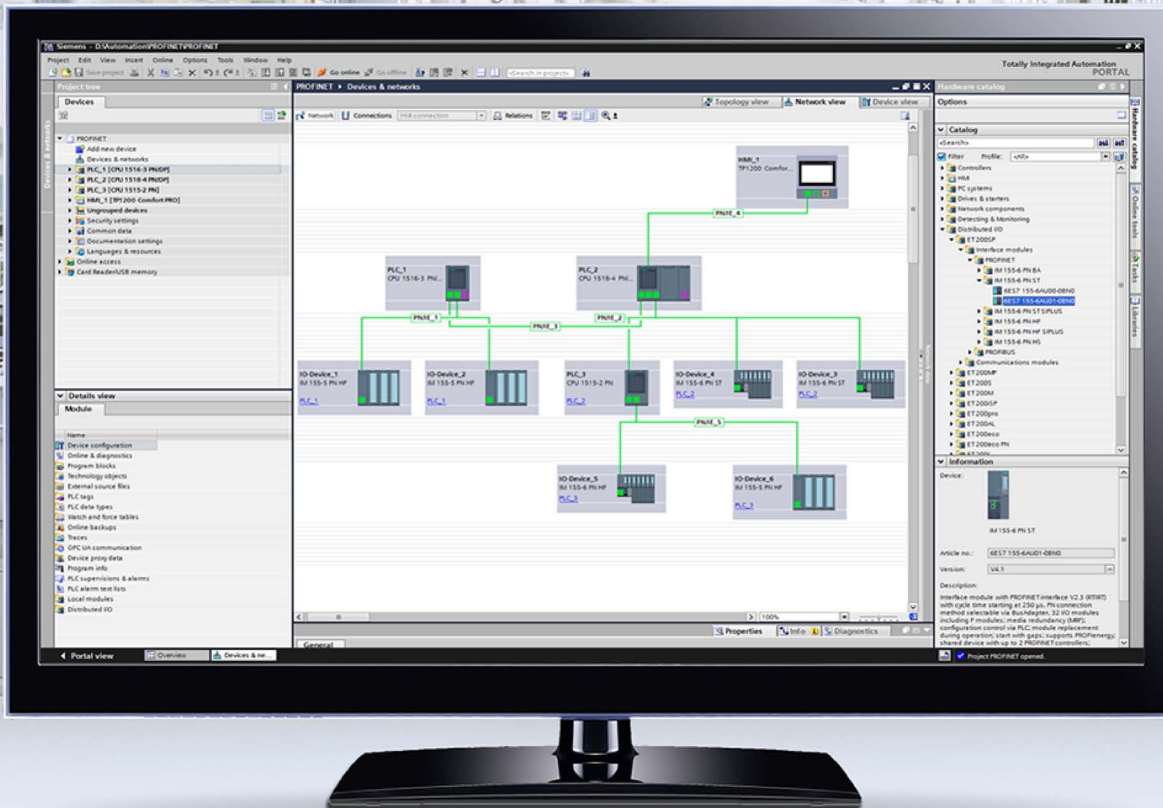


SIEMENS



Funktionshandbuch

SIMATIC

S7-1500, ET 200MP, ET 200SP,
ET 200AL, ET 200pro

Kommunikation

Ausgabe

12/2017

support.industry.siemens.com

SIMATIC

S7-1500, ET 200MP, ET 200SP, ET 200AL, ET 200pro Kommunikation

Funktionshandbuch

Vorwort

Wegweiser Dokumentation

1

Produktübersicht

2

Kommunikationsdienste

3

PG-Kommunikation

4

HMI-Kommunikation

5

Open User Communication

6

S7-Kommunikation

7

Punkt-zu-Punkt-Kopplung

8

OPC UA-Kommunikation

9

Routing

10

Verbindungsressourcen

11

Diagnose von Verbindungen

12




Industrial Ethernet Security
mit CP 1543-1

13

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.

Vorwort

Zweck der Dokumentation

Das vorliegende Funktionshandbuch vermittelt einen Überblick über die Kommunikationsmöglichkeiten, die CPUs, Kommunikationsmodule und -prozessoren, und PC-Systeme der Systeme SIMATIC S7-1500, ET 200MP, ET 200SP, ET 200AL und ET 200pro bieten. Im vorliegenden Funktionshandbuch wird die verbindungsorientierte, asynchrone Kommunikation beschrieben.

