zur Verfügung gestellte archivierte Projekt ‚p01-03-exercise-r1905-de.zip‘ genutzt werden. Der Download des Projekts (bzw. der Projekte) ist beim jeweiligen Modul im SCE Internet hinterlegt.

Zusätzlich wird die Datei p01-04-symbols-r1905-de.dif (im Zip-file p01-04-files-r1905-de.zip‘ hinterlegt) benötigt.

Die Simulation für das Programm SIMIT kann aus der Datei p01-04-plantsim-v10-r1905-de.simarc dearchiviert werden. Es ist im Demo-Modus lauffähig.

3 Benötigte Hardware und Software

- 1 Engineering Station: Voraussetzungen sind Hardware und Betriebssystem
(weitere Informationen siehe Readme/Liesmich auf den PCS 7 Installations-DVDs)
- 2 Software SIMATIC PCS 7 – ab V9 SP1
 - Installierte Programm-Pakete (enthalten im Trainer Paket SIMATIC PCS 7 Software):
 - *Engineering* → *PCS 7 Engineering*
 - *Engineering* → *BATCH Engineering*
 - *Runtime* → *Single Station* → *OS Single Station*
 - *Runtime* → *Single Station* → *BATCH Single Station*
 - *Options* → *SIMATIC Logon*
 - *Options* → *S7-PLCSIM V5.4 SP8*
- 3 Demo-Version SIMIT Simulation Platform V10



4 Theorie

4.1 Theorie in Kürze

Das Ziel der objektorientierten Softwarestrukturierung besteht darin, die Struktur der realen Anlage durch eine entsprechende Modularisierung der Anwendersoftware möglichst eindeutig nachzubilden. Dazu wird für jeden Feldgerätetyp mindestens ein Funktionsbaustein bereitgestellt, der die gesamte Ansteuerungslogik, notwendige Schutz- und Überwachungsfunktionen sowie geeignete Bedien- und Visualisierungsmöglichkeiten bereitstellt. Das Anwenderprogramm nutzt diesen Baustein, um das gewünschte Betriebsverhalten einer Maschine oder eines Prozesses zu realisieren.

Motoren und Ventile sind steuerungstechnische Einrichtungen, die im Sinne einer objektorientierten Automatisierung ebenfalls nicht direkt angesteuert, sondern zunächst als Funktionsbaustein-Typen modelliert werden. Derartige Funktionsbaustein-Typen bezeichnet man als **Einzelsteuerfunktionen (ESF)** oder **Individual Drive Functions (IDF)**. Sie ermöglichen die Steuerung, Überwachung und Bedienung der steuerungstechnischen Einrichtung durch die Bereitstellung von entsprechenden Anschlüssen für Stell- und Steuersignale sowie für Parametrier- und Überwachungsfunktionen. Die technische Umsetzung der Steuerung wird durch eine Instanz des Funktionsbaustein-Typs realisiert und bleibt dem Nutzer verborgen. Abbildung 1 zeigt den Übergang vom realen Motor, in diesem Fall eine Pumpe, hin zu einem Baustein der entsprechenden Einzelsteuerfunktion.

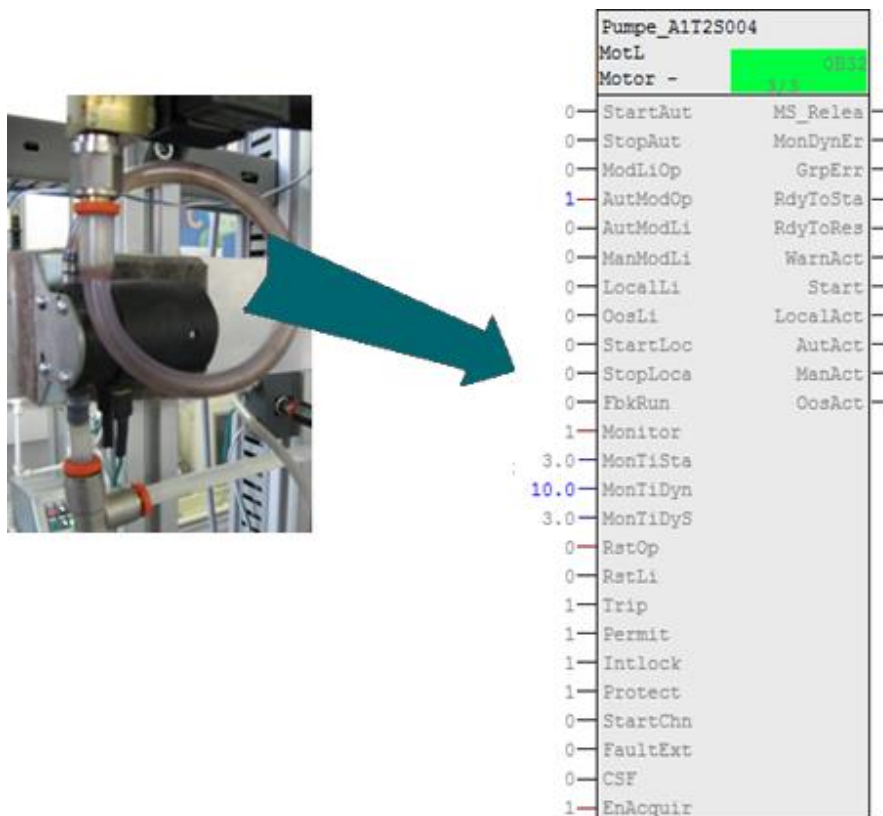


Abbildung 1: Der Übergang vom realen Motor zum Baustein der Einzelsteuerfunktion

Steuerungstechnische Einrichtungen können grundsätzlich in vier verschiedenen Betriebsarten betrieben werden. Das Gerät kann folgende Betriebsarten aufweisen:

- **Außer Betrieb**
- im **Handbetrieb**
- im **Automatikbetrieb** oder
- im **Vor-Ort-Betrieb**

Eine Einzelsteuerfunktion darf stets nur eine Betriebsart einnehmen. Die genannten Betriebsarten können gleichwertig oder über Prioritäten hierarchisch gegliedert sein. Einzelsteuerfunktionen bieten zudem Funktionen zum Schutz vor Geräte- und Prozessfehlern an. Dazu werden verschiedene Verriegelungen sowie eine Laufzeitüberwachung für das Gerät und den gesteuerten Prozess implementiert.

Funktionsbaustein-Typen, in **PCS 7** als Bausteintypen bezeichnet, stellen vorgefertigte Programmteile für die Bearbeitung wiederkehrender Funktionen dar. Sie können in CFC-Pläne eingefügt und dort als Instanzen parametrisiert, verschaltet und an die Projekterfordernisse angepasst werden. Dabei legt der Baustein-Typ die Charakteristik für sämtliche Instanzen dieses Typs fest. **PCS 7** bietet bereits eine Vielzahl von leistungsfähigen und getesteten Einzelsteuerfunktionen als Bausteintypen in den leittechnischen Bibliotheken an. Diese modellieren jeweils eine steuerungstechnische Einrichtung und stellen die gesamte Ansteuerlogik bereit.

Zusätzlich werden Funktionen angeboten:

- zum **Bedienen** und **Beobachten** der Einzelsteuerfunktion
- zum **Steuern** von Signalen
- zur **Überwachung** und **Alarmierung**
- zur **Betriebszustandsauswahl**
- für **Verriegelungen**.

Bildbausteine mit verschiedenen **Sichten** ermöglichen die nahtlose Integration in ein entsprechendes Prozessleitsystem.

Einzelsteuerfunktionen bieten eine effiziente Entwicklung leistungsfähiger, qualitativ hochwertiger Lösungen. Sie modularisieren und typisieren wiederkehrende Funktionalitäten. Daraufhin werden diese wiederverwendbar und zentral änderbar, was den Entwicklungsprozess erheblich beschleunigt.

4.2 Objektorientierte Softwarestrukturierung

Das Ziel der objektorientierten Softwarestrukturierung besteht darin, die Struktur der realen Anlage durch eine entsprechende Modularisierung der Anwendersoftware möglichst eindeutig nach-zubilden. Dazu wird für jedes Feldgerät in der Anlage ein eigenes Programm erstellt. Für jeden Feldgerätetyp wird mindestens ein Funktionsbaustein bereitgestellt.

Dieser Baustein realisiert die gesamte Ansteuerungslogik für diesen Feldgerätetyp. Darüber hinaus stellt er notwendige Schutz- und Überwachungsfunktionen sowie geeignete Bedien- und Visualisierungsmöglichkeiten bereit. Er kapselt somit die gesamte Funktionalität, die im Zusammenhang mit dem entsprechenden Feldgerätetyp notwendig ist. Das Anwenderprogramm nutzt diesen Baustein, um eine gewünschte Steuerung für eine Maschine oder einen Prozess zu realisieren, ohne dabei auf Kenntnisse der internen Daten und Operationen des Funktionsbausteins zurückgreifen zu müssen.

4.3 Kanalfunktionen (Driver)

In Ergänzung zur Behandlung der Feldgeräte durch eigene wiederverwendbare Bausteine ist es häufig sinnvoll, die Peripherieanbindung ebenfalls durch **Kanalbausteine** (engl. Driver) zu abstrahieren. Es ist zwar stets möglich, über Symbolnamen oder Adresse direkt auf das Prozessabbild zuzugreifen, allerdings müssen hierbei die möglicherweise vielfältigen Parameter zur Konfiguration des Kanals an anderer Stelle gesetzt werden. Dies führt schnell zu unübersichtlichen Programmen. **PCS 7** bietet eine Reihe von Kanalbausteinen, die zum einen die Statussignale der Baugruppen auswerten und zum anderen das Testen und die Inbetriebnahme von Automatisierungsprogrammen durch Simulationsmodi unterstützen. In den analogen Kanalbausteinen entsprechend Tabelle 1 erfolgt zudem durch die Parameter VLRANGE und VHRANGE eine Abbildung von den internen digitalen Größen auf die physikalischen Rechen- und Anzeige Größen. **PCS 7** kann bei Verwendung von Kanalbausteinen die notwendigen Treiber automatisch erzeugen. Die Kanalbausteine werden deshalb in den Vorlagen der **PCS 7** Bibliotheken vielfach eingesetzt.

Kanalbaustein	Baustein	Verschaltung, Parameter
Digitaler Ausgang	PCS7DiOu	PV_Out
Digitaler Eingang	PCS7DiIn	PV_In
Analoger Ausgang	PCS7AnOu	PV_Out, Scale
Analoger Eingang	PCS7AnIn	PV_In, Scale

Tabelle 1: Auflistung verschiedener Kanalbausteine zur Abstraktion der Peripherieanbindung

4.4 Einzelsteuerfunktionen

Motoren und Ventile sind als steuerungstechnische Einrichtungen in der Fabrik- und Prozessautomatisierung von zentraler Bedeutung. Es existiert eine Vielzahl gängiger Typen mit spezifischem Bedien- und Meldeverhalten. Im Sinne einer objektorientierten Automatisierung werden derartige Einrichtungen nicht direkt angesteuert, sondern zunächst als Funktionsbaustein-Typen modelliert. Die Ansteuerung erfolgt stets indirekt über eine Instanz des entsprechenden Funktionsbaustein-Typs. Funktionsbausteine für Motoren und Ventile heißen **Einzelsteuerfunktionen (ESF)** oder **Individual Drive Functions (IDF)**. Einzelsteuerfunktionen ermöglichen die Steuerung, Überwachung und Bedienung von steuerungstechnischen Einrichtungen durch die Bereitstellung von entsprechenden Anschlüssen für Stell- und Steuersignale sowie für Parametrier- und Überwachungsfunktionen. Die technische Umsetzung der Steuerung, wie zum Beispiel das Anlaufverhalten, die Ansteuerung des Antriebs oder die Geräteüberwachung wird durch die Funktionsbaustein-Instanz realisiert und bleibt dem Nutzer verborgen. **PCS 7** bietet bereits eine Vielzahl von leistungsfähigen und getesteten Einzelsteuerfunktionen als Bausteintypen in den leittechnischen Bibliotheken an. Tabelle 2 fasst die Einzelsteuerfunktionen der **PCS 7 Advanced Process Library** [2] zusammen.

Einzelsteuerfunktion	Anwendungsbereich	Objektname
MotL	Ansteuerung von Motoren mit einem Steuersignal (ein/aus) und einer Laufrückmeldung.	FB 1850
MotRevL	Ansteuerung von Motoren mit zwei Drehrichtungen (Linkslauf/Rechtslauf) und maximal zwei Rückmeldungen.	FB 1851
MotSpedL	Ansteuerung von Motoren mit zwei Geschwindigkeiten (langsam/schnell) und maximal zwei Rückmeldungen.	FB 1856
VlvL	Ansteuerung von Steuerventilen mit einem Steuersignal (öffnen/schließen) und Stellungsrückmeldesignalen (offen/geschlossen).	FB 1899
VlvMotL	Ansteuerung von Motorventilen mit zwei Steuersignalen und Stellungsrückmeldesignalen (offen/geschlossen).	FB 1900

Tabelle 2: Die Einzelsteuerfunktionen der **PCS 7 Advanced Prozess Library**

4.5 Schutzmaßnahmen

Bei der Ansteuerung von steuerungstechnischen Einrichtungen sind verschiedene Schutzmaßnahmen zu treffen. Zum einen sind die Einrichtungen selbst vor Fehlern zu schützen. Zum anderen muss der gesteuerte Prozess im Fehlerfall in einen sicheren Zustand überführt und dort solange gehalten werden, bis der Fehlerfall beseitigt ist.

Gerätefehler (zum Beispiel Kabelbruch, Achsbruch) sind steuerungstechnisch nicht vermeidbar. Die Auswirkungen können jedoch durch Redundanzkonzepte minimiert werden. **Prozessfehler** (zum Beispiel Behälterüberlauf, Trockenlauf einer Pumpe) gilt es hingegen direkt durch die Steuerung zu verhindern. Dazu werden entsprechende **Verriegelungen** implementiert. Erkennt die Einzelsteuerfunktion aufgrund der aktuellen Eingangswerte einen gefährlichen Prozesszustand, so wird das gesteuerte Gerät in einen sicheren Zustand überführt (siehe Kapitel ‚Anlagensicherung‘). Das Gerät wird solange in diesem Zustand gehalten, wie der gefährliche Prozesszustand andauert. Üblicherweise sind Verriegelungen mit Hilfe einer **Verriegelungsmatrix** spezifiziert.

Um einen eingetretenen Gerätefehler zu erkennen, führt eine Einzelsteuerfunktion häufig eine **Laufzeitüberwachung** durch. Mit Hilfe bestimmter Sensorinformationen, zum Beispiel von Endlagensensoren in Ventilen, überprüft die Einzelsteuerfunktion, ob die ausgegebenen Stellsignale auch die geforderte Wirkung erzielen. Stehen die gemessenen Werte über einen bestimmten Zeitraum hinweg in Widerspruch zu den ausgegebenen Stellsignalen, so liegt eine Störung vor. Wird ein solcher Laufzeitfehler erkannt, so wird das übergeordnete Leitsystem alarmiert und das gesteuerte Gerät wird deaktiviert. Das Gerät bleibt solange inaktiv, bis der Laufzeitfehler beseitigt und der Alarm quittiert worden ist. Häufig werden zur Erkennung von Gerätefehlern einfache binäre Schutzschalter verwendet.

4.6 Betriebsarten

Steuerungstechnische Einrichtungen werden im Allgemeinen nicht ausschließlich automatisch betrieben. Zeitweise ist es notwendig die Steuerung in der Leitwarte manuell durchzuführen oder das Gerät direkt vor Ort anzusteuern, zum Beispiel für Reparaturarbeiten. Es wird daher zwischen vier grundlegenden Betriebsarten unterschieden:

- **Außer Betrieb:** Das Gerät ist nicht aktiv.
- **Automatikbetrieb:** Die Einzelsteuerfunktion wird von einem übergeordneten Programm automatisch angesteuert.
- **Handbetrieb:** Die Einzelsteuerfunktion wird über eine Bediengraphik des Leitsystems direkt durch den Bediener angesteuert.
- **Vor-Ort-Betrieb:** Der Bediener bedient das Gerät direkt vor Ort, zum Beispiel über ein Bedientableau.

Eine Einzelsteuerfunktion darf sich immer nur in genau einer Betriebsart befinden. Es existieren verschiedene Konzepte, wie die damit verbundene Betriebsartenumschaltung sicher und eindeutig realisiert werden kann. Grundsätzlich lassen sich diese Konzepte unterscheiden zwischen einer Gleichberechtigung der Betriebsarten und einer Betriebsartenhierarchie. Im letztgenannten Fall werden die möglichen Betriebsarten zusätzlich eindeutig priorisiert. Eine angewählte Betriebsart wird in diesem Fall genau dann geändert, wenn das Gerät entweder nicht aktiv ist (Betriebsart: **Außer Betrieb**) oder wenn die gewünschte neue Betriebsart eine höhere Priorität hat als die bereits angewählte.

4.7 Funktionsbaustein-Typen in PCS 7

Funktionsbaustein-Typen werden in **PCS 7** als Bausteintypen bezeichnet und stellen vorgefertigte Programmteile für die Bearbeitung wiederkehrender Funktionen dar. Sie können in CFC-Pläne eingefügt und dort als Instanzen parametrisiert, verschaltet und an die Projekterfordernisse angepasst werden.

Dabei legt der Baustein-Typ die Charakteristik für sämtliche Instanzen dieses Typs fest. Die verwendeten Bausteintypen eines Projekts werden dazu in der Stammdatenbibliothek abgelegt. Wird der dort abgelegte Bausteintyp geändert, so werden die Änderungen unmittelbar von sämtlichen Instanzen übernommen. Dieses Konzept der Typisierung unterstützt das rationelle Engineering durch die Wiederverwendbarkeit und die zentrale Änderbarkeit häufig wiederkehrender Funktionen.



Abbildung 2: Baustein der Einzelsteuerfunktion Motor_Lean

Eine Einzelsteuerfunktion in **PCS 7** modelliert eine steuerungstechnische Einrichtung und stellt die gesamte Ansteuerlogik bereit. Abbildung 2 beschreibt am Beispiel der Einzelsteuerfunktion **Motor_Lean** den grundsätzlichen Aufbau des entsprechenden Motorbausteins.

Der Baustein bietet darüber hinaus die folgenden Funktionen:

– **Bedienen, Beobachten, Melden**

Über einen Anzeige- und Bedienbereich können Prozess- und Sollwerte bedient und beobachtet sowie Bedienberechtigungen und Wartungsfreigaben gesteuert werden. Allgemeine und instanzspezifische Meldungen geben Aufschluss über den Geräte- und Prozessstatus.

– **Steuerung von Signalen**

Steuersignale können statisch oder gepulst ausgegeben werden. Der Signalstatus, also die Qualität der Stellsignale, wird überwacht. Es können interne und externe Sollwerte sowie simulierte Werte vorgegeben, als auch Rampen oder Totzonen eingestellt werden.

– **Überwachung**

Der Baustein kann Grenzwerte überwachen und bei Grenzwertverletzungen entsprechende Warnungen oder Alarmer generieren. Außerdem können Rückmeldungen von Stellsignalen überwacht werden.

– **Verriegelungsfunktionen**

Der Baustein ermöglicht eine einfache Einschaltfreigabe sowie eine Verriegelung mit und ohne Rücksetzen. Er implementiert eine Motorschutzfunktion, die den Motor bei thermischer Überlast abschalten kann. Zusätzlich ist für Motoren ein Schnellstopp verfügbar, der in allen Betriebsarten und -zuständen höchste Priorität hat. Im Fall einer Verriegelung wird das Gerät automatisch in einen energielosen Zustand und so in eine definierte Sicherheitsstellung überführt.

– **Betriebszustandsauswahl**

Die eingangs vorgestellten Betriebsarten Vor-Ort-Betrieb, Automatikbetrieb, Handbetrieb und Außer Betrieb stehen für sämtliche Einzelsteuerfunktionen in **PCS 7** zur Verfügung. Sie sind in der angegebenen Reihenfolge absteigend priorisiert, Automatik- und Handbetrieb besitzen die gleiche Priorität. Darüber hinaus ist es möglich, den Baustein über verschaltbare Eingangsparameter unabhängig von der aktuell anstehenden Ansteuerung in einen anderen Betriebszustand zu versetzen (**Erzwingen** bzw. **Forcen** von Betriebszuständen).

– **Bildbausteine mit verschiedenen Sichten**

Bildbausteine bieten für jeden Bausteintyp ein entsprechendes Bausteinsymbol und je nach Anwendungsfall entsprechende Sichten an. Typische Bildbausteine sind zum Beispiel das Bausteinsymbol selbst, die Parametersicht von Motoren und Ventilen oder die Grenzwertsicht von Motoren.

Diese Aufzählung zeigt deutlich die Komplexität und den Funktionsumfang einer üblichen Einzelsteuerfunktion. Die Anzahl der verfügbaren Ein- und Ausgänge bei diesen Bausteintypen ist dementsprechend groß. So besitzt die Einzelsteuerfunktion **Motor_Lean** insgesamt 53 Anschlüsse. Um die Komplexität des Programmentwurfs dennoch gering zu halten, ist es möglich, nicht benötigte Ein- oder Ausgänge zu verbergen. Die Einzelsteuerfunktionen in **PCS 7** verwenden zudem ein einheitliches und durchgängiges Schema für die Bezeichnung der Ein- und Ausgänge.

Die Einzelsteuerfunktionen in **PCS 7** bieten einen großen Funktionsumfang und garantieren eine konstant hohe Qualität und Zuverlässigkeit der verwendeten Algorithmen. Sämtliche Bausteintypen wurden umfangreich getestet und haben sich bereits industriell bewährt. Damit verkürzt sich der Aufwand für die Entwicklung leistungsfähiger, qualitativ hochwertiger Lösungen erheblich.

4.8 Literatur

- [1] Seitz, M. (2008): Speicherprogrammierbare Steuerungen. Hanser Fachbuchverlag
- [2] SIEMENS (2018): SIMATIC Prozessleitsystem PCS 7 Advanced Process Library (V9.0 SP2). A5E39148920-AC. (support.automation.siemens.com/WW/view/de/109760968)

5 Aufgabenstellung

Ein Pumpenmotor zum Ablassen der Flüssigkeit aus dem Reaktor R001 soll als erstes Programm im Continuous Function Chart (CFC) erstellt werden. Der Pumpenmotor hat einen Ausgang zum Ansteuern der Pumpe und eine Rückmeldung zur Kontrolle, ob die Pumpe auch läuft.

Symbol	Adresse	Datentyp	Symbolkommentar
A1.T2.A1T2S003.SO+.O+	E 1.3	BOOL	Pumpe Ablass Reaktor R001 Rückmeldung ein
A1.T2.A1T2S003.SV.C	A 3.4	BOOL	Pumpe Ablass Reaktor R001 Stellsignal

Tabelle 3: Zuordnungsliste

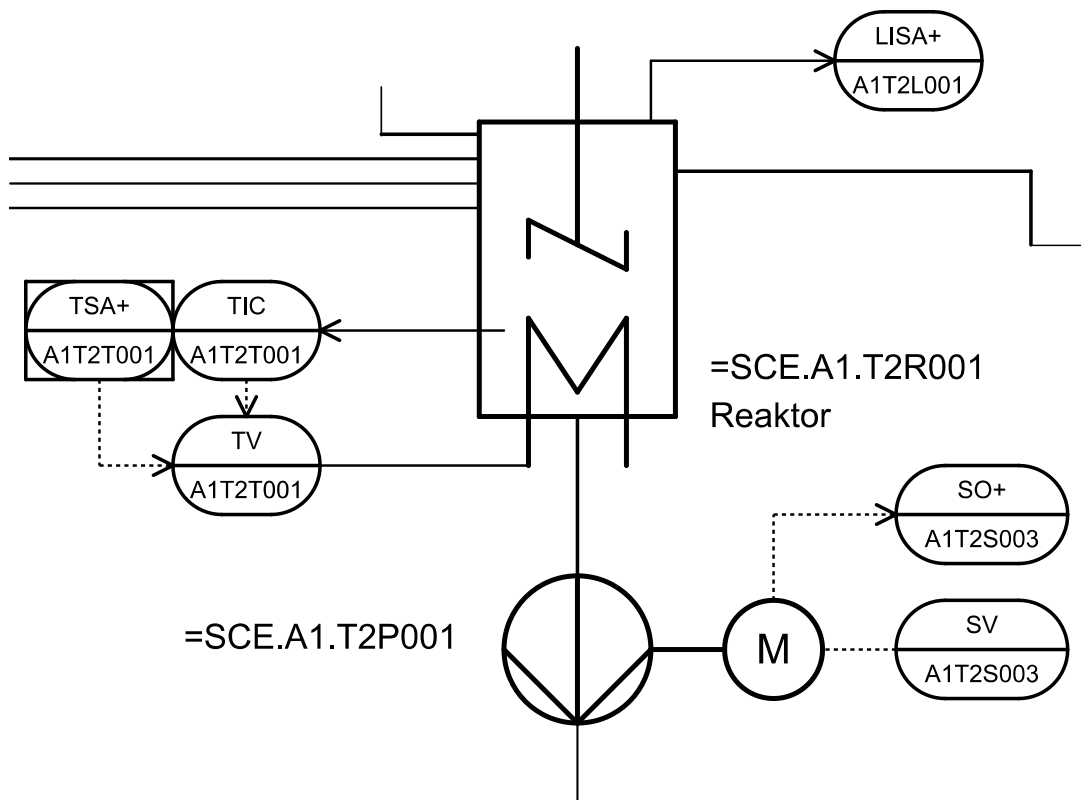


Abbildung 3: Ausschnitt aus dem R&I-Fließbild

6 Planung

Neben den zwei aus Tabelle 3 der Aufgabenstellung bekannten Symbolen gibt es in der Anlage noch viele weitere Symbole. Damit diese nicht alle per Hand eingetragen werden müssen, kann die bereits vorbereitete Datei p01-04-symbols-r1905-de.dif importiert werden. Sie ist als Ergebnis einer guten Anlagenplanung aus dem R&I generierbar. Da ein Motor (siehe Abbildung 4) realisiert werden soll, wird bei der Erstellung des Programms ein vorgefertigter Plan ‚Motor_Lean‘

aus einer **PCS 7**-Bibliothek verwendet. Dieser wird in die projekteigene Stammdatenbibliothek kopiert und dort angepasst. Anschließend wird das Programm in die SPS-Simulation geladen und getestet.

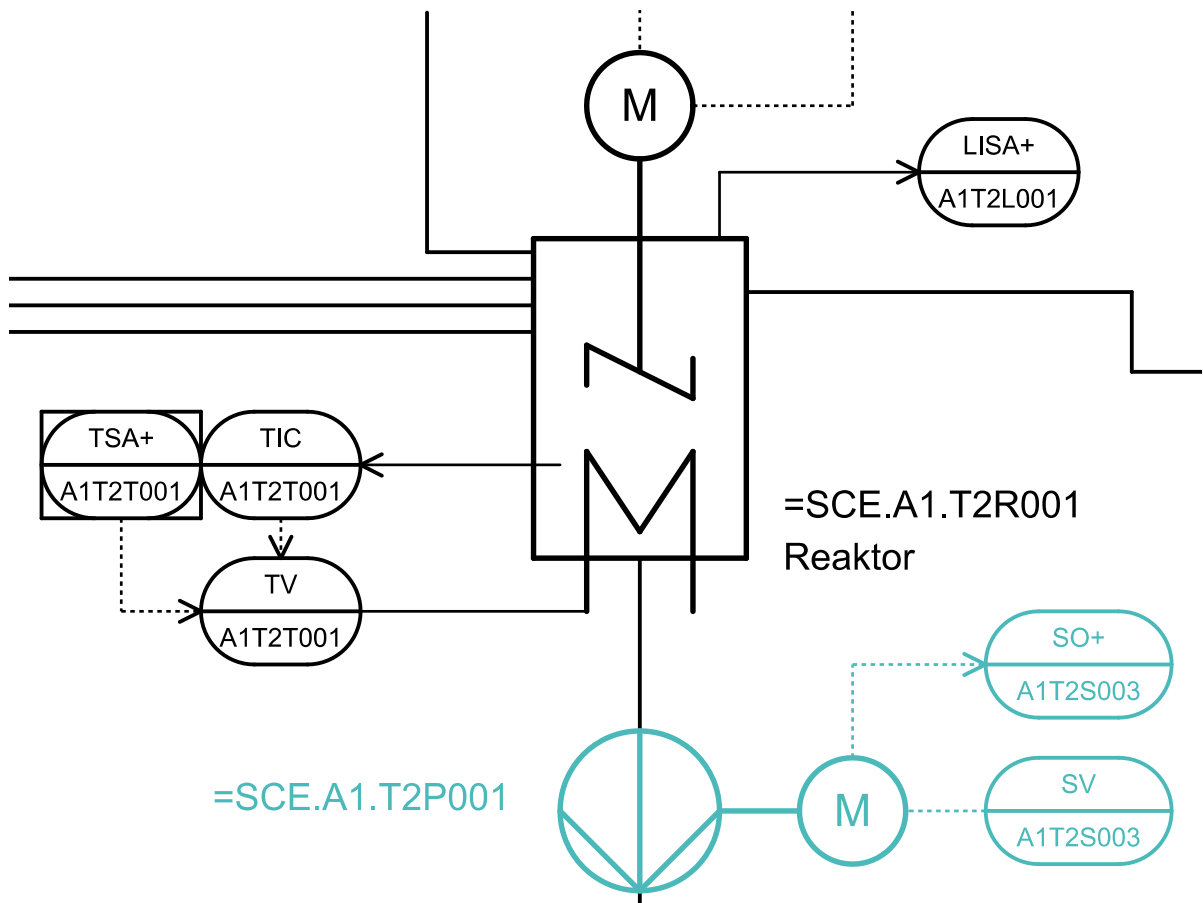


Abbildung 4: Zu bearbeitender Teil des R&I-Fließbilds

7 Lernziel

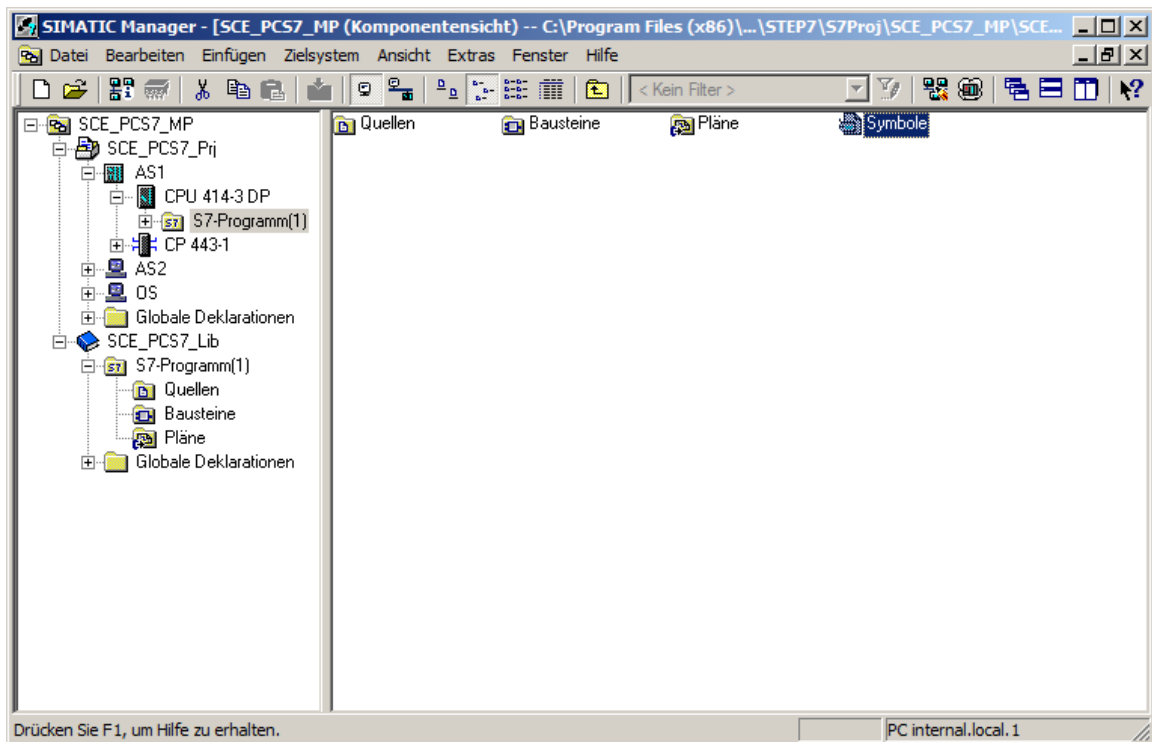
In diesem Kapitel lernen die Studierenden:

- Das Anlegen und importieren von Symbolen via Symboltabelle
- Die Verwendung von Stammdatenbibliotheken
- Das CFC-Pläne anlegen und bearbeiten
- Das Projekt zentral übersetzen und laden
- Das Testen des Programms mit Hilfe der Steuerfunktionen im CFC

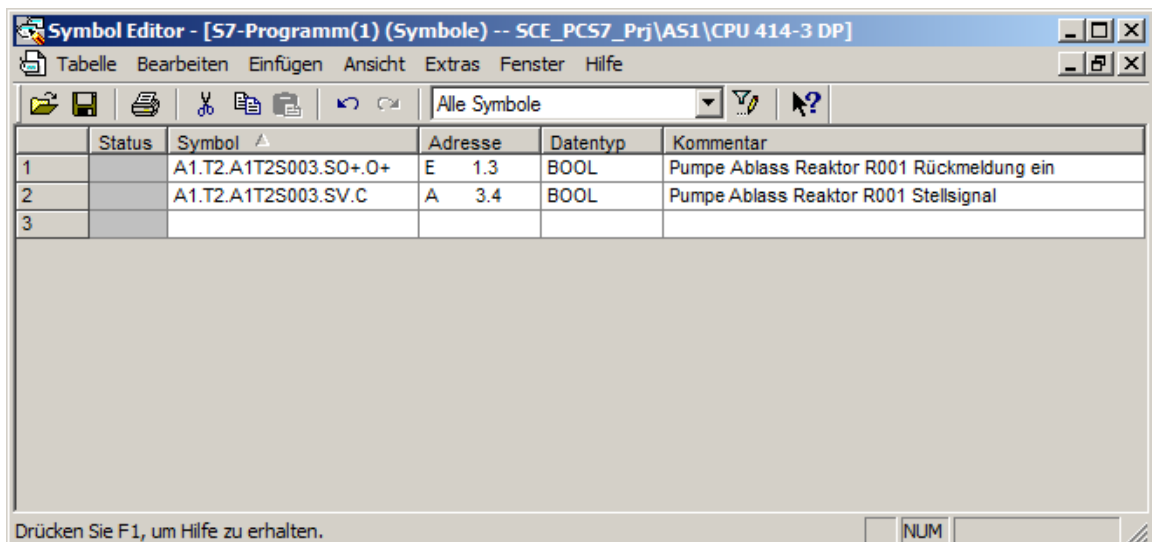
8 Strukturierte Schritt-für-Schritt-Anleitung

8.1 Symboltabelle bearbeiten

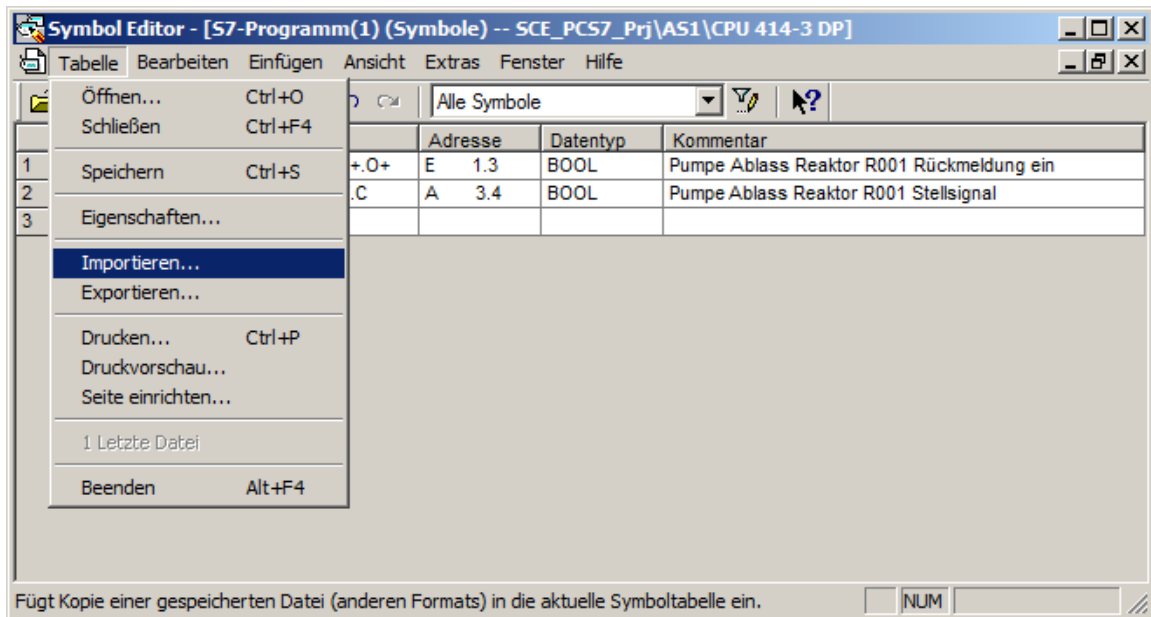
1. Bevor Sie mit der Programmierung der Einzelsteuerfunktion für den Pumpenmotor beginnen, werden die Symbole für die globalen Variablen angelegt. Sie wählen hierzu die Komponentensicht im SIMATIC Manager, markieren den Ordner ‚S7-Programm(1)‘ und öffnen mit einem Doppelklick auf Symbole die Symboltabelle. (→ SIMATIC Manager → Ansicht → Komponentensicht → AS1 → CPU 414-3 DP → S7-Programm(1) → Symbole)




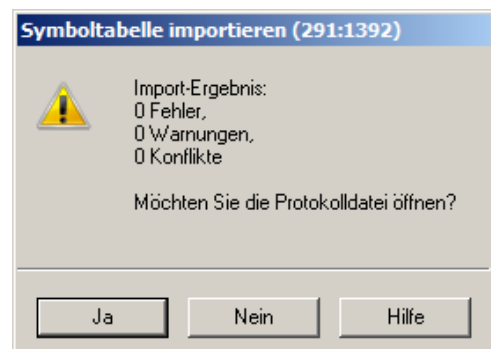
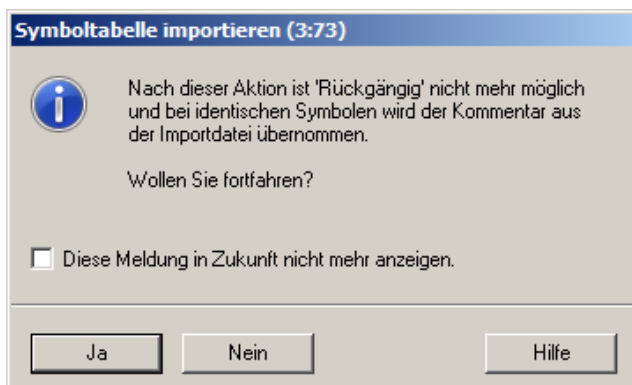
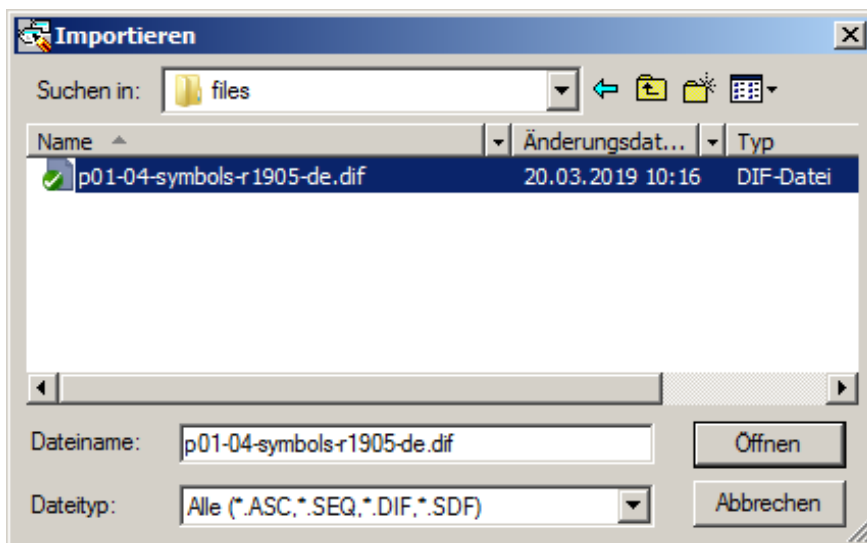
2. In der Symboltabelle können anschließend Symbol und Symbolkommentar zu jedem Operanden festgelegt werden.



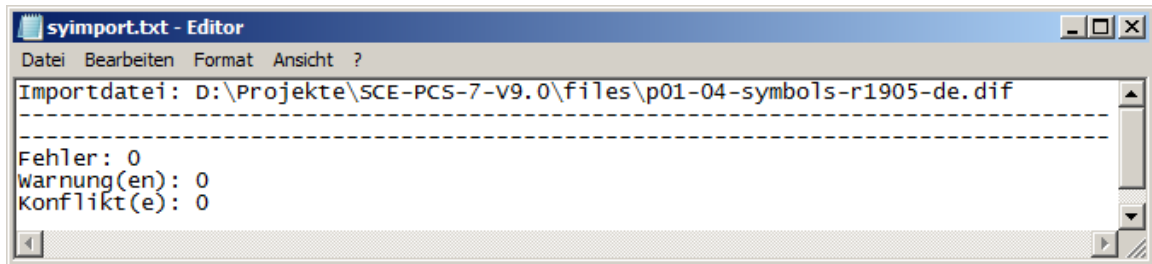
3. Falls vorhanden, kann auch der Inhalt für die komplette Symboltabelle im *.dif- Format importiert werden. In diesem Fall wird die importierte Tabelle in die bereits vorhandene Tabelle integriert. (→ Tabelle → Importieren)





4. Nun erfolgt die Auswahl der Quelldatei im ‚Data Interchange Format‘ (*.DIF) (→ p01-04-symbols-r1905-de.dif → Öffnen → Ja → Ja → )



5. Es wird ein Protokoll angezeigt.



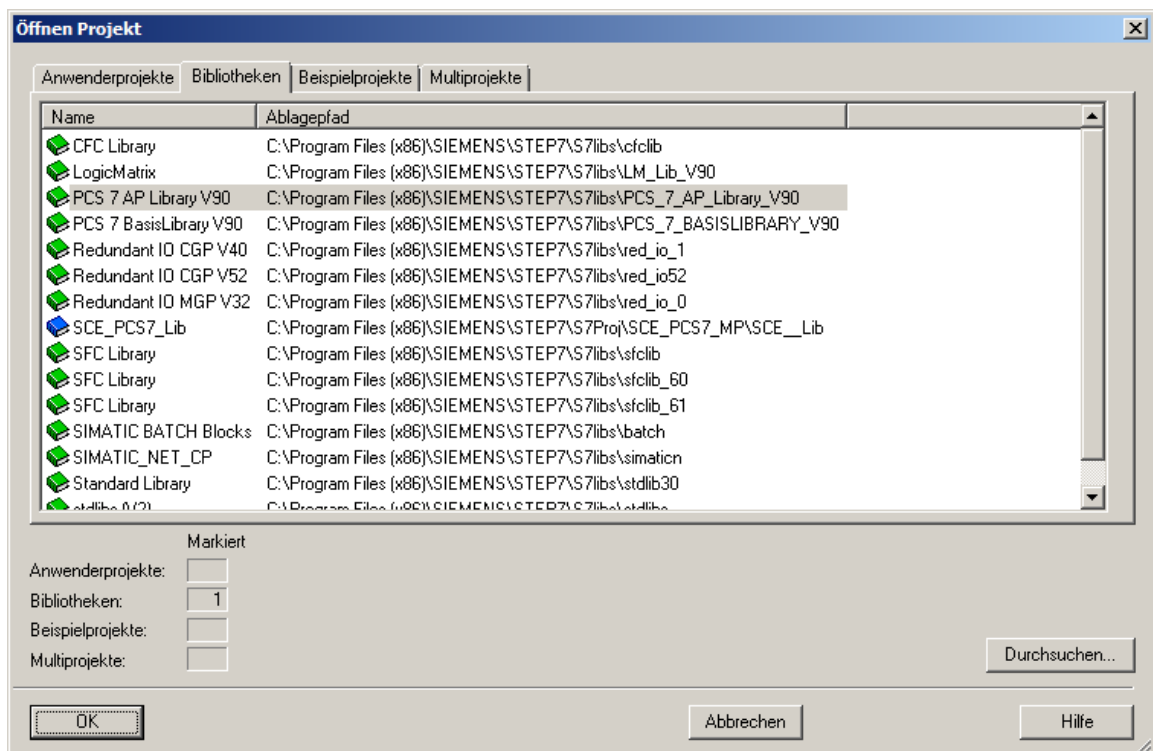
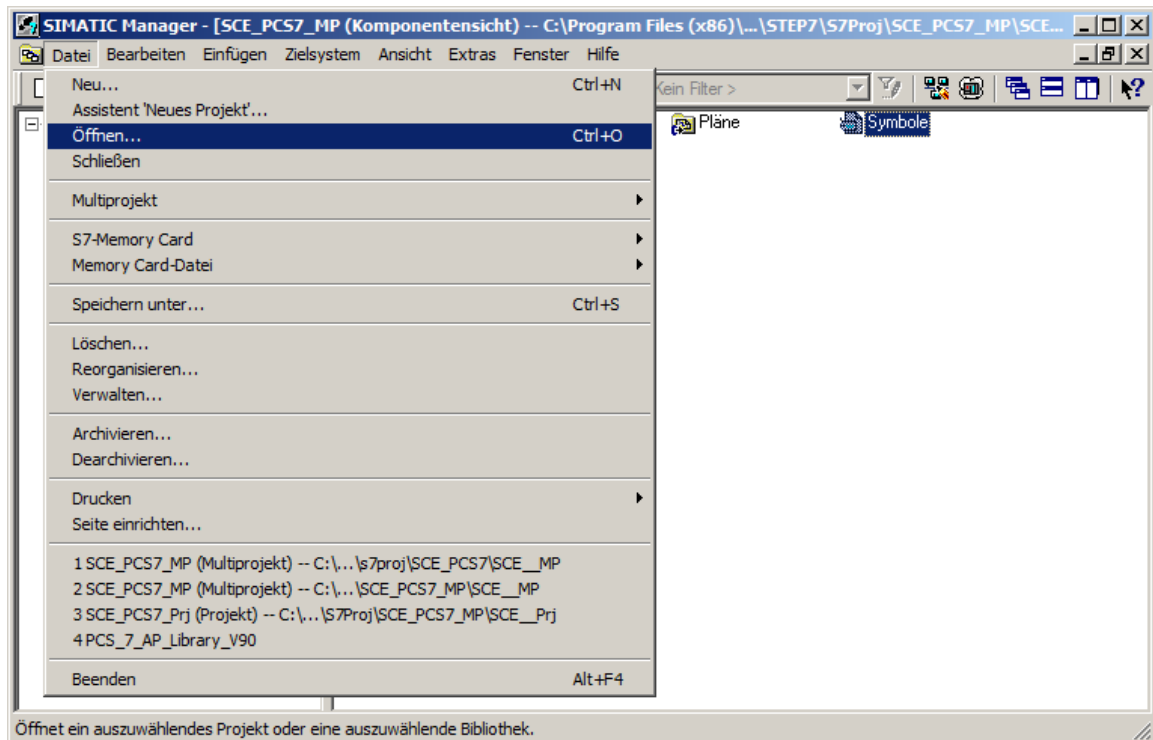
6. Die fertige Symboltabelle muss vor dem Schließen noch gespeichert werden.
(→ Speichern  → )

The screenshot shows the 'Symbol Editor' window for 'S7-Programm(1) (Symbole) -- SCE_PCS7_Prj\AS1\CPU 414-3 DP'. The menu bar includes 'Tabelle', 'Bearbeiten', 'Einfügen', 'Ansicht', 'Extras', 'Fenster', and 'Hilfe'. The toolbar contains icons for file operations and editing. The main area displays a table of symbols. The status bar at the bottom indicates 'Speichert die aktuelle Symboltabelle.' and 'NUM'.

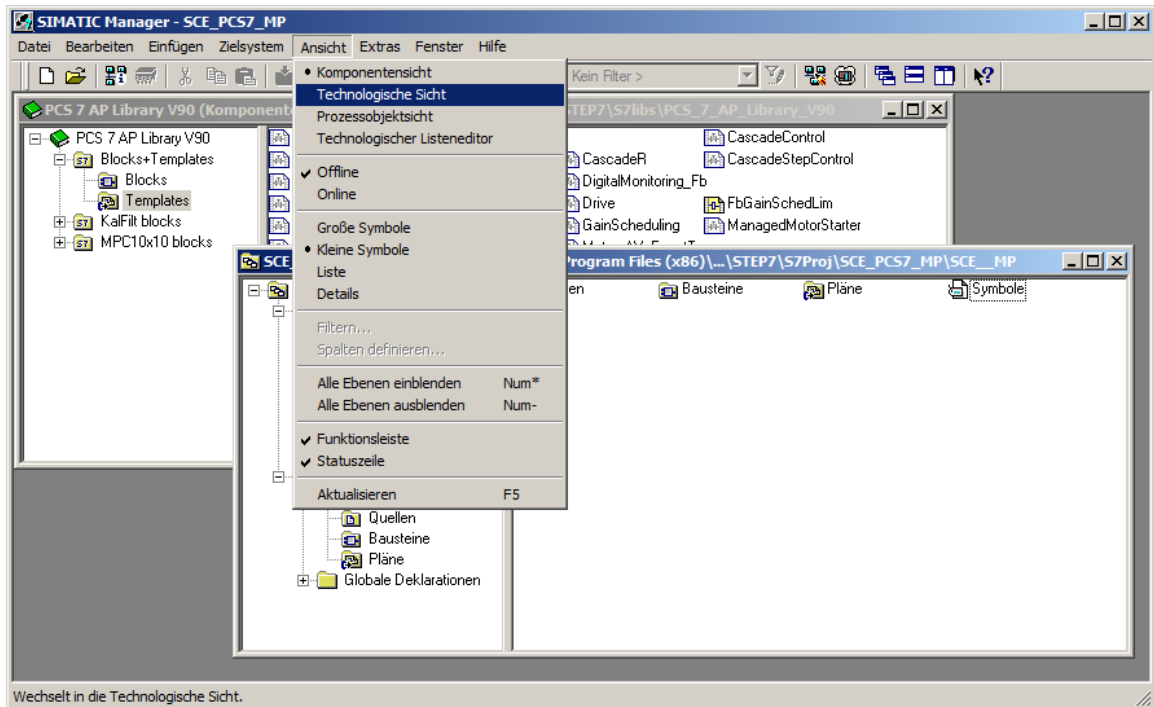
Status	Symbol	Adresse	Datentyp	Kommentar
1	A1.T2.A1T2S003.SO+.O+	E 1.3	BOOL	Pumpe Ablass Reaktor R001 Rückmeldung ein
2	A1.T2.A1T2S003.SV.C	A 3.4	BOOL	Pumpe Ablass Reaktor R001 Stellsignal
3	A1.T1.A1T1X001.XV.C	A 0.0	BOOL	Auf/Zu-Ventil Zufluss Edukttank B001 Stellsignal
4	A1.T1.A1T1X002.XV.C	A 0.1	BOOL	Auf/Zu-Ventil Zufluss Edukttank B002 Stellsignal
5	A1.T1.A1T1X003.XV.C	A 0.2	BOOL	Auf/Zu-Ventil Zufluss Edukttank B003 Stellsignal
6	A1.T1.A1T1X004.XV.C	A 0.3	BOOL	Auf/Zu-Ventil Ablass Edukttank B001 Stellsignal
7	A1.T1.A1T1X005.XV.C	A 0.4	BOOL	Auf/Zu-Ventil Ablass Edukttank B002 Stellsignal
8	A1.T1.A1T1X006.XV.C	A 0.5	BOOL	Auf/Zu-Ventil Ablass Edukttank B003 Stellsignal
9	A1.T3.A1T3X001.XV.C	A 0.6	BOOL	Auf/Zu-Ventil Zufluss Produkttank B001 Stellsignal
10	A1.T3.A1T3X002.XV.C	A 0.7	BOOL	Auf/Zu-Ventil Zufluss Produkttank B002 Stellsignal
11	A1.T4.A1T4X001.XV.C	A 1.0	BOOL	Auf/Zu-Ventil Spülwasserzufluss Reaktor R001...
12	A1.T4.A1T4X002.XV.C	A 1.1	BOOL	Auf/Zu-Ventil Spülwasserzufluss Reaktor R002...
13	A1.T4.A1T4X003.XV.C	A 1.2	BOOL	Auf/Zu-Ventil Spülwasserablass Reaktor R001 ...
14	A1.T4.A1T4X004.XV.C	A 1.3	BOOL	Auf/Zu-Ventil Spülwasserablass Reaktor R002 ...
15	A1.T2.A1T2X001.XV.C	A 1.4	BOOL	Auf/Zu-Ventil Zufluss Reaktor R001 aus Edukta...
16	A1.T2.A1T2X002.XV.C	A 1.5	BOOL	Auf/Zu-Ventil Zufluss Reaktor R001 aus Edukta...
17	A1.T2.A1T2X003.XV.C	A 1.6	BOOL	Auf/Zu-Ventil Zufluss Reaktor R001 aus Edukta...
18	A1.T2.A1T2X004.XV.C	A 1.7	BOOL	Auf/Zu-Ventil Zufluss Reaktor R002 aus Edukta...
19	A1.T2.A1T2X005.XV.C	A 2.0	BOOL	Auf/Zu-Ventil Zufluss Reaktor R002 aus Edukta...
20	A1.T2.A1T2X006.XV.C	A 2.1	BOOL	Auf/Zu-Ventil Zufluss Reaktor R002 aus Edukta...
21	A1.T2.A1T2X007.XV.C	A 2.2	BOOL	Auf/Zu-Ventil Zufluss Reaktor R002 aus Reakto...
22	A1.T2.A1T2X008.XV.C	A 2.3	BOOL	Auf/Zu-Ventil Zufluss Reaktor R001 aus Reakto...
23	A1.T1.A1T1S001.SV.C	A 3.0	BOOL	Pumpe Ablass Edukttank B001 Stellsignal
24	A1.T1.A1T1S002.SV.C	A 3.1	BOOL	Pumpe Ablass Edukttank B002 Stellsignal
25	A1.T1.A1T1S003.SV.C	A 3.2	BOOL	Pumpe Ablass Edukttank B003 Stellsignal
26	A1.T4.A1T4S001.SV.C	A 3.3	BOOL	Spülpumpe Stellsignal
27	A1.T2.A1T2S004.SV.C	A 3.5	BOOL	Pumpe Ablass Reaktor R002 Stellsignal
28	A1.T2.A1T2H007.HO+-.O+	A 4.0	BOOL	Reaktor R001 Rühren Statuswert

8.2 Baustein Motor_Lean aus der PCS 7 AP Library V90 in Projektbibliothek einfügen

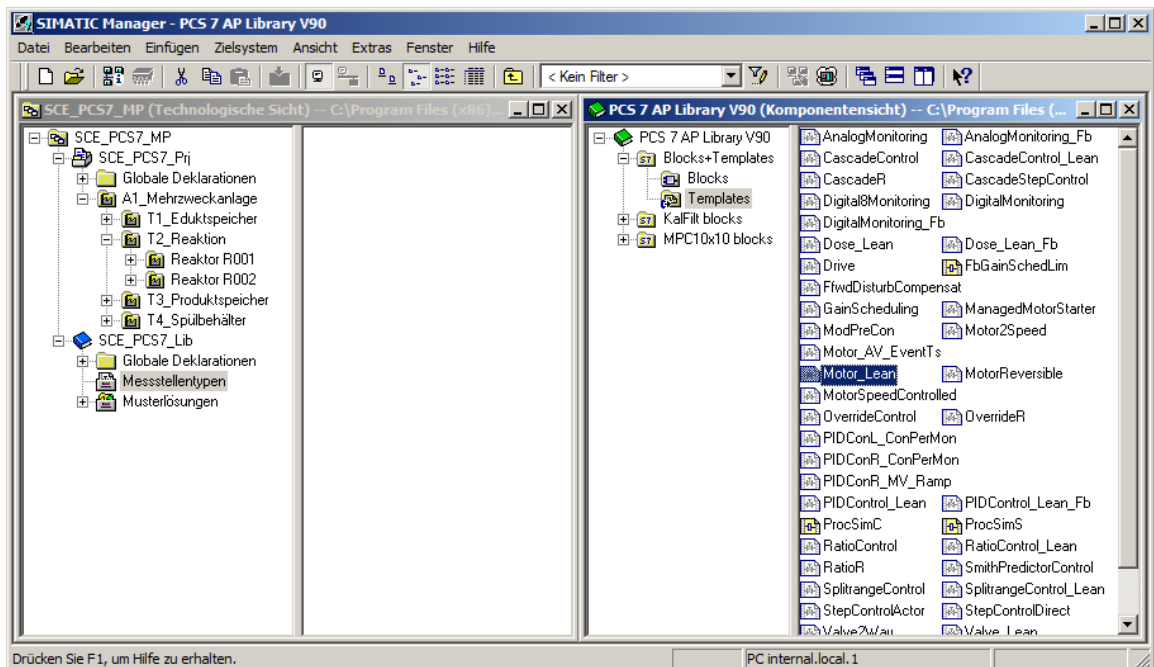
1. PCS 7 stellt in umfangreichen Bibliotheken eine Vielzahl an fertigen Bausteinen und auch vorgefertigte Pläne, sogenannte Templates, zur Verfügung. Für den Pumpenmotor nutzen Sie ein solches Template. Dafür öffnen Sie die PCS 7 AP Library V9.0. (→ Datei → Öffnen → Bibliotheken → PCS 7 AP Library V90 → OK)



- Um dieses Template aus der Bibliothek in das Projekt zu ziehen, gehen Sie auf die Technologische Sicht des Projekts. (→ Ansicht → Technologische Sicht)

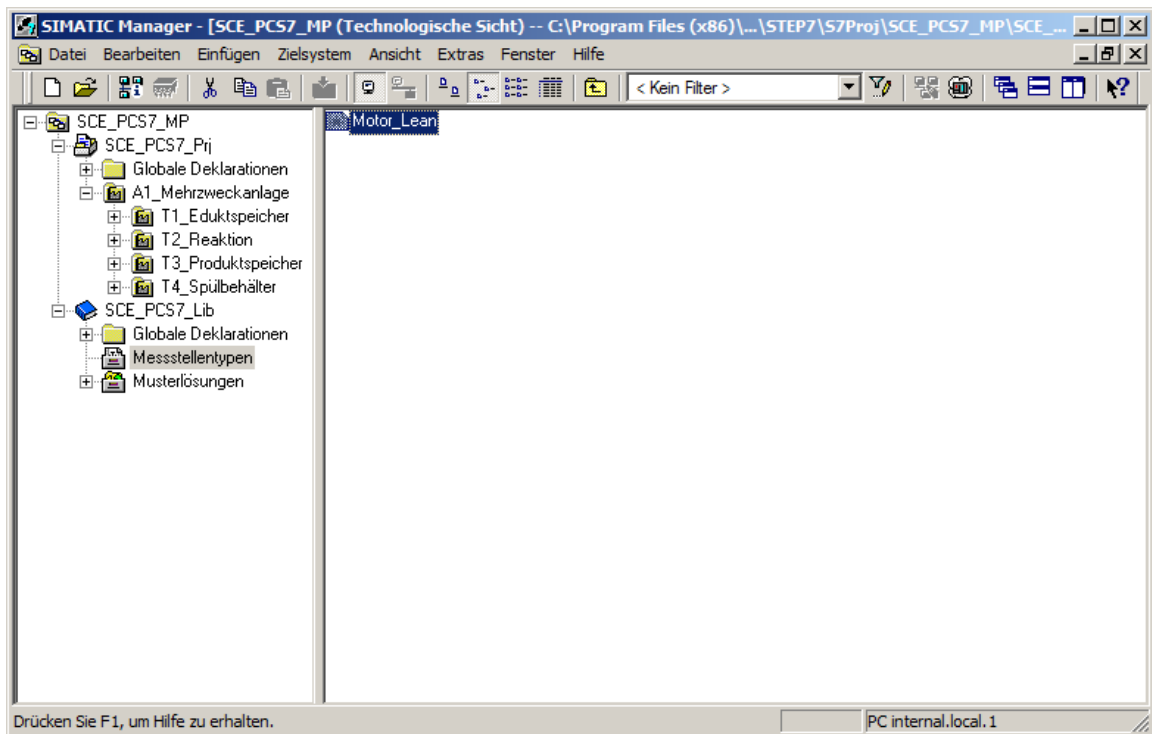


- Per Drag&Drop wird Motor_Lean (Bibliothek, Templates) nach Messstellentypen (Stammdatensbibliothek) verschoben. (→ PCS 7 Library → Blocks+Templates → Templates → Motor_Lean nach Technologische Sicht → SCE_PCS7_Lib → Messstellentypen ziehen)




8.3 Motor_Lean zentral bearbeiten

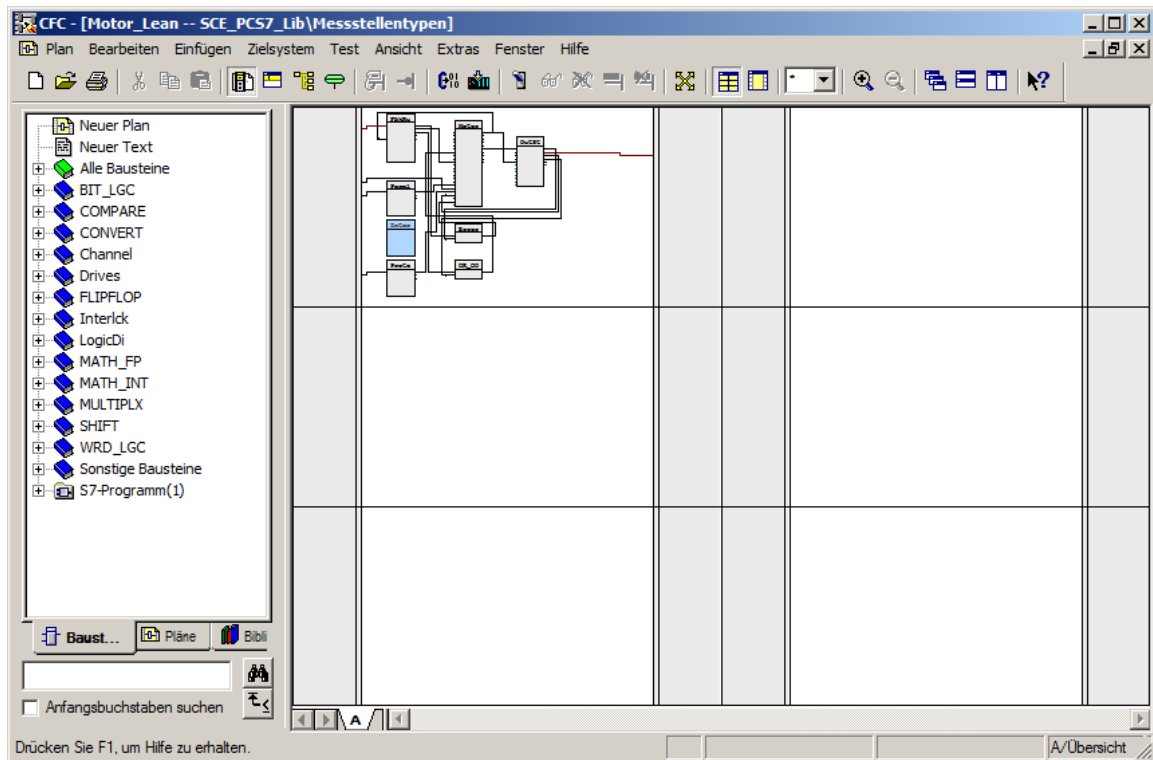
1. Jetzt soll eine Änderung zentral für alle Messstellen des Typs ‚Motor_Lean‘ erfolgen. Dies geschieht, indem der CFC-Plan ‚Motor_Lean‘ aus der Stammdatenbibliothek per Doppelklick geöffnet wird. (→ Motor_Lean)



Hinweis:

- CFC steht für *Continuous Function Chart* und ist eine grafische Programmiersprache für die Beschreibung kontinuierlicher Vorgänge. Im CFC werden vorgefertigte Bausteine platziert, parametrisiert und verschaltet. So erstellt der Programmierer eine Gesamt-Software-Struktur für die Steuerung und Regelung einer Maschine.

2. Ein CFC-Plan besteht aus Teilplänen mit jeweils sechs Blättern. In der Übersicht werden alle sechs Blätter mit ihren grauen Randleisten angezeigt. An den Randleisten sehen Sie links die kommenden Signale und rechts die gehenden Signale eines Blattes. Mit einem Doppelklick auf ein Blatt können Sie in die Blattsicht wechseln. (→ Übersicht  → Doppelklick auf erstes Blatt)



Hinweis:

- Mit den Registern in der unteren Leiste können Sie zwischen den Teilplänen (maximal A-Z) wechseln. Hier existiert jedoch zuerst nur ein Teilplan A.

4. Die allgemeinen Eigenschaften wie Bedien- und Beobachtbarkeit wollen Sie nicht verändern, deshalb wechseln Sie zu den Anschlüssen. (→ Anschlüsse)

The screenshot shows a software dialog box titled "Eigenschaften - Baustein -- Motor_Lean\Motor". It has two tabs: "Allgemein" (selected) and "Anschlüsse". The "Allgemein" tab contains the following fields and options:

- Typ: MotL
- Name: Motor
- Kommentar: Motor - Large
- Eingänge: 94
- Interner Bezeichner: FB1850
- Instanz-DB: DB66
- Name (Header): MotL
- Familie: Drives
- Autor: AdvLib90
- Einbauen in OB/Ablaufebenen: OB100 [Neustart]
- Bedien- und beobachtbar: (with a "Bedienen und Beobachten..." button)
- Bausteinsymbol erzeugen: (with an empty text field)
- MES-relevant:
- Spezielle Eigenschaften: (with a "Meldungen..." button)
- Rücklesen erlaubt:

At the bottom of the dialog are buttons for "OK", "Drucken", "Abbrechen", and "Hilfe".

5. Die Anschlüsse werden in einer Tabelle zusammen mit einer Vielzahl an einstellbaren Eigenschaften dargestellt. Die wichtigsten Eigenschaften werden Sie im Folgenden noch kennen lernen. In dem Baustein ‚MotL‘ wollen Sie hier lediglich bei ‚AutModOp‘ die Unsichtbarkeit löschen. Dadurch wird dieser Anschluss in dem Blatt dargestellt. Mit ‚OK‘ verlassen Sie die Eigenschaften. (→ AutModOp → Unsichtbar → → OK)

Eigenschaften - Baustein -- Motor_Lean\Motor														
Allgemein Anschlüsse														
#	Name	I/O	Typ	W...	Ve...	Fo...	Fo...	Fo...	SF...	Te...	Ko...	Unsichtbar	Fü...	Ar...
296	AutAct	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1=...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
298	AutAct.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
297	AutAct.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	AutModLi	IN	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1=...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
17	AutModLi.ST	IN	...	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	AutModLi.Value	IN	...	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	AutModOp	IN	BOOL	0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1=...	<input type="checkbox"/>	<input type="checkbox"/>	K.
325	AutoST	OUT	BYTE	16...		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	W...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
238	AV	IN	ANY			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Pr...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
114	AV_AH_Lim	IN	REAL	95.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
119	AV_AL_Lim	IN	REAL	5.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
120	AV_Hyst	IN	REAL	1.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
318	AV_OpScale	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
319	AV_OpScale.High	10...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Hi...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
320	AV_OpScale.Low	0.0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Lo...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
314	AV_Out	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Ad...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
316	AV_Out.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
315	AV_Out.Value	0.0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
116	AV_TH_Lim	IN	REAL	85.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
117	AV_TL_Lim	IN	REAL	15.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
317	AV_Unit	OUT	INT	0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	En...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
115	AV_WH_Lim	IN	REAL	90.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
118	AV_WL_Lim	IN	REAL	10.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
99	BatchEn	IN	BOOL	0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	En...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
100	BatchID	IN	D...	16...		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Cu...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
101	BatchName	IN	ST...	"		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Cu...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
53	ByProt	IN	BOOL	0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	By...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	K.
107	CSF	IN	ST...		Mo...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Co...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Hinweis:

- Durch einen Klick mit der Maus auf die Überschrift einer Eigenschaft kann die Darstellung nach dieser Spalte alphabetisch sortiert werden.

4. Bei den allgemeinen Eigenschaften wird der Name des Bausteins geändert.
(→ Pumpe_A1T2S003 → OK)

Eigenschaften - Baustein -- A1T2S003\Motor

Allgemein | Anschlüsse

Typ: MotL Bausteingruppe:

Name:

Kommentar:

Eingänge: 94

Interner Bezeichner: FB1850

Instanz-DB: DB66

Name (Header): MotL

Familie: Drives

Autor: AdvLib90

Bedien- und beobachtbar

Bausteinsymbol erzeugen:

MES-relevant

Einzubauen in OB/Ablaufebenen:

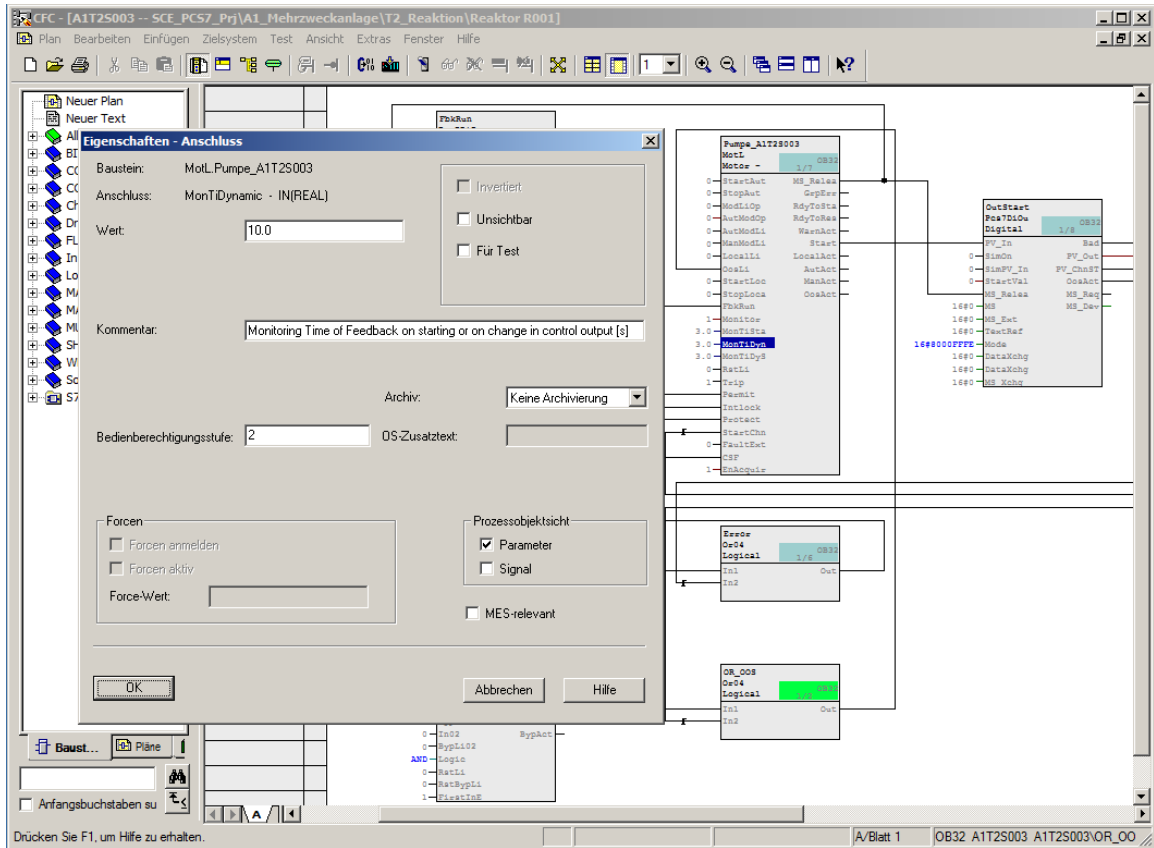
OB100 [Neustart]

Spezielle Eigenschaften

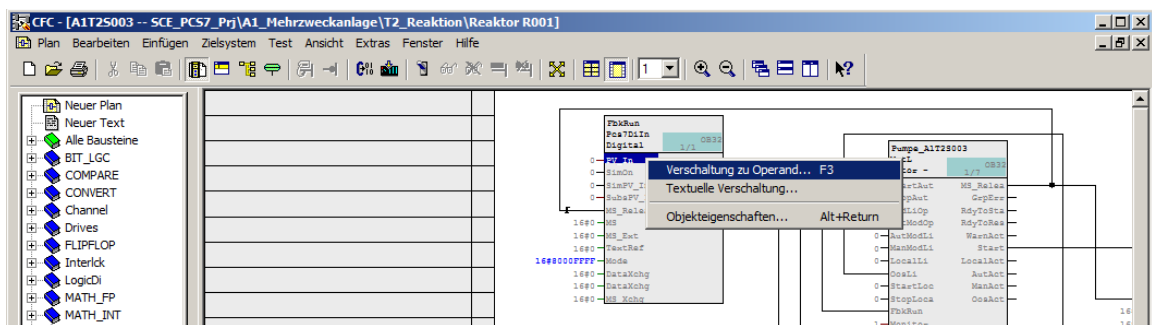
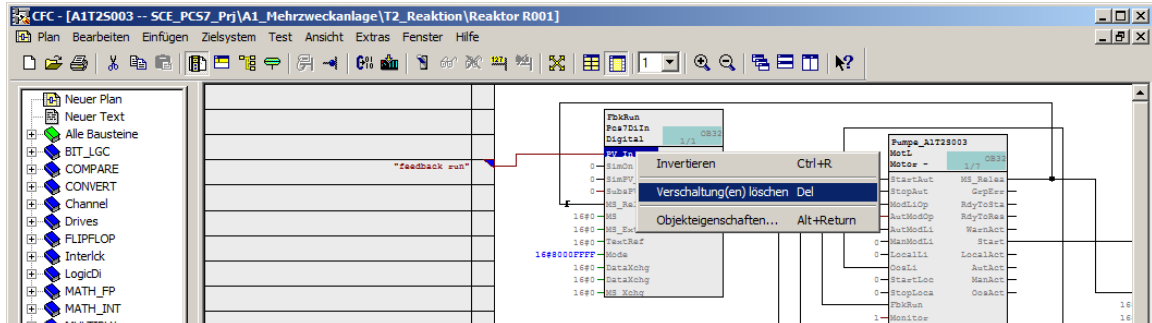
Rücklesen erlaubt

OK Drucken Abbrechen Hilfe

5. Nun wollen Sie die Zeit für die Rückmeldungsüberwachung nach erfolgreicher Bedienung des Motorbausteins auf 10.0 Sekunden ändern. Dafür wird der Eigenschaftsdialog für den Eingang ‚MonTiDyn‘ mit einem Doppelklick geöffnet und bei Wert 10.0 eingetragen. Mit ‚OK‘ verlassen Sie den Dialog wieder. (→ MonTiDyn → Wert: 10.0 → OK)



6. Daraufhin erfolgt noch die Verschaltung der Rückmeldung zum Eingangsoperanden. Dies geschieht, indem der Eingang ‚PV_In‘ am Baustein ‚FbkRun‘ (PCS7DiIn) mit der rechten Maustaste angeklickt und hier die Verschaltung zum Operanden gewählt wird. (→ FbkRun → PV-In → Verschaltung(en) löschen → Verschaltung zum Operand)



Hinweise:

- Der Baustein PCS7DiIn ist ebenso wie PCS7DiOu ein Treiberbaustein für die Kopplung zur Peripherie der SPS. Wird am Eingabe-Baustein der Wert ‚PV_In‘ bzw. am Ausgabe-Baustein der Wert ‚PV_Out‘ mit einem Operanden verschaltet, der auch in der Hardwarekonfiguration projiziert ist, so wird später beim Übersetzungslauf automatisch der Eingang MODE mit Daten versorgt.
- Damit das geschieht, muss später beim Übersetzungslauf ‚Baugruppentreiber erzeugen‘ angewählt werden.

7. Der Nutzer wählt den Operanden, welcher rückmeldet ob die Pumpe läuft, komfortabel direkt aus der Symboltabelle aus. (→ A1.T2.A1T2S003.SO+.O+)

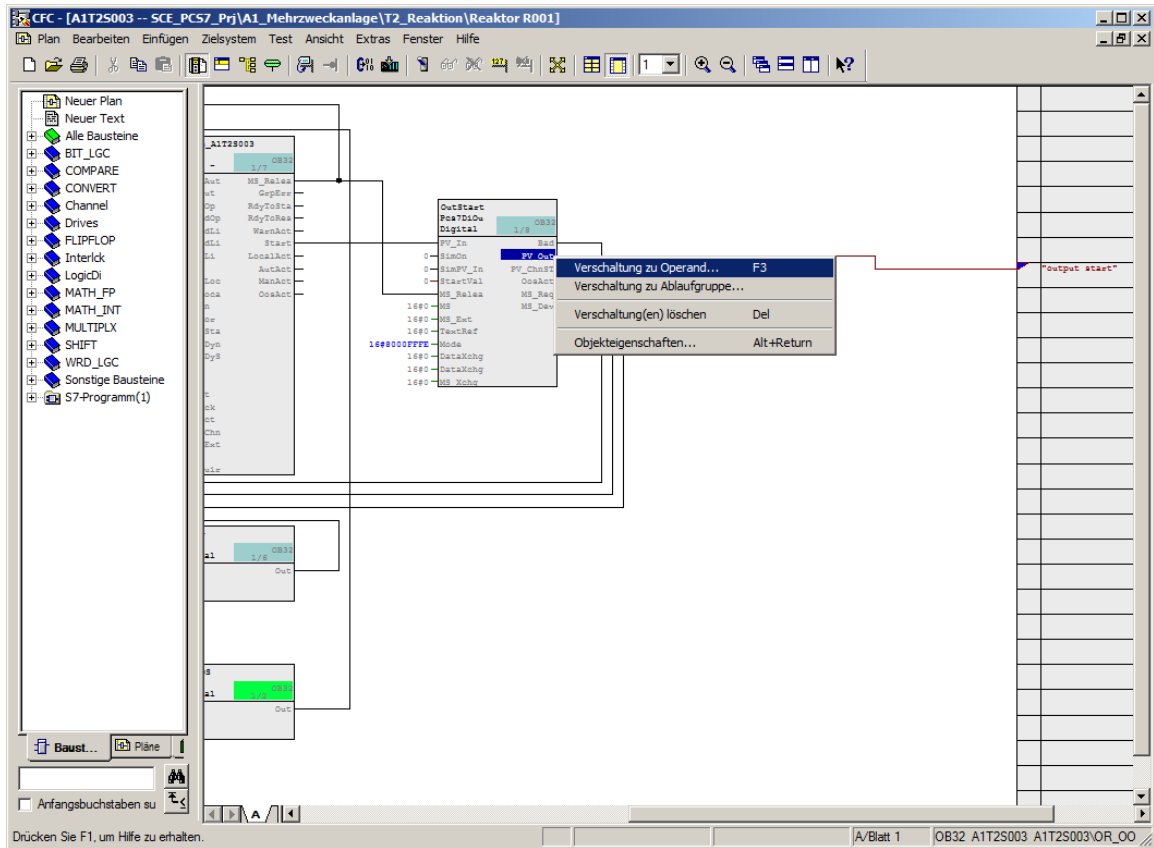
The screenshot displays the Siemens STEP 7 interface for a project named 'A1T2S003'. The main window shows a ladder logic network with the following components:

- Network 1:** An AND gate with inputs 'Interlock' and 'Protect'. The output is connected to a coil labeled 'OR_OOS'.
- Network 2:** A coil labeled 'OR_OOS'.

The symbol table, titled '"A1.T2.A1T2S003.SO+.O+"', lists the following signals:

Symbol	Variable	Value	Comment
A1.T2.A1T2H015.HS-.STOP	BOOL	E 7.7	Reaktor R001 Umfüllen nach Reaktor R002 Stoppen
A1.T2.A1T2H016.HO+-.O+	BOOL	A 5.4	Reaktor R002 Umfüllen nach Reaktor R001 Statuswert
A1.T2.A1T2H016.HS+.START	BOOL	E 5.6	Reaktor R002 Umfüllen nach Reaktor R001 Beginnen
A1.T2.A1T2H016.HS-.STOP	BOOL	E 5.7	Reaktor R002 Umfüllen nach Reaktor R001 Stoppen
A1.T2.A1T2S001.SO+-.O+	BOOL	E 2.0	Rührer Reaktor R001 Rückmeldung ein
A1.T2.A1T2S001.SV.C	BOOL	A 4.6	Rührer Reaktor R001 Stellsignal
A1.T2.A1T2S002.SO+-.O+	BOOL	E 2.1	Rührer Reaktor R002 Rückmeldung ein
A1.T2.A1T2S002.SV.C	BOOL	A 5.6	Rührer Reaktor R002 Stellsignal
A1.T2.A1T2S003.SO+.O+	BOOL	E 1.3	Pumpe Ablass Reaktor R001 Rückmeldung ein
A1.T2.A1T2S003.SV.C	BOOL	A 3.4	Pumpe Ablass Reaktor R001 Stellsignal
A1.T2.A1T2S004.SO+-.O+	BOOL	E 1.4	Pumpe Ablass Reaktor R002 Rückmeldung ein
A1.T2.A1T2S004.SV.C	BOOL	A 3.5	Pumpe Ablass Reaktor R002 Stellsignal
A1.T2.A1T2T001.TV.S	BOOL	A 4.5	Temperatur Reaktor R001 Stellsignal
A1.T2.A1T2T002.TV.S	BOOL	A 5.5	Temperatur Reaktor R002 Stellsignal
A1.T2.A1T2X001.GO+-.O+	BOOL	E 65.4	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B001 Rückmeldun...
A1.T2.A1T2X001.XV.C	BOOL	E 66.0	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B001 Rückmeldun...
A1.T2.A1T2X002.GO+-.O+	BOOL	E 1.4	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B002 Rückmeldun...
A1.T2.A1T2X002.XV.C	BOOL	E 65.5	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B002 Rückmeldun...
A1.T2.A1T2X003.GO+-.O+	BOOL	E 66.1	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B003 Rückmeldun...
A1.T2.A1T2X003.XV.C	BOOL	E 66.2	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B003 Rückmeldun...

8. Als Nächstes erfolgt die Verschaltung der Ansteuerung des Ausgangsoperanden. Dies geschieht, indem der Ausgang ‚PV_Out‘ am Baustein ‚OutStart‘ (PCS7DiOu) mit der rechten Maustaste angeklickt und anschließend Verschaltung zum Operand gewählt wird. (→ OutStart → PV_Out → Verschaltung zum Operand)



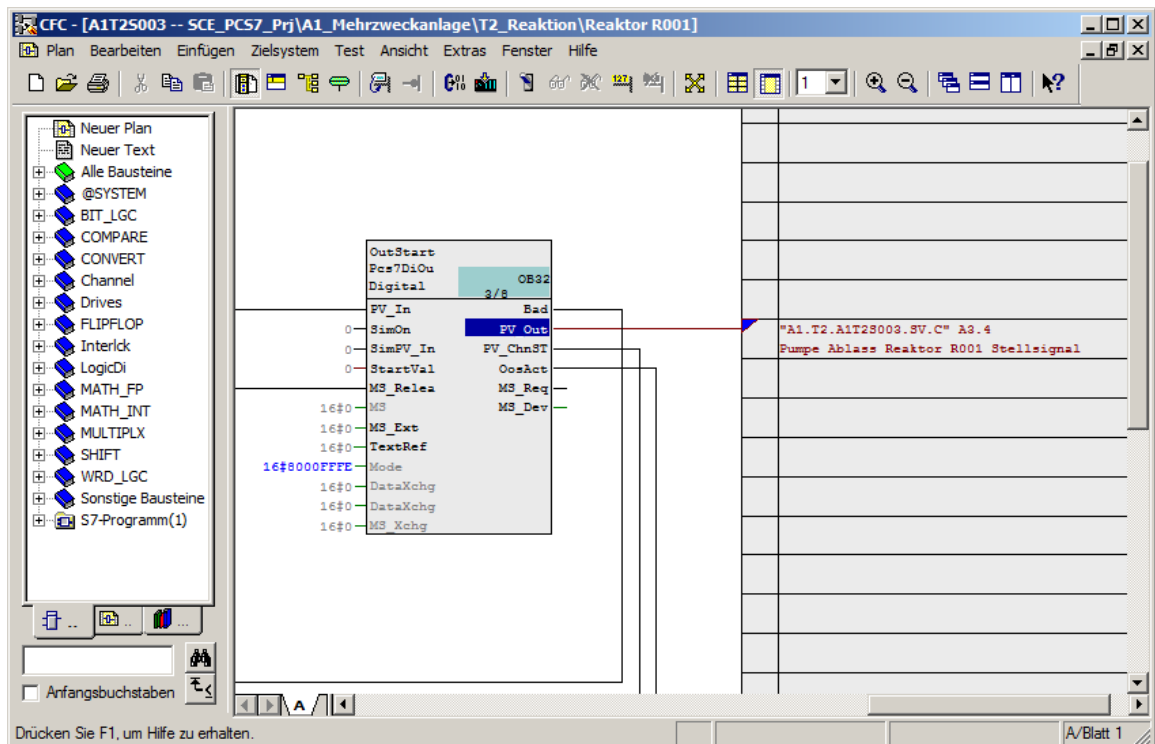
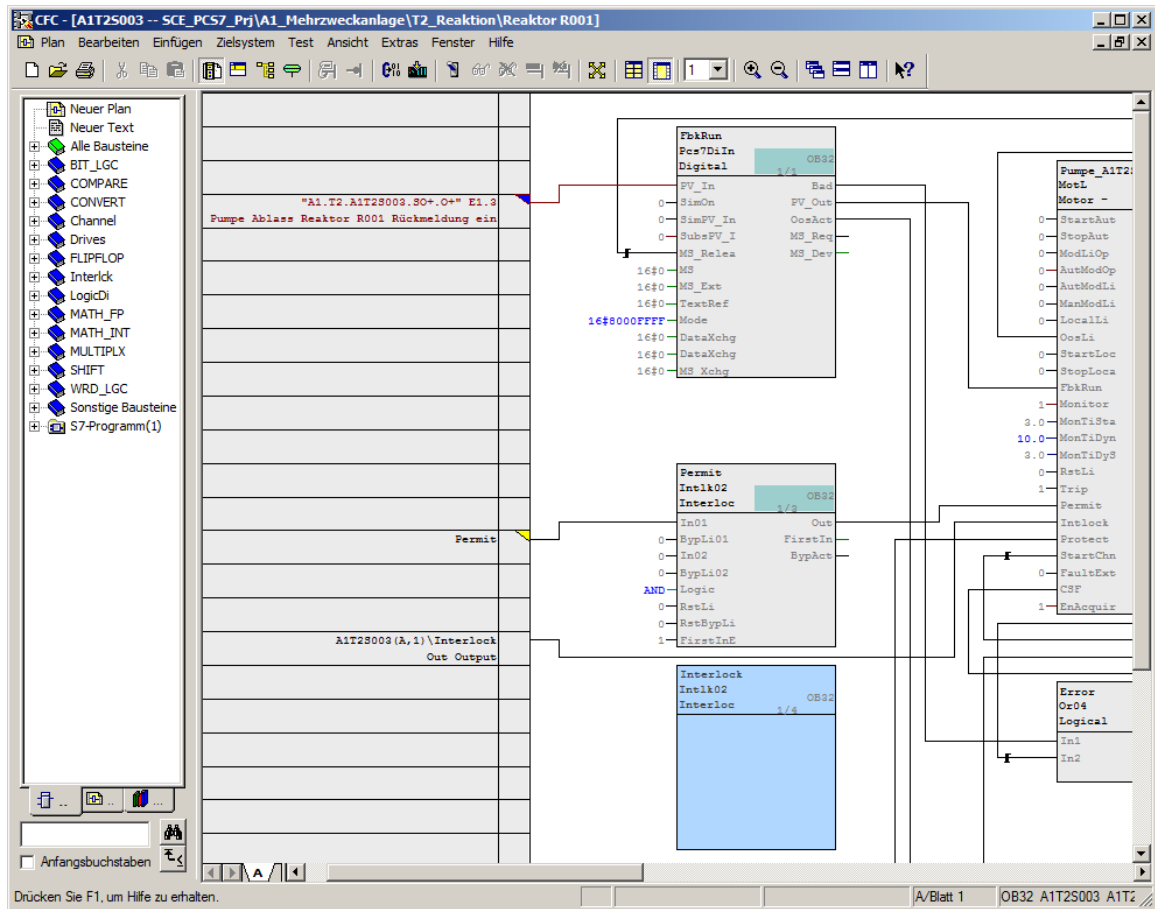
9. Der Operand zur Ansteuerung der Pumpe kann jetzt wieder komfortabel direkt aus der Symboltabelle ausgewählt werden. (→ A1.T2.A1T2S003.SV.C)

Signal	Data Type	Address	Description
A1.T2.A1T2H016.HS-.STOP	BOOL	E 5.7	Reaktor R002 Umfüllen nach Reaktor R001 Stoppen
A1.T2.A1T2S001.SO+-O+	BOOL	E 2.0	Rührer Reaktor R001 Rückmeldung ein
A1.T2.A1T2S001.SV.C	BOOL	A 4.6	Rührer Reaktor R001 Stellsignal
A1.T2.A1T2S002.SO+-O+	BOOL	E 2.1	Rührer Reaktor R002 Rückmeldung ein
A1.T2.A1T2S002.SV.C	BOOL	A 5.6	Rührer Reaktor R002 Stellsignal
A1.T2.A1T2S003.SO+-O+	BOOL	E 1.3	Pumpe Ablass Reaktor R001 Rückmeldung ein
A1.T2.A1T2S003.SV.C	BOOL	A 3.4	Pumpe Ablass Reaktor R001 Stellsignal
A1.T2.A1T2S004.SO+-O+	BOOL	E 1.4	Pumpe Ablass Reaktor R002 Rückmeldung ein
A1.T2.A1T2S004.SV.C	BOOL	A 3.5	Pumpe Ablass Reaktor R002 Stellsignal
A1.T2.A1T2T001.TV.S	BOOL	A 4.5	Temperatur Reaktor R001 Stellsignal
A1.T2.A1T2T002.TV.S	BOOL	A 5.5	Temperatur Reaktor R002 Stellsignal
A1.T2.A1T2X001.GO+-O+	BOOL	E 65.4	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B001 Rückmeldun..
A1.T2.A1T2X001.GO+-O-	BOOL	E 66.0	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B001 Rückmeldun..
A1.T2.A1T2X001.XV.C	BOOL	A 1.4	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B001 Stellsignal
A1.T2.A1T2X002.GO+-O+	BOOL	E 65.5	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B002 Rückmeldun..
A1.T2.A1T2X002.GO+-O-	BOOL	E 66.1	Auf/Zu-Ventil Zufuss Reaktor R001 aus Edukttank B002 Rückmeldun..

Hinweis:

- Der Platzhalter am Ausgang von Pcs7DiOu ,output start' sollte gelöscht werden, sonst gibt es beim Übersetzen eine Warnung!

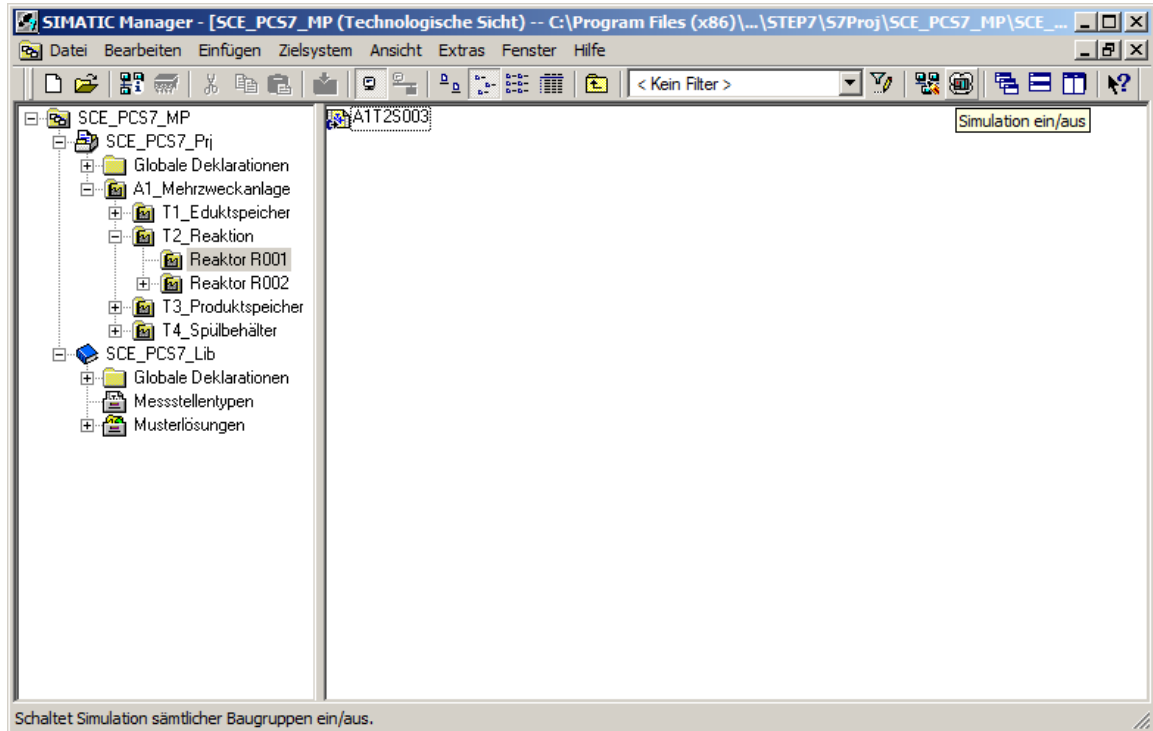
10. Das Ergebnis sieht folgendermaßen aus:



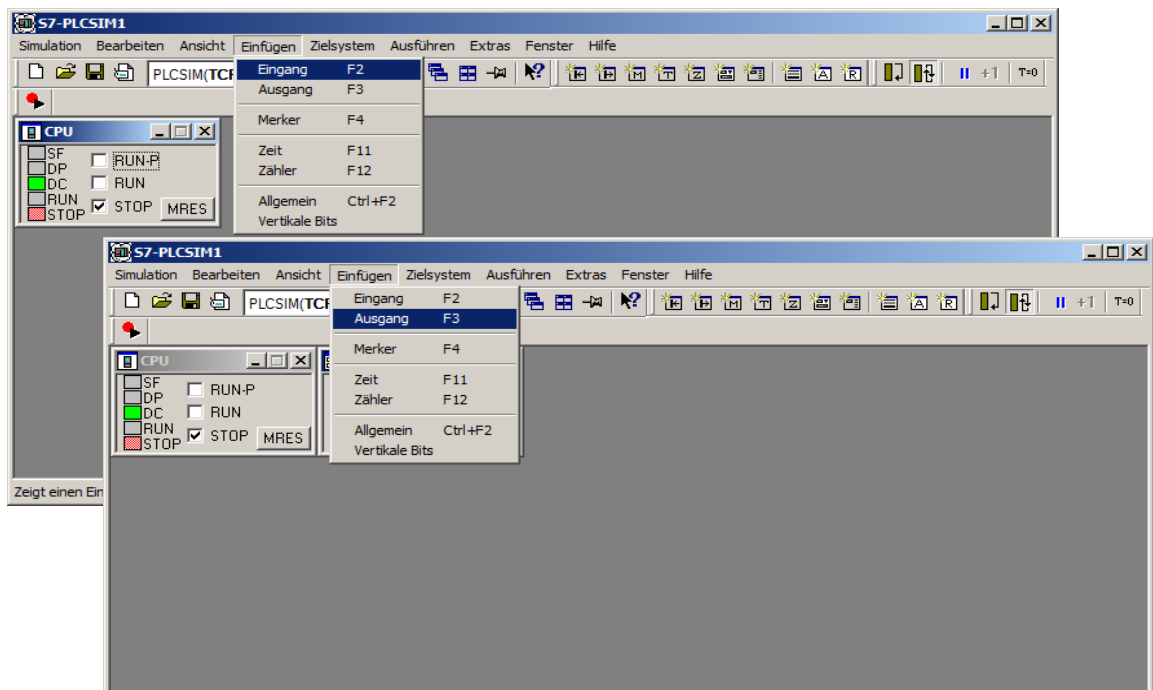
8.5 Simulation mit PLCSIM vorbereiten

1. Bevor das Programm für den Pumpenmotor übersetzt und geladen werden kann, muss aus dem SIMATIC Manager heraus die SPS-Simulation **S7-PLCSIM** gestartet werden.

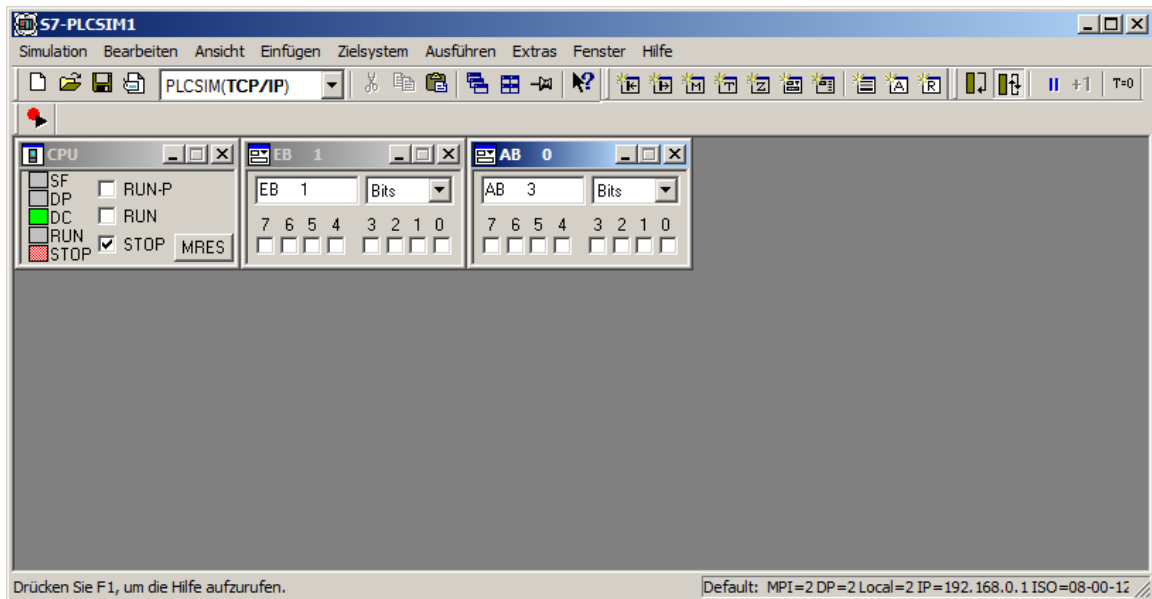
(→ SIMATIC Manager → )



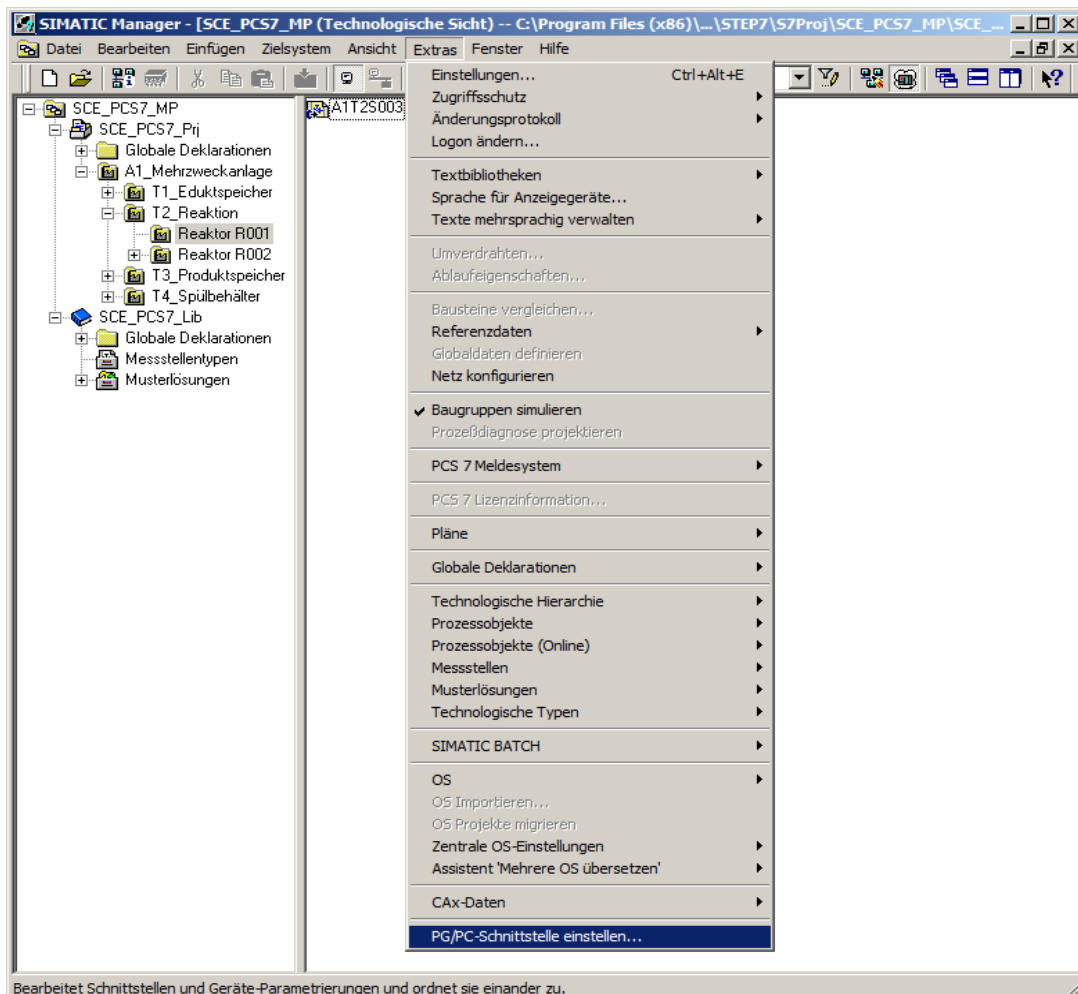
2. Die SPS-Simulation verhält sich wie eine richtige SIMATIC S7 CPU. Jedoch müssen Ein- und Ausgänge zuerst eingefügt werden, bevor diese beobachtet und geschaltet werden können. (→ Einfügen → Eingang → Einfügen → Ausgang)



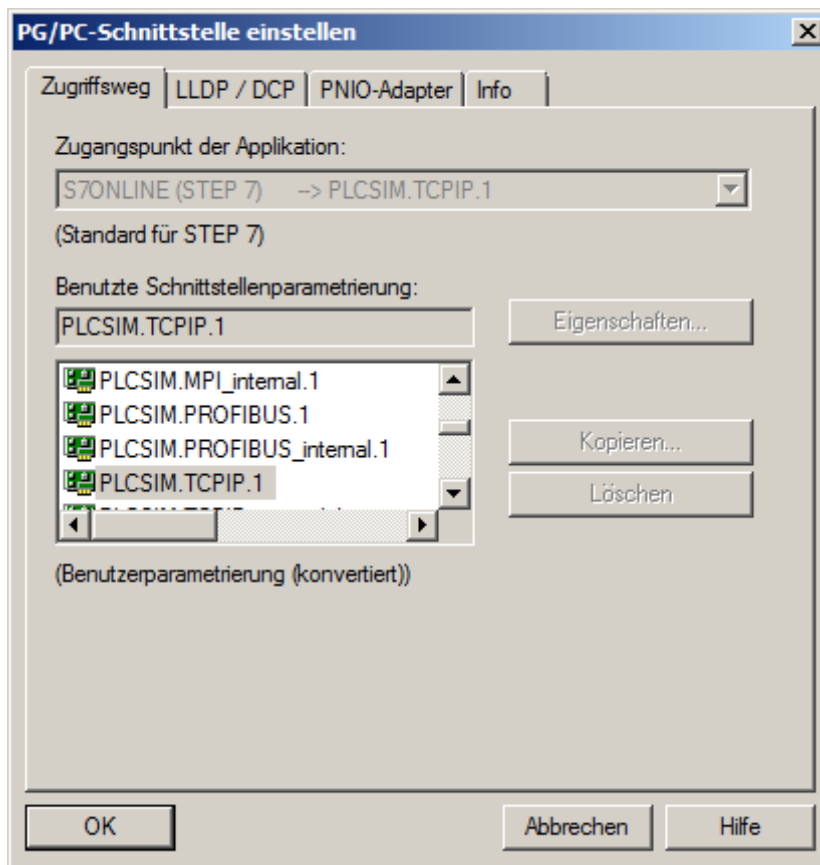
3. Nun müssen noch die richtigen Byte-Adressen eingetragen werden. (→ EB1 → AB3)



4. Damit aus dem SIMATIC Manager auch über die richtige Schnittstelle in **S7-PLCSIM** das Programm geladen werden kann, wird anschließend noch die PG/PC-Schnittstelle richtig eingestellt. (→ SIMATIC Manager → Extras → PG/PC-Schnittstelle einstellen)

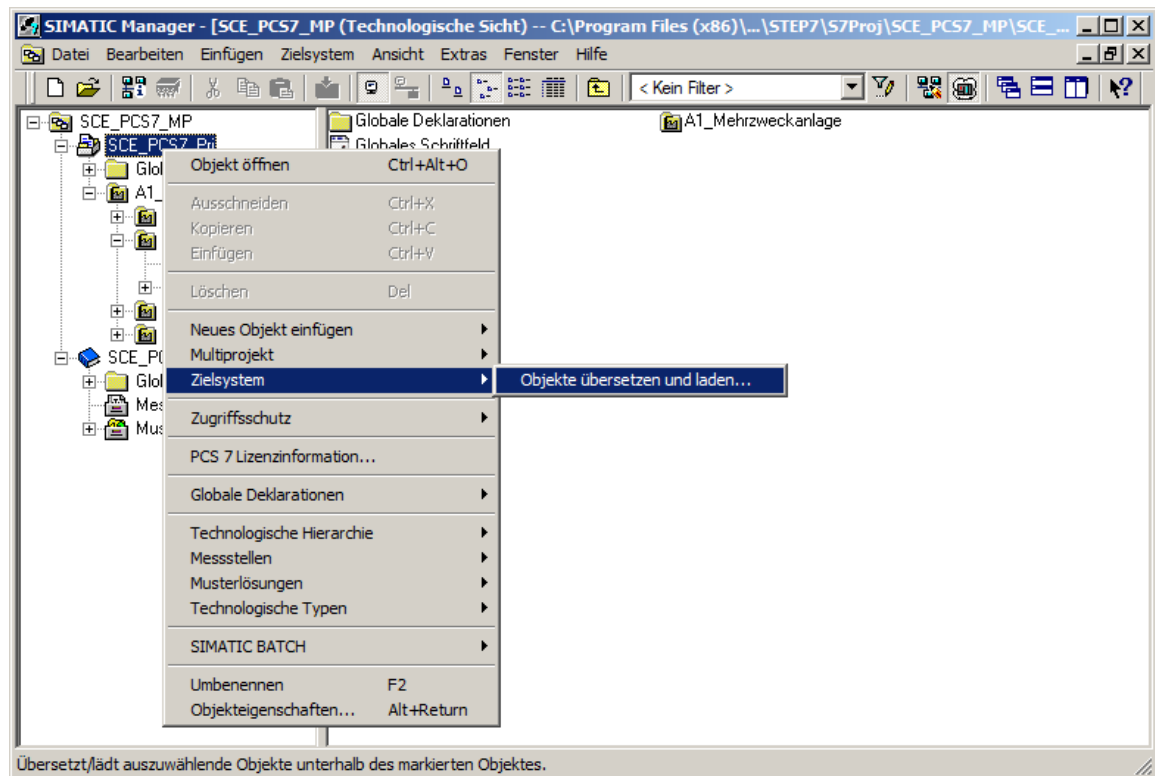


5. Als Schnittstelle wird hier PLCSIM.TCPIP.1 eingestellt. (→ PLCSIM.TCPIP.1 → OK)

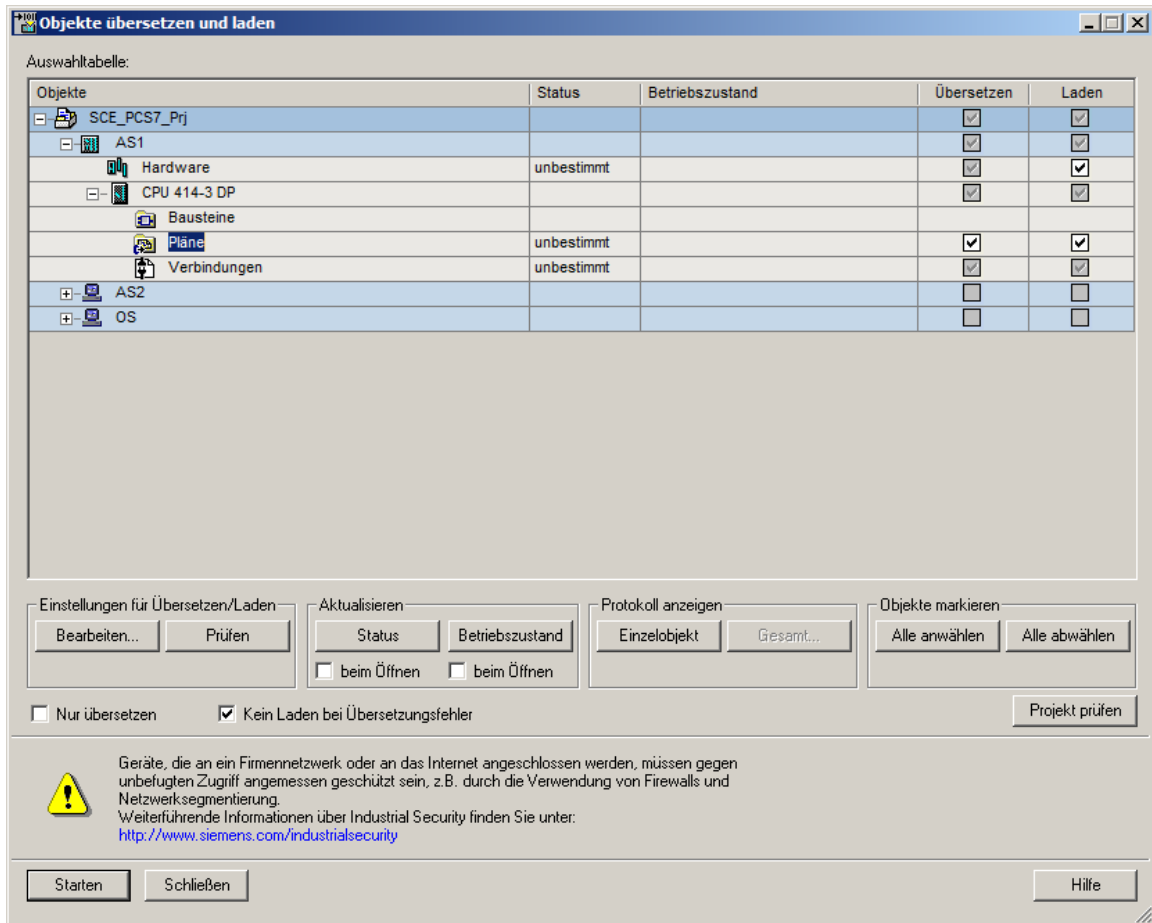


8.6 Objekte übersetzen und laden

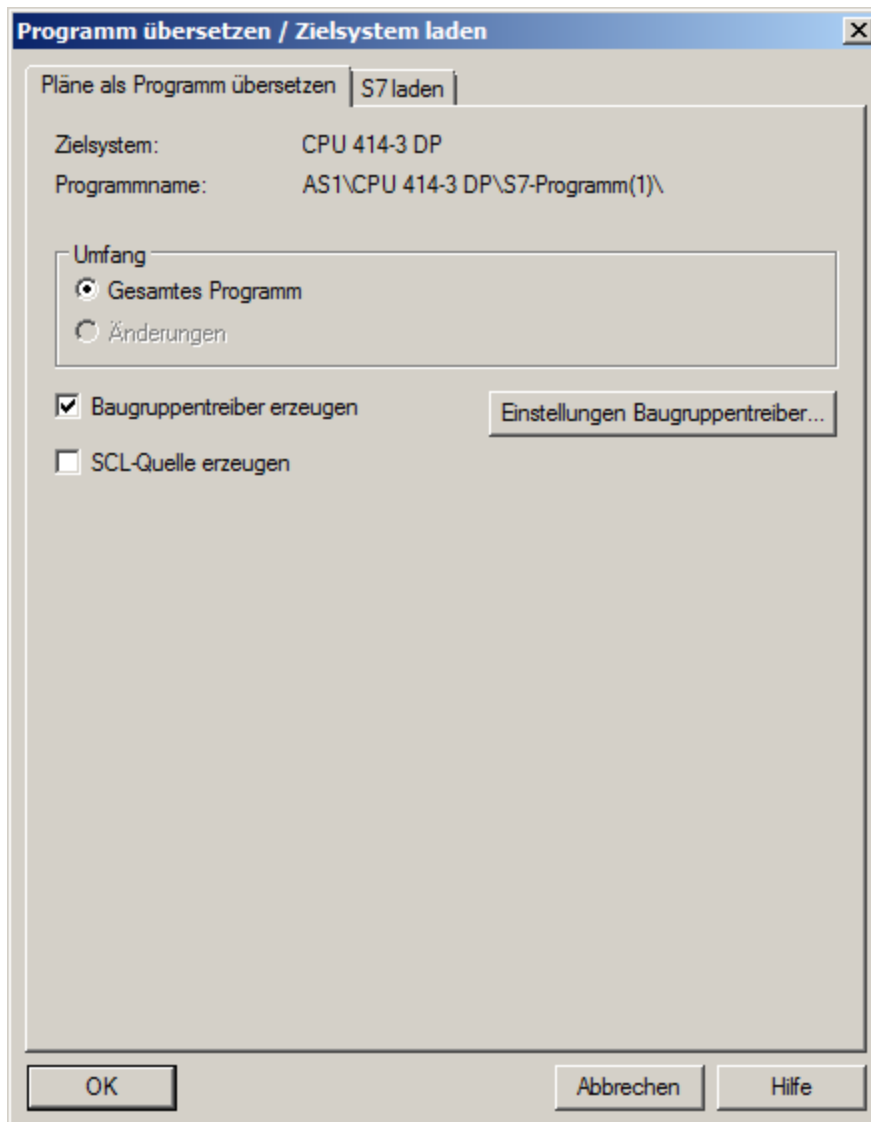
1. Nun kann der Projektordner markiert und mit dem Übersetzen und Laden der Objekte begonnen werden. (→ Technologische Sicht → SCE_PCS7_Prj → Zielsystem → Objekte übersetzen und laden)



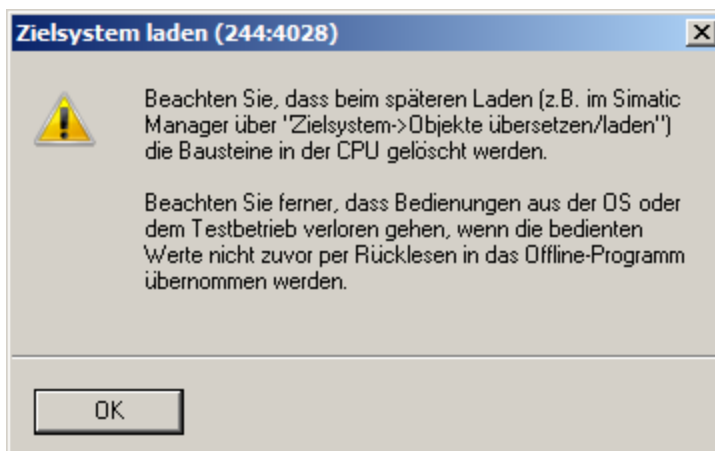
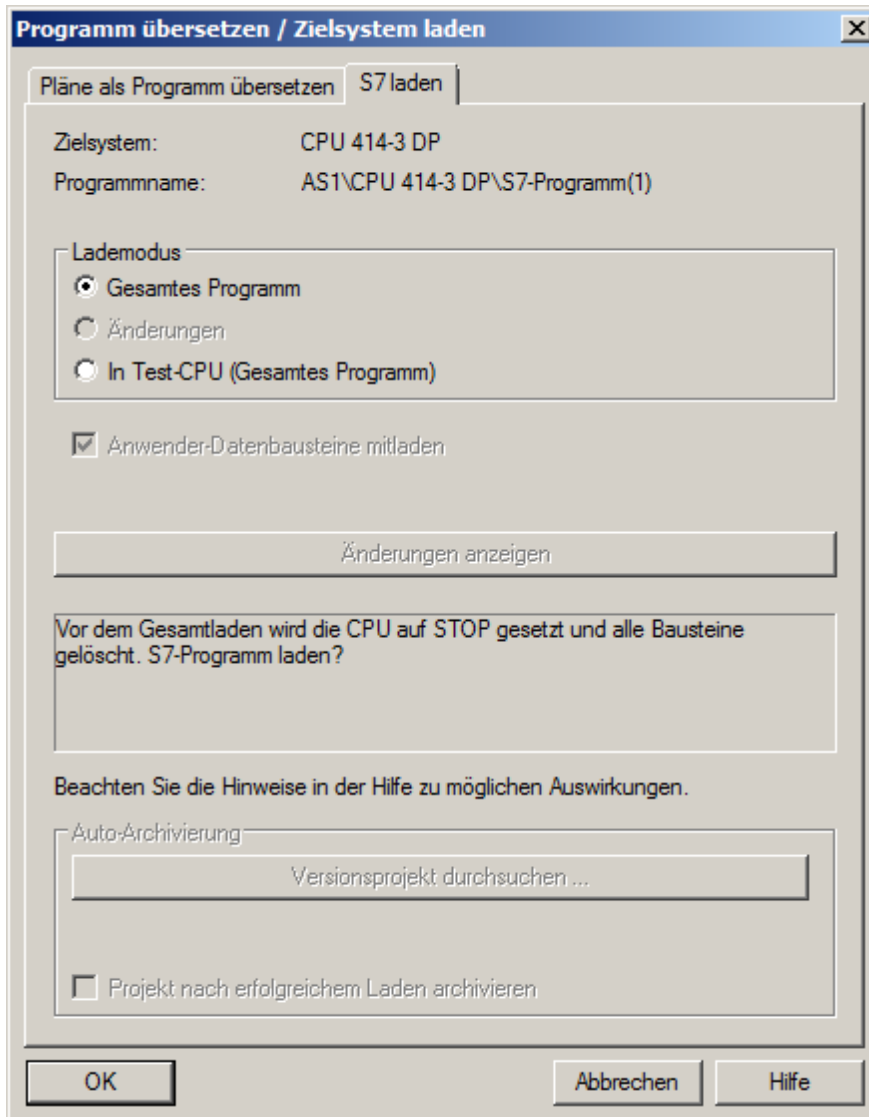
2. In der folgenden Auswahl wird für die Hardware und die Pläne der AS1 ‚Übersetzen und Laden‘ angewählt. Darauffolgend wird der Ordner ‚Pläne‘ markiert und dessen Einstellungen über ‚Bearbeiten‘ kontrolliert. (→ → → → Pläne → Bearbeiten)



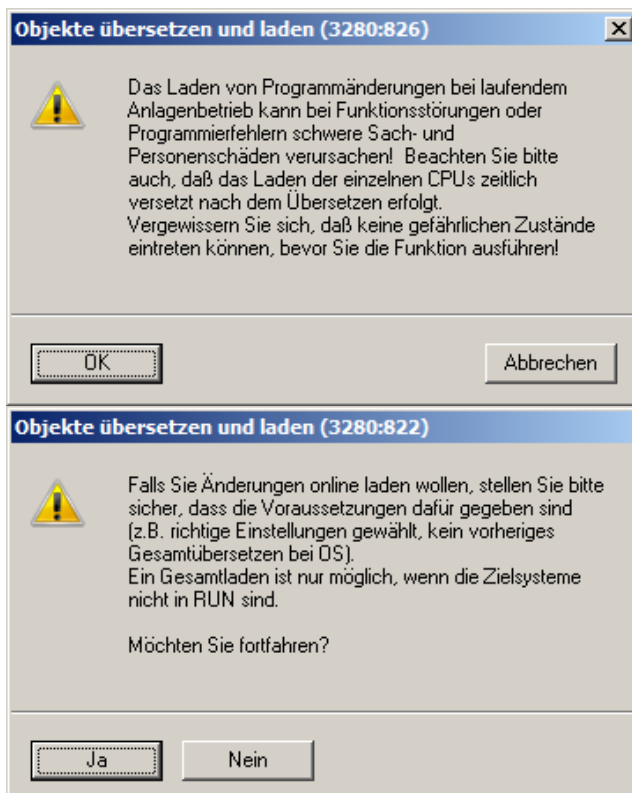
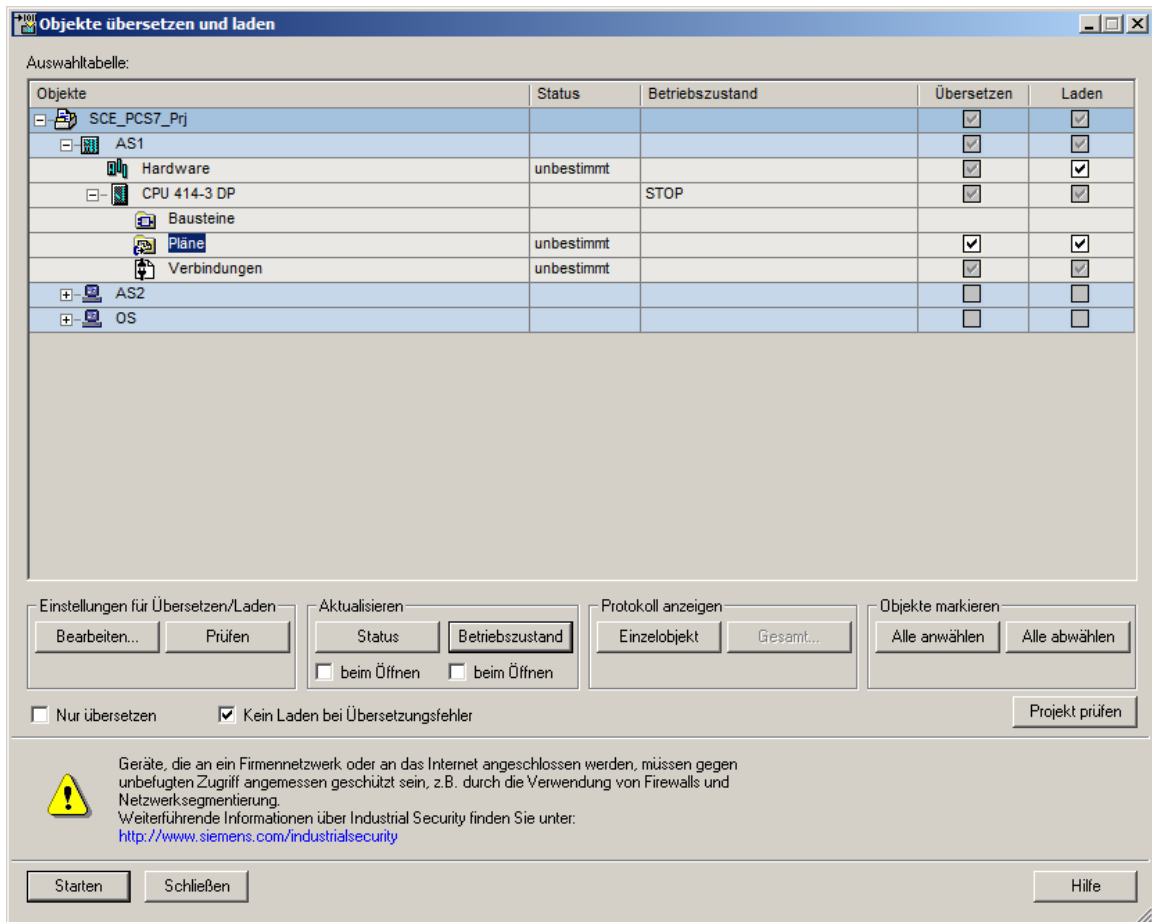
3. Beim Übersetzen der Pläne ist es wichtig das gesamte Programm zu übersetzen und die Baugruppentreiber erzeugen zu lassen. (→ Umfang: Gesamtes Programm → Baugruppentreiber erzeugen → S7 laden)




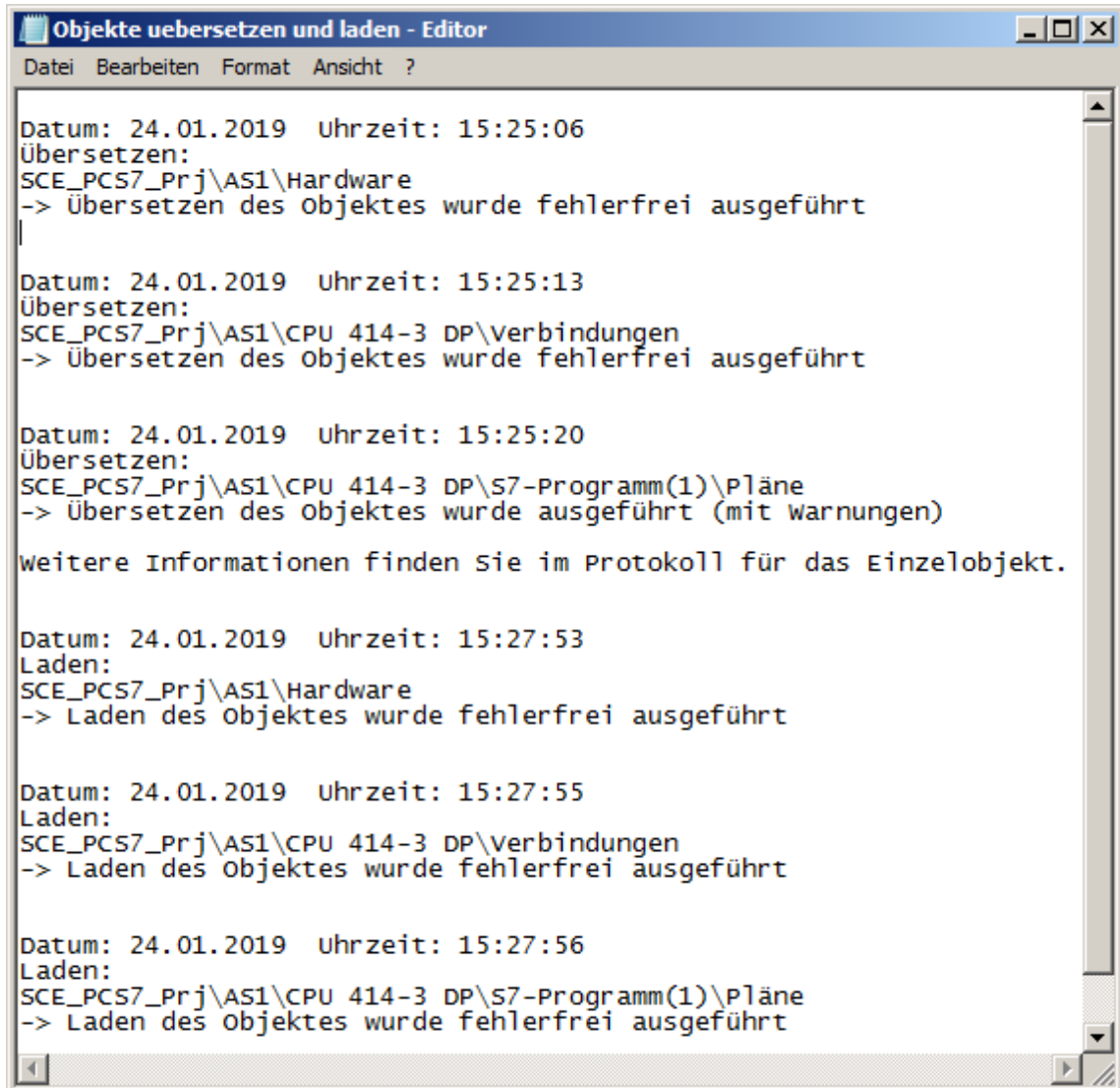
4. Beim Laden der Pläne ist es ebenfalls wichtig das gesamte Programm zu laden.
 (→ Lademodus: Gesamtes Programm → OK → OK)



5. Schließlich wird ‚Übersetzen und Laden‘ gestartet. Die Warnungen und Hinweise zur Anlagensicherheit sollten sorgfältig gelesen werden. Die CPU muss vor ‚Übersetzen und Laden‘ auf ‚STOP‘ geschaltet werden. (→ Starten → OK → Ja)



6. Am Ende werden Fehler und Warnungen in einem Protokoll angezeigt. Sie schließen das Fenster. (→ )



```
Objekte uebersetzen und laden - Editor
Datei Bearbeiten Format Ansicht ?

Datum: 24.01.2019  Uhrzeit: 15:25:06
Übersetzen:
SCE_PCS7_Prj\AS1\Hardware
-> Übersetzen des Objektes wurde fehlerfrei ausgeführt

Datum: 24.01.2019  Uhrzeit: 15:25:13
Übersetzen:
SCE_PCS7_Prj\AS1\CPU 414-3 DP\Verbindungen
-> Übersetzen des Objektes wurde fehlerfrei ausgeführt

Datum: 24.01.2019  Uhrzeit: 15:25:20
Übersetzen:
SCE_PCS7_Prj\AS1\CPU 414-3 DP\S7-Programm(1)\Pläne
-> Übersetzen des Objektes wurde ausgeführt (mit warnungen)

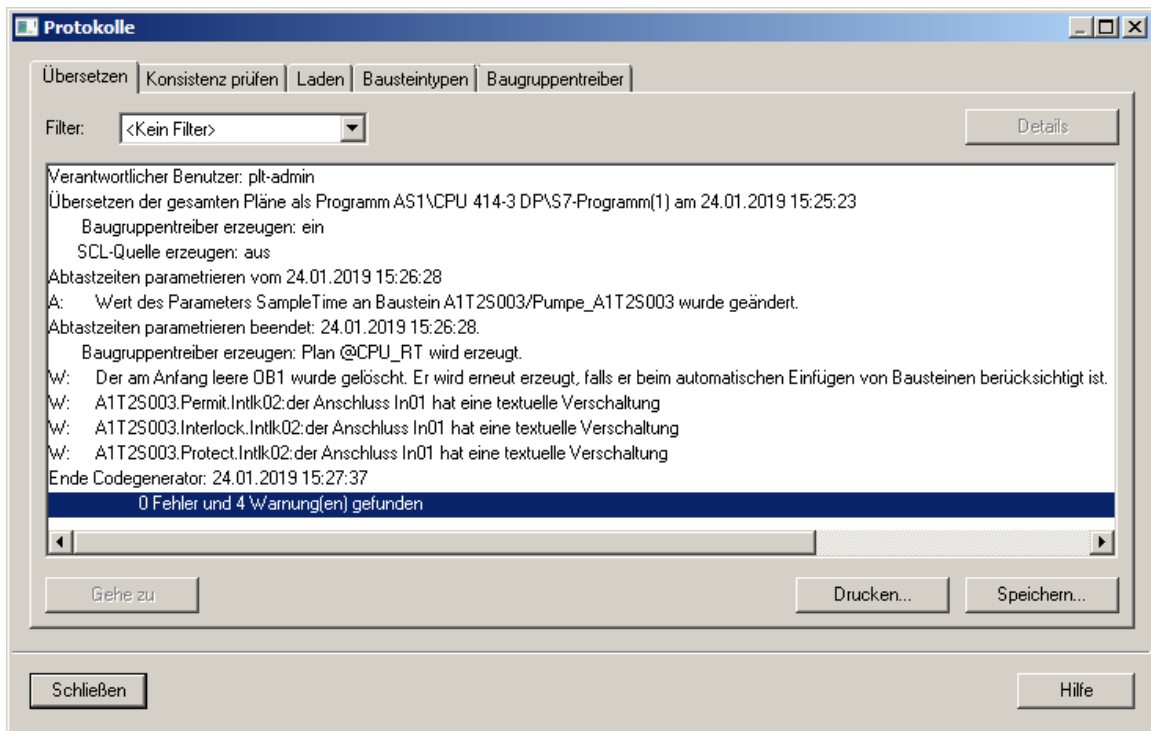
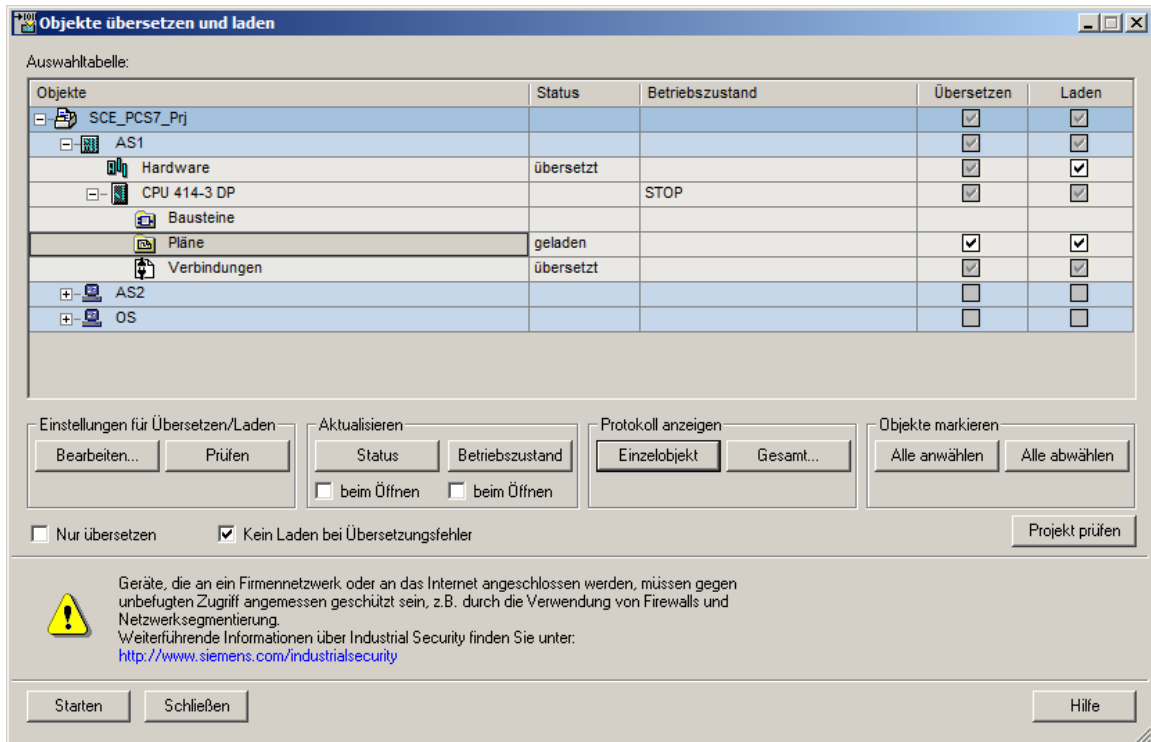
Weitere Informationen finden Sie im Protokoll für das Einzelobjekt.

Datum: 24.01.2019  Uhrzeit: 15:27:53
Laden:
SCE_PCS7_Prj\AS1\Hardware
-> Laden des Objektes wurde fehlerfrei ausgeführt

Datum: 24.01.2019  Uhrzeit: 15:27:55
Laden:
SCE_PCS7_Prj\AS1\CPU 414-3 DP\Verbindungen
-> Laden des Objektes wurde fehlerfrei ausgeführt

Datum: 24.01.2019  Uhrzeit: 15:27:56
Laden:
SCE_PCS7_Prj\AS1\CPU 414-3 DP\S7-Programm(1)\Pläne
-> Laden des Objektes wurde fehlerfrei ausgeführt
```

7. Wollen Sie Details zu dem Protokoll ansehen, so müssen Sie bei Protokoll anzeigen auf Einzelobjekt klicken. (→ Einzelobjekt → Schließen → Schließen)

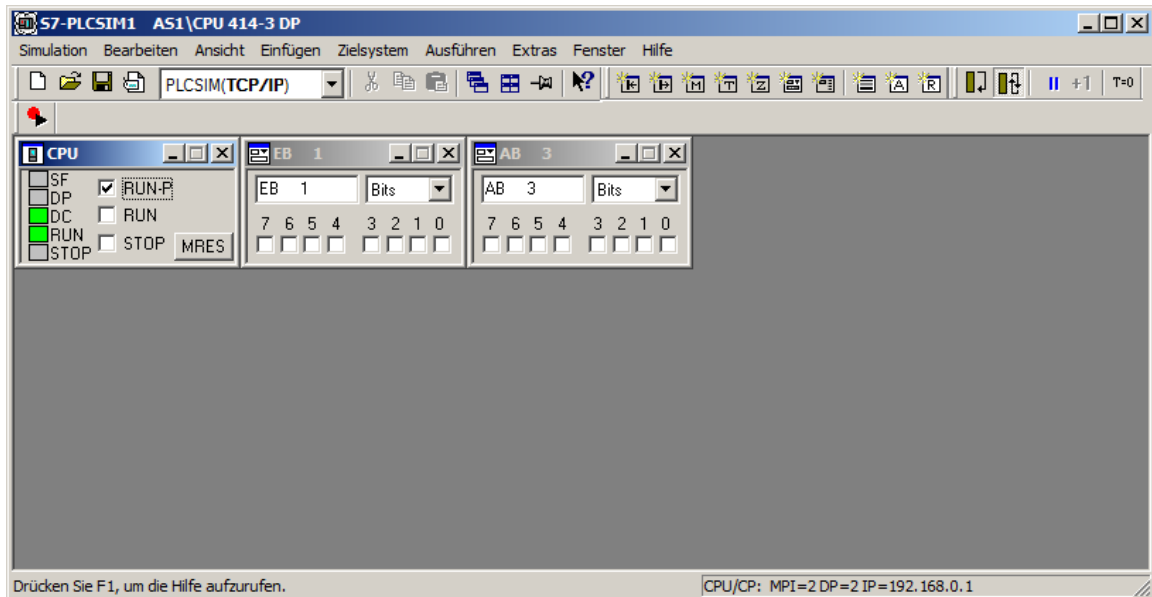



Hinweis:

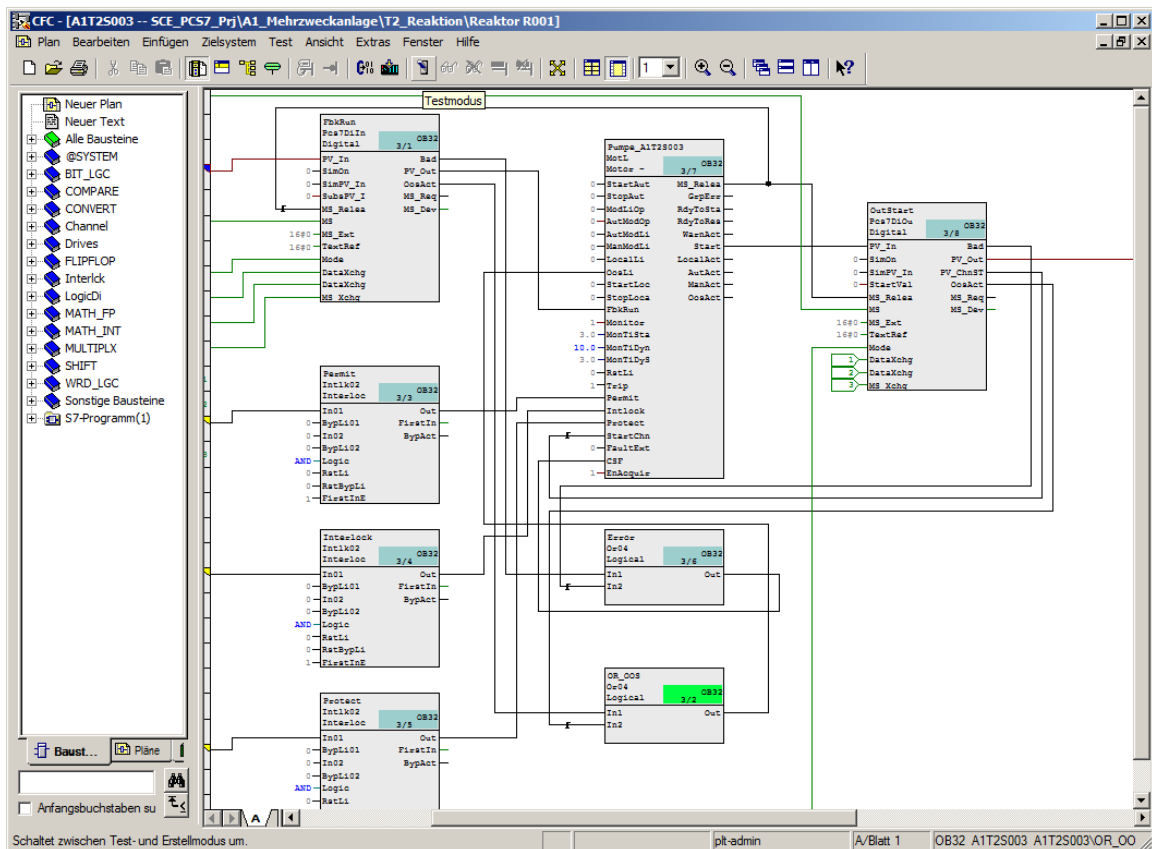
- Hier erscheinen vier Warnungen: Löschen des leeren OB 1 und das Vorhandensein von textuellen Verschaltungen. Letztere entstehen durch die nicht verknüpften Anschlüsse des Templates. In Kapitel P01-05 werden diese Anschlüsse verbunden.

8.7 Testen des Bausteins A1T2S003

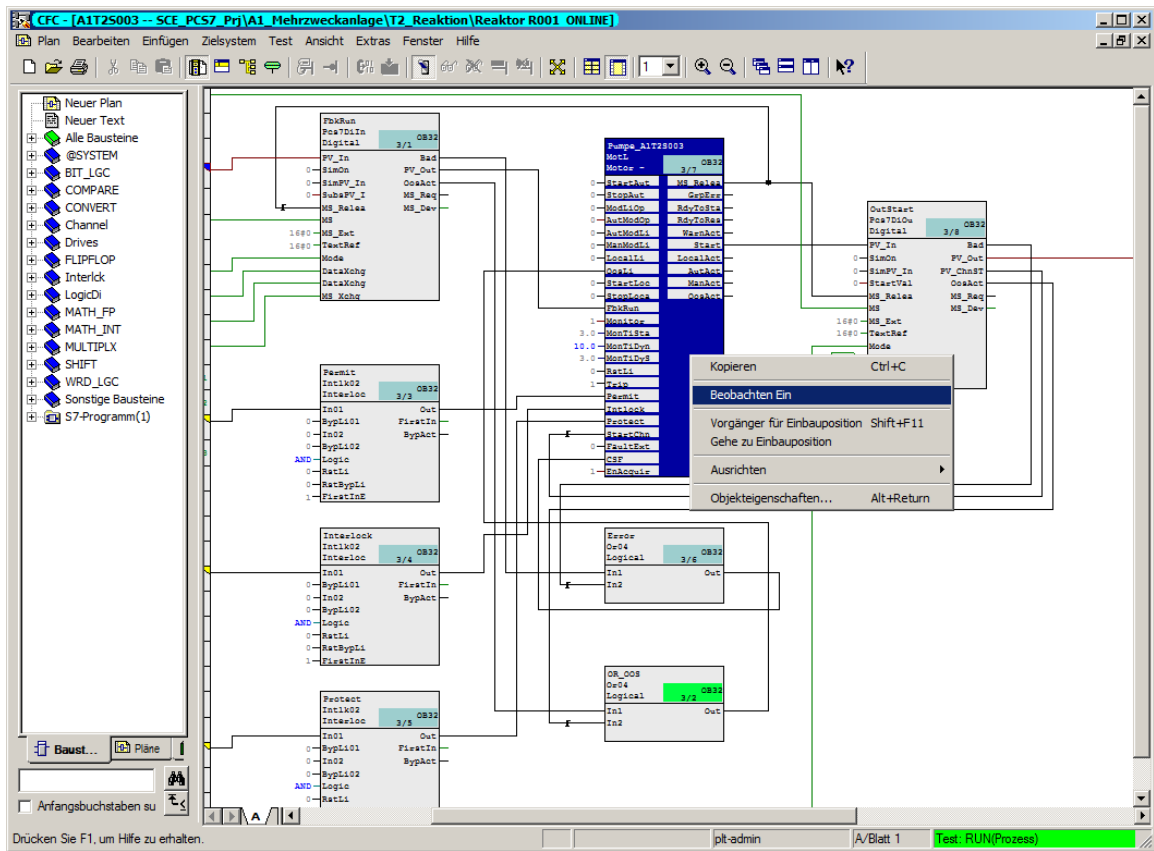
1. Zum Testen des Programms wird nun die CPU in **S7-PLCSIM** auf ‚RUN-P‘ geschaltet.
(→ S7-PLCSIM → RUN-P)



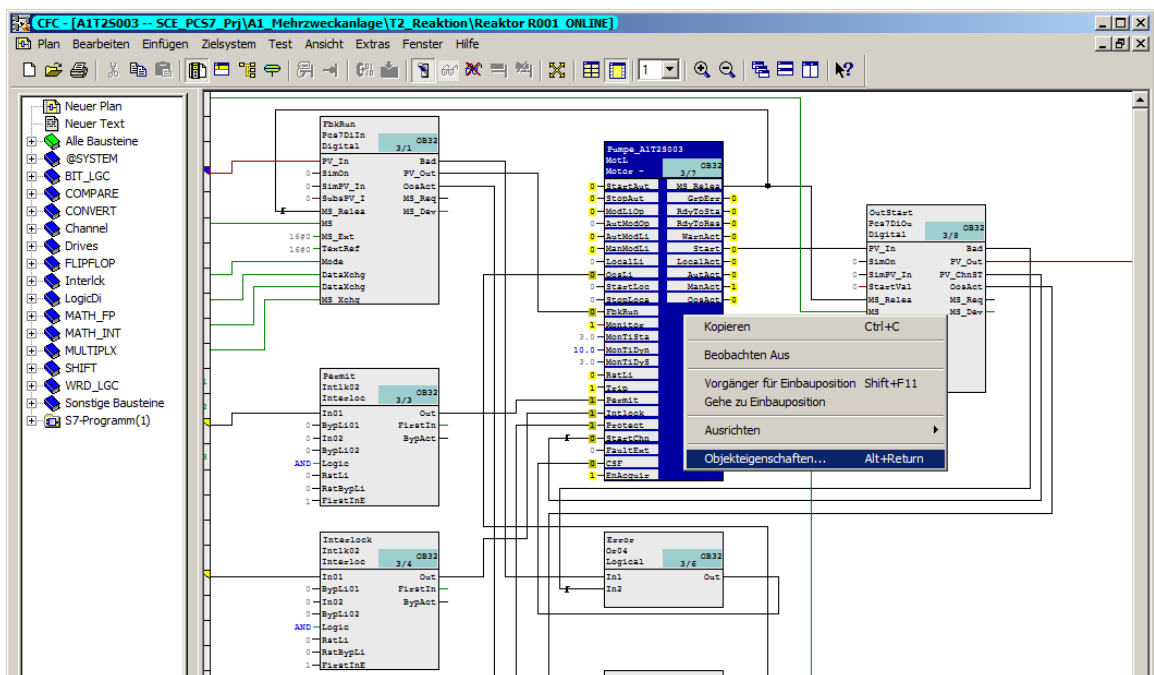
2. Bevor im CFC die einzelnen Bausteine beobachtet werden können, muss der Plan zuerst einmal in den Testmodus geschaltet werden. (→ CFC → )



- Die Bausteine, welche beobachtet werden sollen nun explizit zum Beobachten freigeschaltet werden. Dasselbe gilt anschließend für einzelne Anschlüsse des Bausteins. (→ Pumpe_A1T2S003 → Beobachten Ein)



- Für das weitere Vorgehen ist es notwendig, dass die Anschlüsse für die Automatiksteuerung ‚StartAut‘ und ‚StopAut‘ des Bausteins ‚MotL‘ sichtbar sind. Für den Fehlerfall sollten auch ‚RstOp‘ und ‚MonDynErr‘ sichtbar geschaltet werden. (→ Objekteigenschaften)



(→ RstOp → Unsichtbar →)

Eigenschaften - Baustein -- A1T25003\Pumpe_A1T25003

Algemein Anschlüsse

#	Name	I/O	Typ	W...	Ve...	Fo...	Fo...	Fo...	SF...	Te...	Ko...	Unsichtbar	Fü...	Ar...
263	R_StpAct	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1 ...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
265	R_StpAct.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
264	R_StpAct.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
48	RapidStp	IN	BOOL	0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1 ...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Kei...
275	RdyToReset	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1 ...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
277	RdyToReset.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
276	RdyToReset.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
272	RdyToStart	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1 ...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
274	RdyToStart.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
273	RdyToStart.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
50	RstLi	IN	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Lin...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
52	RstLi.ST	IN	...	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
51	RstLi.Value	IN	...	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
49	RstOp	IN	BOOL	0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Op...	<input type="checkbox"/>	<input type="checkbox"/>	Kei...
308	Run	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1=...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
310	Run.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
309	Run.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
97	RunUpCyc	IN	INT	3		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Nu...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
89	SampleTime	IN	REAL	1.0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Sa...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
136	SelfP1	IN	ANY			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Sel...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
137	SelfP2	IN	ANY			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Sel...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
110	SimAV	IN	REAL	0.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	AV...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Kei...
111	SimAV_Li	IN	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Lin...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
113	SimAV_Li.ST	IN	...	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
112	SimAV_Li.Value	IN	...	0.0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
90	SimLiOp	IN	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
92	SimLiOp.ST	IN	...	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
91	SimLiOp.Value	IN	...	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

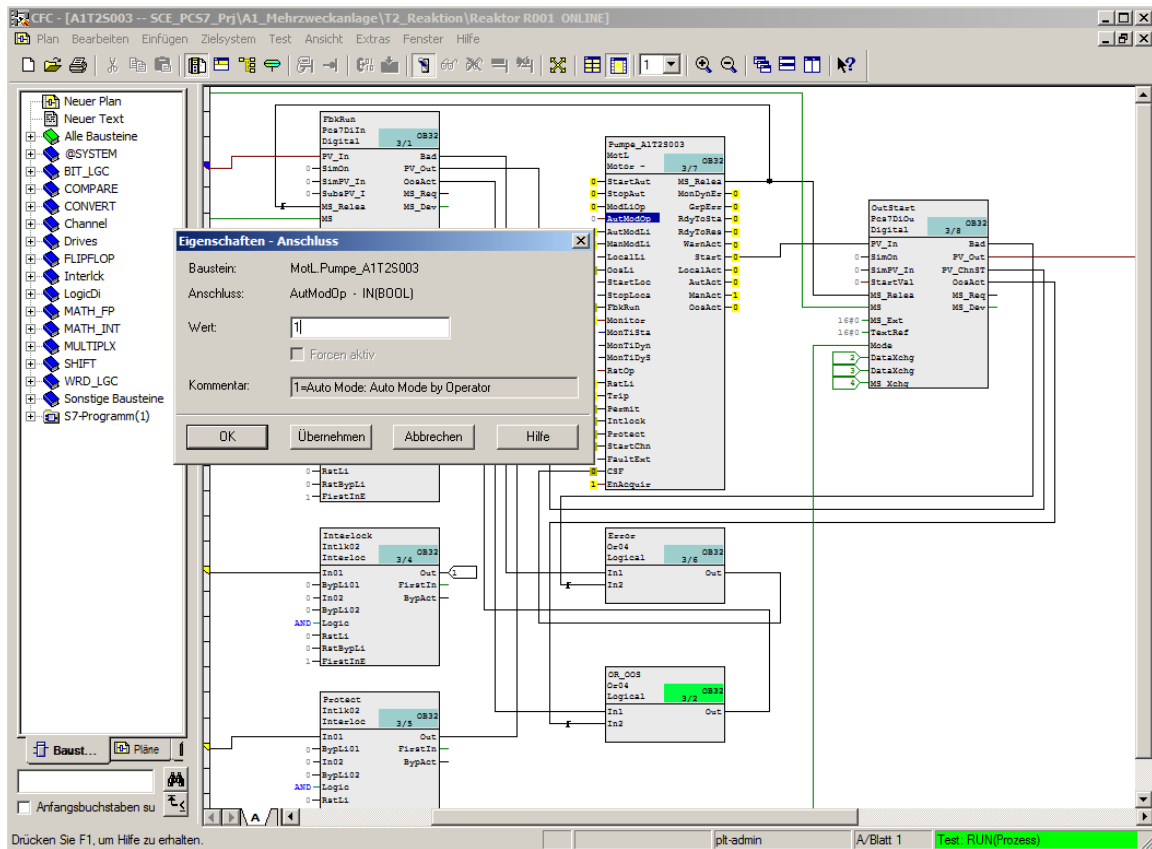
OK Werte übernehmen Drucken Abbrechen Hilfe

(→ MonDynErr → Unsichtbar → → OK)

10	ModLiOp	IN	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1=...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
12	ModLiOp.ST	IN	...	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	ModLiOp.Value	IN	...	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
253	MonDynErr	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Fe...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
255	MonDynErr.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
254	MonDynErr.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
256	MonDynStopErr	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Fe...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
258	MonDynStopErr.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
257	MonDynStopErr....	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
40	Monitor	IN	BOOL	1		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1=...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Kei...
259	MonStaErr	OUT	ST...			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Fe...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
261	MonStaErr.ST	16...	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Si...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
260	MonStaErr.Value	0	<n...	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	V...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
42	MonTiDynamic	IN	REAL	10.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Mo...	<input type="checkbox"/>	<input type="checkbox"/>	Kei...
43	MonTiDyStop	IN	REAL	3.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Mo...	<input type="checkbox"/>	<input type="checkbox"/>	Kei...
41	MonTiStatic	IN	REAL	3.0		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	Mo...	<input type="checkbox"/>	<input type="checkbox"/>	Kei...
250	MS_Release	OUT	ST...		A1T2 S003 Out Start	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	1: ...	<input type="checkbox"/>	<input type="checkbox"/>	

OK Werte übernehmen Drucken Abbrechen Hilfe

5. Als Nächstes wird die Umschaltung für den Hand-/Automatikbetrieb auf Automatik durch ‚AutModOp‘ == 1 vorgenommen. (→ AutModOp → Eigenschaften → Wert: 1 → OK)

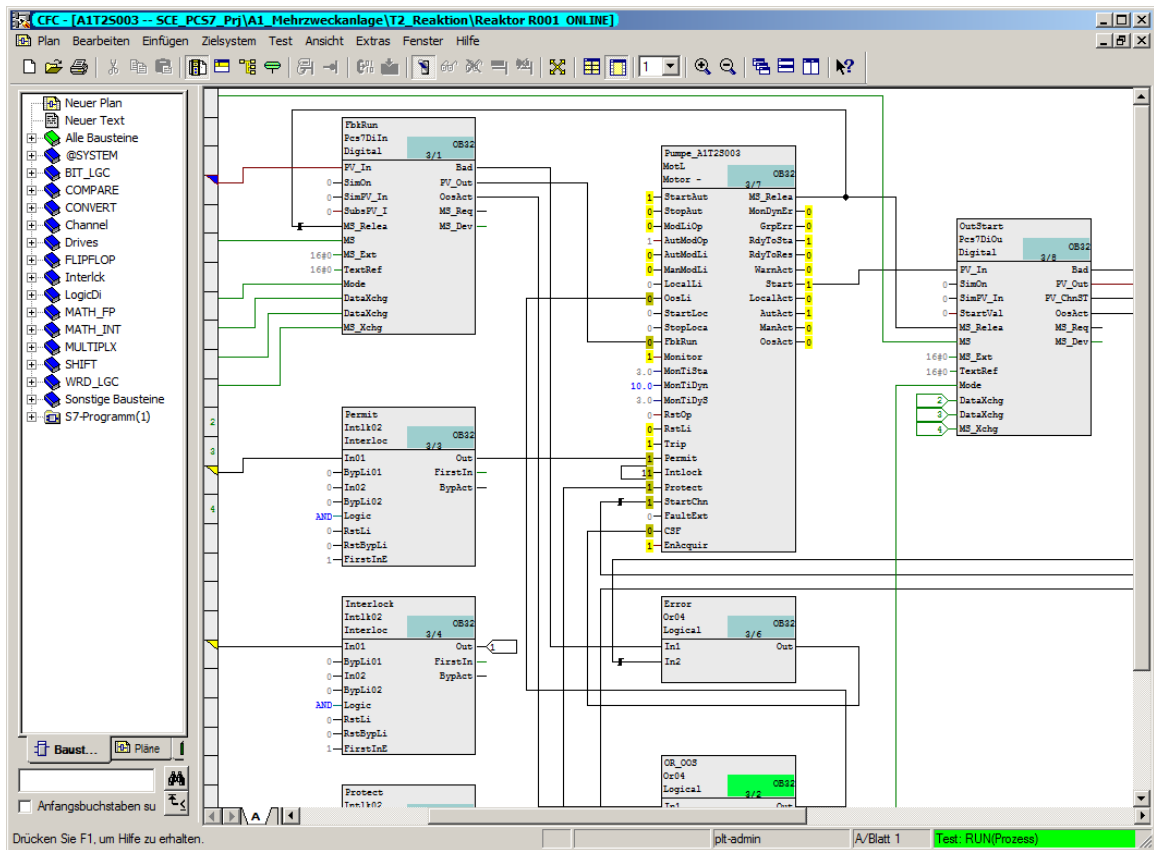


Hinweise:

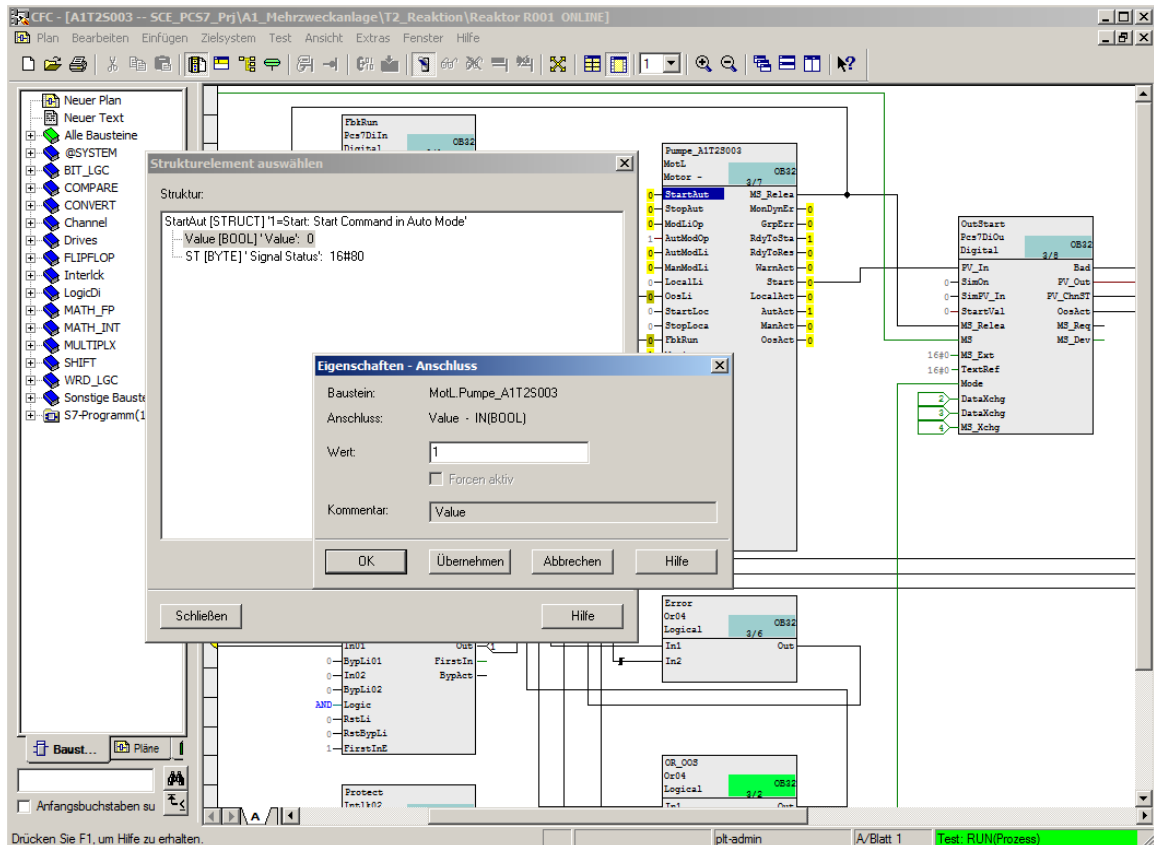
- Beim Testen sollte nicht vergessen werden innerhalb von 10 Sekunden nach Ansteuerung des Ausgangs A 3.4 in **S7-PLCSIM** die Rückmeldung E 1.3 zu setzen. Wird dies vergessen, so schaltet der Baustein *Pumpe_A1T2S003* ab und gibt einen Fehler aus.
- Der Pumpenbaustein schaltet bei fehlender Rückmeldung nicht nur das Stellsignal Start auf 0, sondern zeigt durch Setzen des Ausgangs ‚ModDynErr‘ auf 1 auch an, dass die Laufmeldung der Pumpe nicht rechtzeitig eingegangen war. Dazu müssen die entsprechenden Anschlüsse sichtbar gemacht werden. Um Schäden durch wiederholte Einschaltversuche zu vermeiden, muss der Pumpenbaustein erst zurückgesetzt werden, bevor ein neuerlicher Versuch gestartet werden kann. Hierzu ist der Eingang ‚RstOp‘ kurz auf 1 und anschließend wieder auf 0 zu setzen!

Mit Doppelklick auf diesen Eingangsparameter des Pumpenbausteins *Pumpe_A1T2S003* wird das oben gezeigte Dialogfenster geöffnet. Im Feld Wert wird zunächst eine 1 eingetragen, dieser Wert wird mit Click auf den Button ‚Übernehmen‘ an das Leitsystem übertragen und die Fehlerausgänge auf 0 gesetzt. Um zur normalen Funktionsweise zurückzukehren, muss ‚RstOp‘ durch Eingabe von 0 und nochmals ‚Übernehmen‘ abschließend auf den ursprünglichen Zustand zurückgesetzt werden.

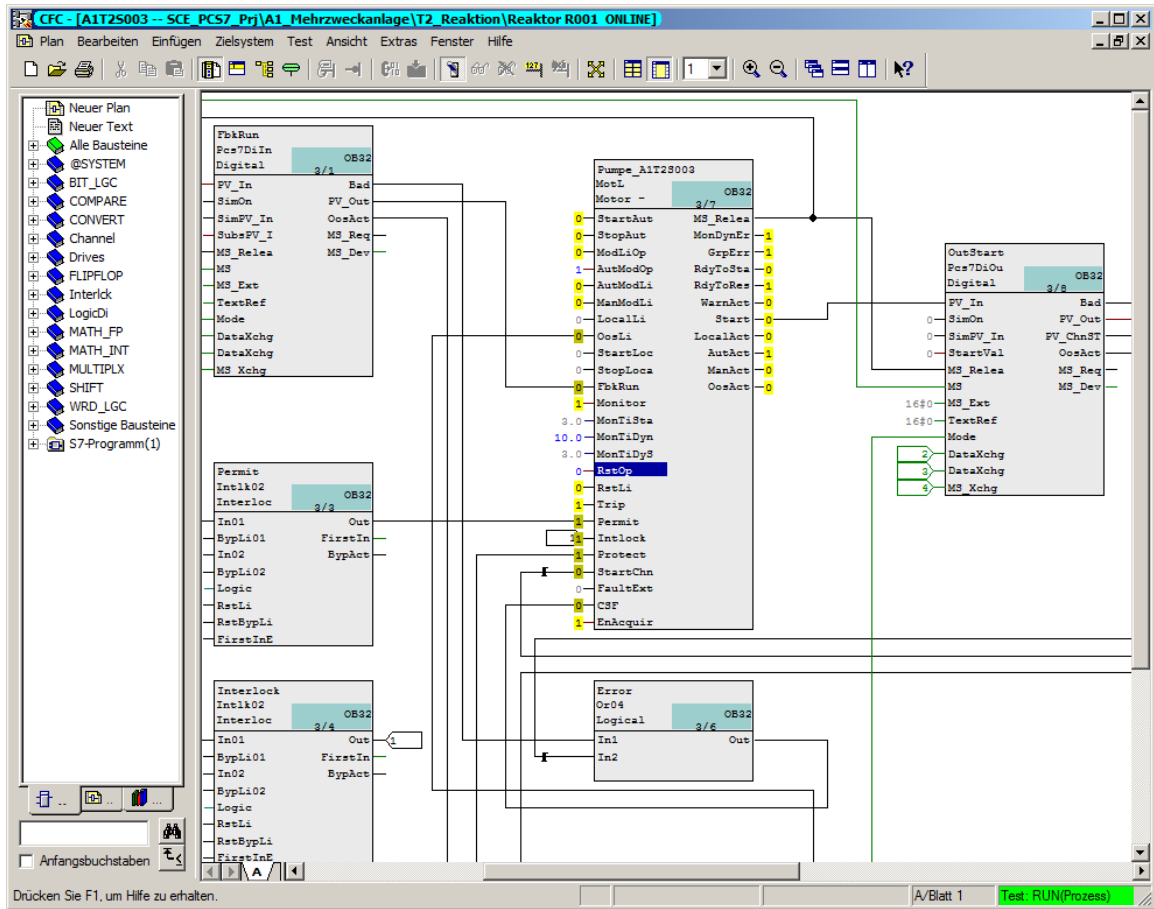
6. Im nächsten Schritt wird die Pumpe mit (StartAut == 1 und StopAut == 0) gestartet und kann daraufhin mit (StartAut == 0 und StopAut == 1) wieder gestoppt werden.



(→ StartAut → Eigenschaften → Value → Wert: „1“)



7. Falls die Laurrückmeldung nicht rechtzeitig im PLCSIM angeschaltet wurde, wird im Anschluss ‚MonDynErr‘ ein Fehler angezeigt. Dieser kann mit ‚RstOp‘ == 1 quittiert werden.



8.8 Checkliste – Schritt-für-Schritt-Anleitung

Die nachfolgende Checkliste hilft den Studierenden selbstständig zu überprüfen, ob alle Arbeitsschritte der Schritt-für-Schritt-Anleitung sorgfältig abgearbeitet wurden und ermöglicht eigenständig das Modul erfolgreich abzuschließen.

Nr.	Beschreibung	Geprüft
1	Symboltabelle vollständig	
2	Motor_Lean in Messstellentypen (Stammdatenbibliothek) vorhanden und konfiguriert	
3	Reaktor R001\A1T2S003 als Instanz von Motor_Lean angelegt und konfiguriert	
4	Objekte übersetzen und laden erfolgreich	
5	Testen von A1T2S003 erfolgreich	
6	Projekt erfolgreich archiviert	

Tabelle 4: Checkliste für Schritt-für-Schritt-Anleitung

9 Testen der Automatisierungslogik mit der Simulation (optional)

Die manuelle Eingabe von Prozesszuständen an das simulierte Leitsystem ist beim Testen kleiner Funktionen noch mit vertretbarem Aufwand möglich. Bei aufwändigeren Abläufen mit mehreren dynamischen Prozessgrößen sind jedoch schnell die Grenzen erreicht. Hier empfiehlt sich der Einsatz einer Prozesssimulation.

Für diesen Kurs wurden deshalb die wesentlichen Zusammenhänge des hier zu automatisierenden Prozesses mit der Simulationssoftware **SIMIT** abgebildet. Das Modell bildet das dynamische Verhalten der Pumpen, Ventile, Behälter, Reaktoren sowie das Vor-Ort Bedienpanel mit Hauptschalter, NOTAUS, Umschaltung auf lokale Vor-Ort Bedienung und die entsprechenden Bedienelemente ab. Die dynamischen Vorgänge sind gegenüber der Realität um den Faktor 5 bis 50 beschleunigt, um die Wartezeiten kurz zu halten.

Die Bedienoberfläche des Simulators ist in Abbildung 5 abgebildet. Sie stellt auf der linken Seite das Prozessschema sowie die Signalpegel von Stell- und Messgrößen dar. Auf der rechten Seite ist oben das ockerfarben hinterlegte Vor-Ort-Bedienpanel dargestellt, unten sind außerdem eine Reihe Steuerelemente für die Simulation angebracht.

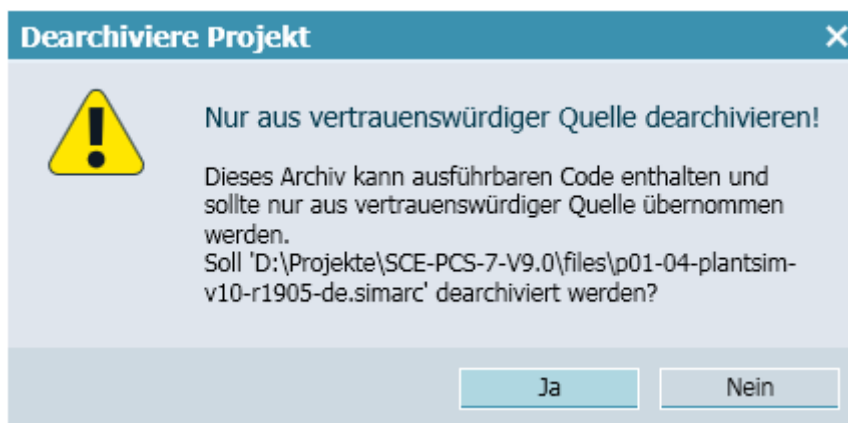
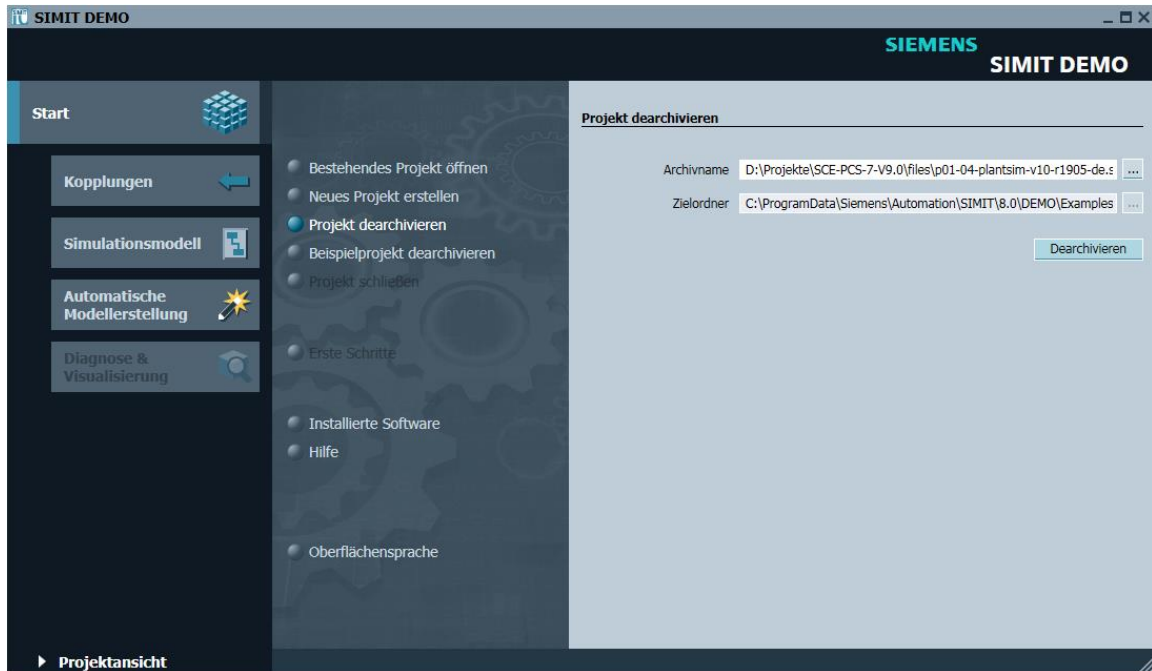
Die Anwendung des Simulators ist denkbar einfach – es muss lediglich darauf geachtet werden, dass die Belegung der Ein- und Ausgänge nicht verändert wurde.

Mit dem Simulator kann die Pumpenansteuerung nun sehr einfach überprüft werden:

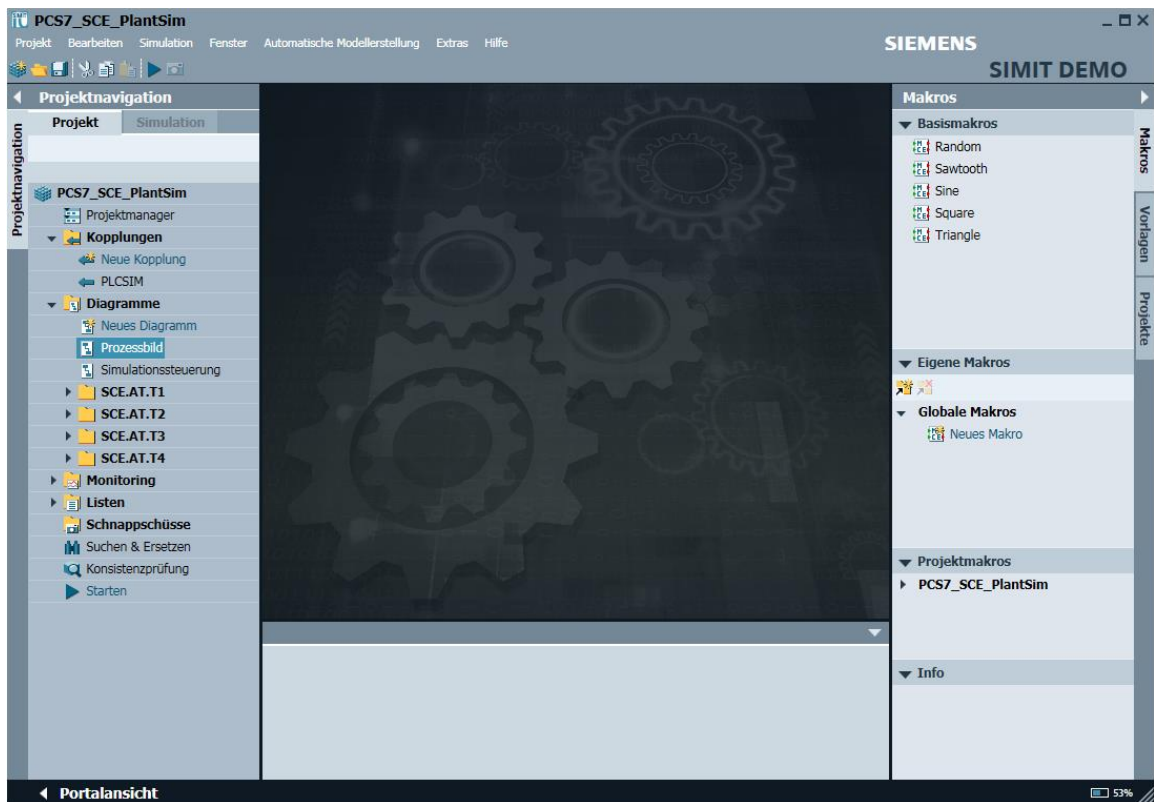
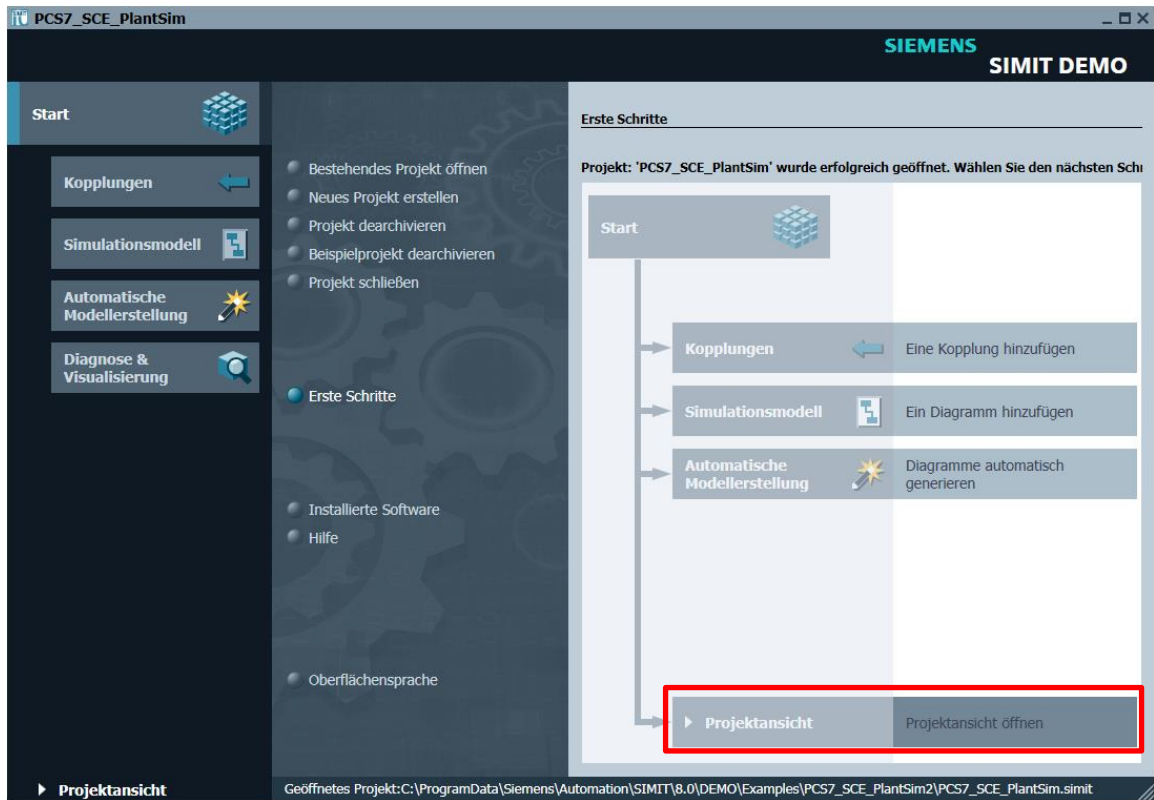
1. Nach S7-PLCSIM wird das Simulationsprogramm gestartet.




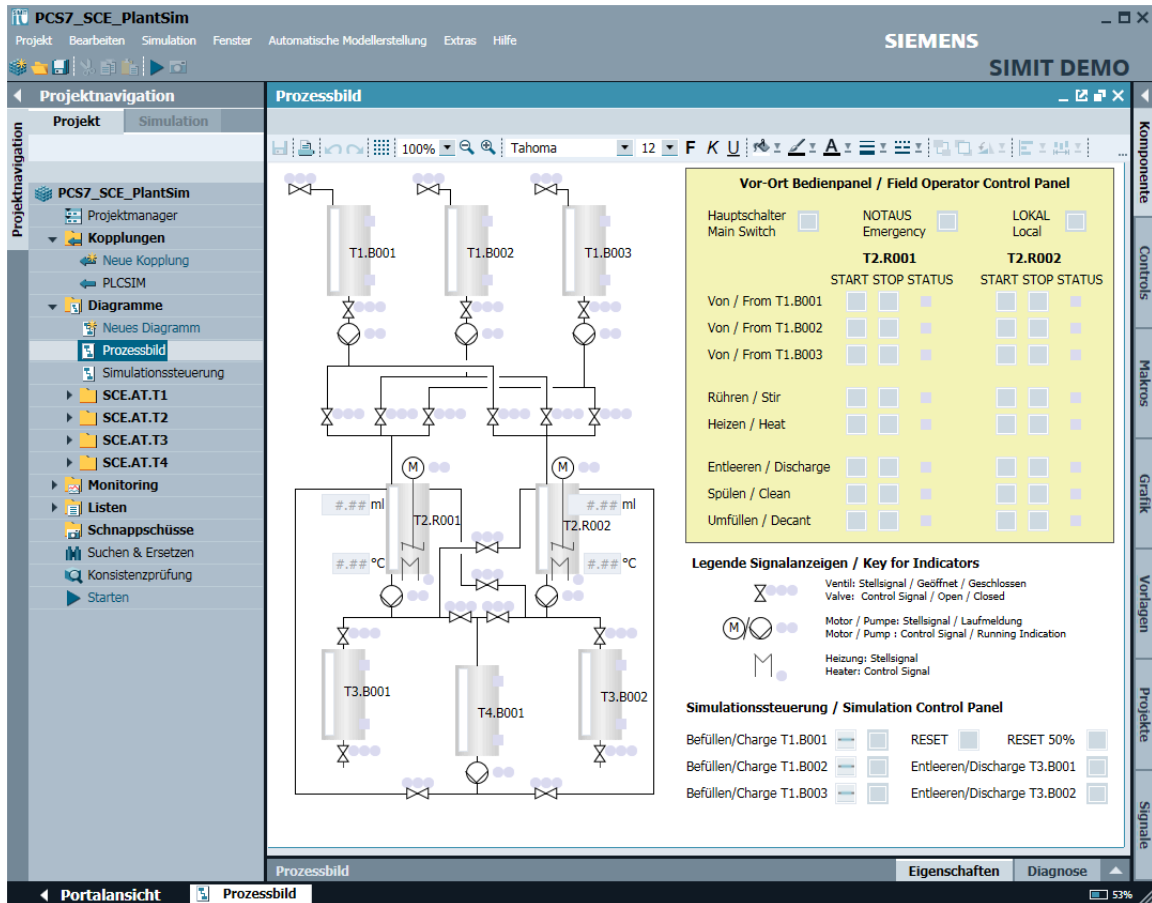
2. Anschließend wird das Simulationsprojekt, wenn noch nicht geschehen, dearchiviert und geöffnet. Nutzen Sie dazu die mitgelieferte Datei ‚p01-04-plantsim-v10-r1905-de.simarc‘.



3. Wechseln Sie in die Projektansicht und öffnen Sie das Prozessbild.



4. Starten Sie die Simulation der Anlage in SIMIT mit dem Button  und bestätigen Sie die Meldung zur beschränkten Laufzeit mit ‚OK‘.



The screenshot shows the SIMIT DEMO software interface. The main window displays a process diagram with tanks (T1.B001, T1.B002, T1.B003, T3.B001, T3.B002, T4.B001), pumps (T2.R001, T2.R002), and valves. The interface includes a project navigation tree on the left, a toolbar at the top, and a right-hand panel with control and simulation options.

Vor-Ort Bedienpanel / Field Operator Control Panel

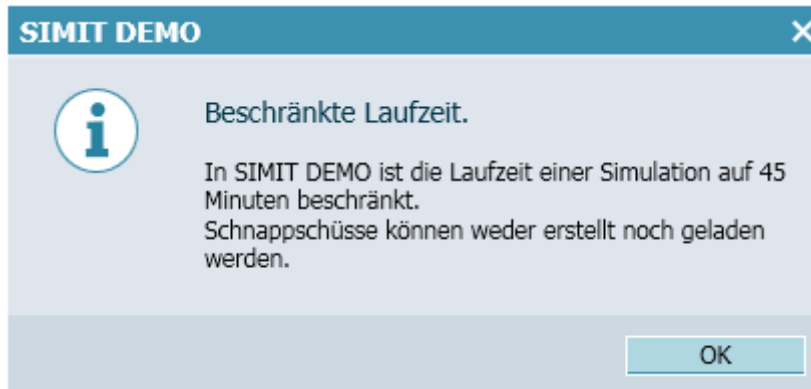
Hauptschalter Main Switch	<input type="checkbox"/>	NOTAUS Emergency	<input type="checkbox"/>	LOKAL Local	<input type="checkbox"/>
		T2.R001		T2.R002	
		START	STOP	STATUS	STATUS
Von / From T1.B001	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Von / From T1.B002	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Von / From T1.B003	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rühren / Stir	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heizen / Heat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entleeren / Discharge	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spülen / Clean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Umfüllen / Decant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Legende Signalanzeigen / Key for Indicators

- Ventil: Stellsignal / Geöffnet / Geschlossen
Valve: Control Signal / Open / Closed
- Motor / Pumpe: Stellsignal / Laufmeldung
Motor / Pump: Control Signal / Running Indication
- Heizung: Stellsignal
Heater: Control Signal

Simulationssteuerung / Simulation Control Panel

Befüllen/Charge T1.B001	<input type="checkbox"/>	RESET	<input type="checkbox"/>	RESET 50%	<input type="checkbox"/>
Befüllen/Charge T1.B002	<input type="checkbox"/>	Entleeren/Discharge T3.B001	<input type="checkbox"/>		
Befüllen/Charge T1.B003	<input type="checkbox"/>	Entleeren/Discharge T3.B002	<input type="checkbox"/>		



SIMIT DEMO ✕

Beschränkte Laufzeit.

In SIMIT DEMO ist die Laufzeit einer Simulation auf 45 Minuten beschränkt. Schnappschüsse können weder erstellt noch geladen werden.

5. Indem Sie von STOP auf RUN-P umstellen, starten Sie die Simulation der SPS in S7-PLCSIM.
6. Die Simulation beginnt, wie in Abbildung 5 dargestellt, mit 75 % gefüllten Edukttanks, alle anderen sind entleert. Dieser Zustand kann jederzeit mit der Option ‚RESET‘ in der Simulationssteuerung wiederhergestellt werden. Die Option ‚RESET 50 %‘ füllt alle Tanks zu 50 %.
7. Zum Testen wird der Motorbaustein, wie im vorherigen Abschnitt 8.7 beschrieben, angesteuert – in der Simulation leuchtet das Stellsignal der Pumpe grün auf.

8. Der simulierte Motoranlauf dauert etwa 2 Sekunden. Danach leuchtet zum einen die Laufanzeige des Motors in der Simulation grün auf, zum anderen wird der Signalpegel für den Binäreingang E 1.3 des Leitsystems gesetzt.
9. Um den Förderweg zu öffnen, muss zudem das Ventil zum Produkttank T3.B001 geöffnet werden. Dieses Ventil wird in der anschließenden Übung über eine geeignete Einzelsteuerfunktion angesprochen. In der Simulation kann bei eingeschalteter Pumpe und offenem Ventil beobachtet werden, wie der Inhalt des Reaktors T2.R001 in den Produkttank T3.B001 gepumpt wird.
10. Über die Simulationssteuerungen können die Produkttanks entleert und die Edukttanks befüllt werden. Beim Befüllen über die Simulationssteuerung ist zunächst die Simulation von dem Leitsystem zu trennen. Dazu wird der Button mit dem waagerechten Strich (☐) einmal angeklickt. Anschließend kann das Ventil mit dem Knopf rechts daneben geöffnet werden. Das Stellsignal leuchtet grün auf, nach etwa einer Sekunde folgt das Signal des Endlagenschalters für die Offen-Position, nach weiteren 5 bis 10 Sekunden sind erste Änderungen im Füllstand sichtbar.
11. Durch ‚RESET‘ und ‚RESET 50 %‘ kann die Simulation jederzeit wieder in einen definierten Zustand gebracht werden.

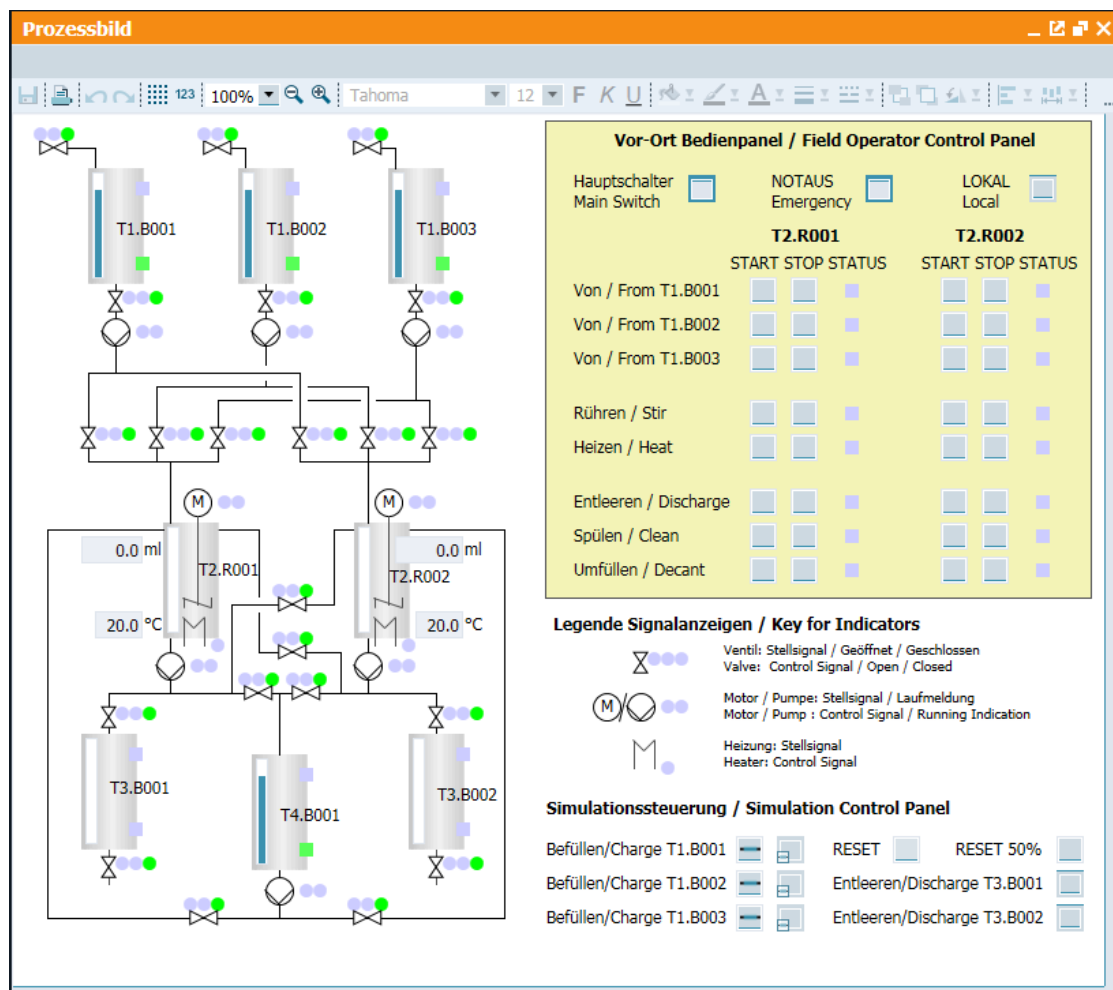


Abbildung 5: Bedienoberfläche der Prozesssimulation

10 Übungen

In den Übungsaufgaben soll Gelerntes aus der Theorie und der Schritt-für-Schritt-Anleitung umgesetzt werden. Hierbei soll das schon vorhandene Multiprojekt aus der Schritt-für-Schritt-Anleitung (p01-04-project-r1905-de.zip) genutzt und erweitert werden. Der Download des Projekts ist beim jeweiligen Modul als Zip-file Projekte im SCE Internet hinterlegt.

Ziel dieser Übung ist es einen CFC zu erstellen, mit dem die Ventile der Anlage gesteuert werden können. Hierbei soll auf das Wissen aus der Schritt-für-Schritt-Anleitung aufgebaut werden, in der ein ähnlicher CFC zur Steuerung des Motors erstellt wurde.

Außerdem wird ein CFC zur Normierung des Füllstandes, also eines analogen Eingangswertes, vom digitalisierten Wert auf den physischen Wert erstellt. Diese Aufgabe erweitert das Wissen, da der CFC ohne ein Template aber trotzdem mit Bausteinen aus der Bibliothek angelegt wird. Dieser CFC wird für das Kapitel Anlagensicherung benötigt.

10.1 Übungsaufgaben

Die folgenden Übungen orientieren sich an der Schritt-für-Schritt-Anleitung. Für jede Übungsaufgabe können die entsprechenden Schritte der Anleitung zur Hilfe genutzt werden.

1. Fügen Sie das Template ‚Valve_Lean‘ als Vorlage in die Messstellentypen ein (analog zu ‚Motor_Lean‘). Diese Vorlage dient der Implementierung der Ventile.
2. Im Planordner ‚Produkttank B001‘ der Teilanlage T3_Produktspeicher soll nun eine Objektinstanz des Ventiltemplates eingefügt und in A1T3X001 umbenannt werden. Öffnen Sie den CFC-Plan und passen Sie auch den Namen des Bausteins ‚VlvL‘ an. Schließen Sie nun die Rückmeldungs- und Steuersignale an (siehe Abbildung 7 und Tabelle 5).
3. Laden und Testen Sie jetzt Ihre Implementierung mit dem SIMIT-Modell. Dafür sollten die folgenden Anschlüsse sichtbar sein: ‚ModLiOp‘, ‚AutModOp‘, ‚ManModOp‘, ‚OpenAut‘, ‚CloseAut‘, ‚MonDynErr‘ und ‚RstOp‘.
4. Zum Einbinden des analogen Füllstandsensors A1T2L001 (siehe Abbildung 7) erstellen Sie einen neuen CFC im Planordner ‚Reaktor R001‘. Benennen Sie diesen mit A1T2L001 und öffnen Sie ihn. Ziehen Sie den Baustein ‚Pcs7AnIn‘ (FB1869) per Drag&Drop aus dem Katalog in den CFC. Wählen Sie dafür im linken Rahmen das Register Bibliotheken an und nutzen hier entweder die Suchfunktion ganz unten oder öffnen PCS 7 AP Library V90/Blocks+ Templates\Blocks\Channel. Sobald der Baustein eingefügt wurde, benennen Sie ihn in Stand_A1T2L001 um.
5. Parametrieren Sie den Baustein ‚Pcs7AnIn‘ indem Sie die Eingangswerte ‚Scale‘ auf High = 0.0 und Low = 1158.0 und ‚PV_InUni‘ auf 1040 (für die Einheit ml) setzen. Verbinden Sie den Eingang ‚PV_In‘ des Bausteins ‚Pcs7AnIn‘ mit dem Symbol für den Füllstandwert (siehe Tabelle 5) des Reaktors R001.

6. Realisieren Sie jetzt den oberen und unteren Füllstandsensoren vom Produktbehälter B001. Legen Sie dafür einen CFC-Plan im Planordner ‚Produkttank B001‘ an und benennen Sie ihn mit A1T3L001. Öffnen Sie den Plan und fügen Sie zweimal den Baustein ‚Pcs7Diln‘ aus dem Katalog (analog zu Aufgabe 5) hinzu. Benennen Sie den einen Baustein mit A1T3L001_LSA+ und den anderen mit A1T3L001_LSA-. Verschalten Sie jeweils ‚PV_In‘ mit den Sensorsignalen (siehe Tabelle 4).
7. Legen Sie nun CFC-Pläne für den Hauptschalter, den NOTAUS-Schalter sowie den Schalter für die lokale Bedienung an. Dazu wird wie in Aufgabe 6 jeweils ein CFC-Plan für A1H001, A1H002 und A1H003 (siehe auch Abbildung 6) im Planordner ‚Mehrzweckanlage‘ angelegt, der Baustein ‚Pcs7Diln‘ hinzugefügt, benannt und mit dem jeweiligen Operanden nach ‚PV_In‘ verschaltet.

Aufgabe	Symbol	Adresse	Datentyp	Kommentar	
	2	A1.T3.A1T3X001.XV.C	A 0.6	BOOL	Auf/Zu-Ventil Zufluss Produkttank B001 Stellsignal
		A1.T3.A1T3X001.GO+-O+	E 67.4	BOOL	Auf/Zu-Ventil Zufluss Produkttank B001 Rückmeldung auf/ein
		A1.T3.A1T3X001.GO+-O-	E 67.5	BOOL	Auf/Zu-Ventil Zufluss Produkttank B001 Rückmeldung zu
	5	A1.T2.A1T2L001.LISA+.M	EW 72	WORD	Füllstandswert Reaktor R001
	6	A1.T3.A1T3L001.LSA+.SA+	E 70.6	BOOL	Füllstandsüberwachung Produkttank B001 Schaltpunkt H
		A1.T3.A1T3L001.LSA-.SA-	E 70.7	BOOL	Füllstandsüberwachung Produkttank B001 Schaltpunkt L
	7	A1.A1H001.HS+-.START	E 0.0	BOOL	Mehrzweckanlage einschalten
		A1.A1H002.HS+-.OFF	E 0.1	BOOL	NOTAUS aktivieren (Schließer)
A1.A1H003.HS+-.LOC		E 0.2	BOOL	Lokale Bedienung aktivieren	

Tabelle 5: Die Symbole für die Realisierung der Ventilsteuerung und des Füllstandsensors

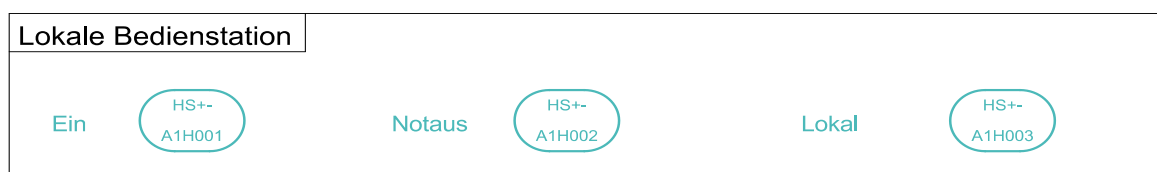


Abbildung 6: Ausschnitt aus der lokalen Bedienstation

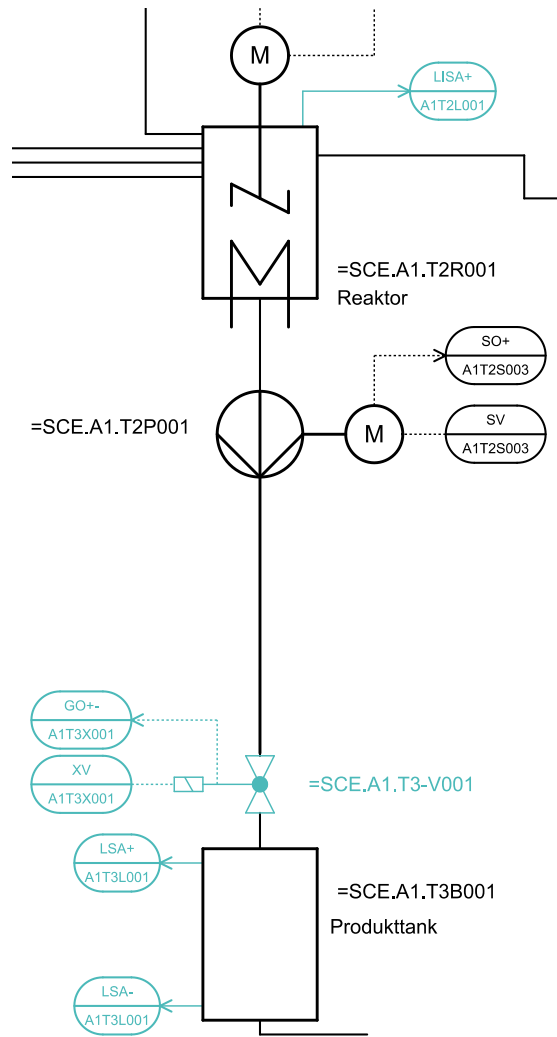


Abbildung 7: Ausschnitt aus dem R&I-Fließbild

10.2 Checkliste – Übung

Die nachfolgende Checkliste hilft den Studierenden selbstständig zu überprüfen, ob alle Arbeitsschritte der Übung sorgfältig abgearbeitet wurden und ermöglicht eigenständig das Modul erfolgreich abzuschließen.

Nr.	Beschreibung	Geprüft
1	Valve_Lean in Messstellentypen	
2	Produkttank B001\ A1T3X001 angelegt, konfiguriert und getestet	
3	Reaktor R001\ A1T2L001 angelegt und konfiguriert	
4	Produkttank B001\A1T3L001 angelegt und konfiguriert	
5	A1H001, A1H002 und A1H003 angelegt und konfiguriert	
6	Projekt erfolgreich archiviert	

Tabelle 6: Checkliste für Übungen

11 Weiterführende Information

Zur Einarbeitung bzw. Vertiefung finden Sie als Orientierungshilfe weiterführende Informationen, wie z.B.: Getting Started, Videos, Tutorials, Apps, Handbücher, Programmierleitfaden und Trial Software/Firmware, unter nachfolgendem Link:

[siemens.de/sce/pcs7](https://www.siemens.de/sce/pcs7)

Voransicht „Weiterführende Informationen“

Getting Started, Videos, Tutorials, Apps, Handbücher, Trial Software/Firmware

- > SIMATIC PCS 7 Überblick
- > SIMATIC PCS 7 Videos
- > Getting Started
- > Applikationsbeispiele
- > Download Software/Firmware
- > SIMATIC PCS 7 Website
- > SIMATIC S7-400 Website

Weitere Informationen

Siemens Automation Cooperates with Education

[siemens.de/sce](https://www.siemens.de/sce)

Siemens SIMATIC PCS 7

[siemens.de/pcs7](https://www.siemens.de/pcs7)

SCE Lehrunterlagen

[siemens.de/sce/module](https://www.siemens.de/sce/module)

SCE Trainer Pakete

[siemens.de/sce/tp](https://www.siemens.de/sce/tp)

SCE Kontakt Partner

[siemens.de/sce/contact](https://www.siemens.de/sce/contact)

Digital Enterprise

[siemens.de/digital-enterprise](https://www.siemens.de/digital-enterprise)

Industrie 4.0

[siemens.de/zukunft-der-industrie](https://www.siemens.de/zukunft-der-industrie)

Totally Integrated Automation (TIA)

[siemens.de/tia](https://www.siemens.de/tia)

TIA Portal

[siemens.de/tia-portal](https://www.siemens.de/tia-portal)

SIMATIC Controller

[siemens.de/controller](https://www.siemens.de/controller)

SIMATIC Technische Dokumentation

[siemens.de/simatic-doku](https://www.siemens.de/simatic-doku)

Industry Online Support

support.industry.siemens.com

Katalog- und Bestellsystem Industry Mall

mall.industry.siemens.com

Siemens

Digital Industries, FA

Postfach 4848

90026 Nürnberg

Deutschland

Änderungen und Irrtümer vorbehalten

© Siemens 2020

[siemens.de/sce](https://www.siemens.de/sce)