Die Dokumentation behandelt im Einzelnen:

- Überblick der Kommunikationsdienste
- Eigenschaften der Kommunikationsdienste
- Anwendertätigkeiten zum Einrichten der Kommunikationsdienste im Überblick

Erforderliche Grundkenntnisse

Zum Verständnis des Funktionshandbuchs sind folgende Kenntnisse erforderlich:

- allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik
- Kenntnisse des Industrieautomatisierungssystems SIMATIC
- Kenntnisse im Umgang mit STEP 7 (TIA Portal)

Gültigkeitsbereich der Dokumentation

Die vorliegende Dokumentation gilt als Grundlagendokumentation für alle Produkte der Systeme SIMATIC S7-1500, ET 200MP, ET 200SP, ET 200AL und ET 200pro. Die Produktdokumentationen bauen auf dieser Dokumentation auf.

Was ist neu im Funktionshandbuch Kommunikation, Ausgabe 12/2017 gegenüber Ausgabe 09/2016

Was ist neu?		Was ist der Kundennutzen?	Wo finden Sie die Informationen?
Neue Inhalte	OPC UA Companion Specification	Über OPC UA Companion Specification lassen sich Methoden einheitlich und herstellerunabhängig spezifizieren. Über diese spezifizierten Methoden können Geräte der verschiedensten Hersteller einfacher in die Anlage und in Produktionsabläufe integriert werden.	Kap. OPC UA-Informationsmodelle nutzen (Seite 218)
	Gesicherte Verbindung zu einem Mailserver über die Schnittstelle der CPU	Sie können eine gesicherte Verbindung zu einem Mailserver aufbauen ohne zusätzliche Hardware.	Kap. Secure OUC über E-Mail (Seite 116)
	Gesicherte Kommunikation über Modbus TCP	Sie können gesicherte TCP-Verbindungen zwischen einem Modbus TCP-Client und einem Modbus TCP-Server aufbauen.	Kap. Secure OUC mit Modbus TCP (Seite 115)

Was ist neu im Funktionshandbuch Kommunikation, Ausgabe 09/2016 gegenüber Ausgabe 12/2014

Was ist neu?		Was ist der Kundennutzen?	Wo finden Sie die Informationen?
Neue Inhalte	OPC UA-Server	<p>OPC UA ist ein einheitlicher Standard zum Datenaustausch und ist unabhängig von bestimmten Betriebssystemplattformen. OPC UA nutzt integrierte Sicherheitsmechanismen auf verschiedenen Automatisierungssystemen, z. B. beim Datenaustausch, auf Anwendungsebene, zur Legitimation des Anwenders.</p> <p>Der OPC UA Server stellt zahlreiche Daten bereit:</p> <ul style="list-style-type: none"> • Werte von PLC-Variablen, auf die Clients zugreifen dürfen • Datentypen dieser PLC-Variablen • Angaben zum OPC UA Server selbst und zur CPU <p>Clients können sich dadurch einen Überblick über den Variablenhaushalt verschaffen und Werte einlesen und schreiben.</p>	Kap. OPC UA-Kommunikation (Seite 135)
	Secure Open User Communication	Sicherer Datenaustausch mit anderen Geräten.	Kap. Secure Open User Communication (Seite 98)
	Zertifikate-Handling in STEP 7	<p>Sie können in STEP 7 Zertifikate verwalten für die folgenden Anwendungen:</p> <ul style="list-style-type: none"> • OPC UA Server • Secure Open User Communication • Webserver der CPU 	Kap. Verwalten von Zertifikaten mit STEP 7 (Seite 47)

Was ist neu?		Was ist der Kundennutzen?	Wo finden Sie die Informationen?
	SNMP für die CPU deaktivieren	Sie können für die CPU SNMP deaktivieren. Das kann unter bestimmten Voraussetzungen sinnvoll sein, z. B. wenn die Sicherheitsrichtlinien in Ihrem Netzwerk kein SNMP zulassen.	Kap. SNMP deaktivieren (Seite 61)

Konventionen

STEP 7: Zur Bezeichnung der Projektier- und Programmiersoftware verwenden wir in der vorliegenden Dokumentation "STEP 7" als Synonym für "STEP 7 ab V12 (TIA Portal)".

Mit "S7-1500 CPUs" sind auch die CPU-Varianten S7-1500F, S7-1500T, S7-1500TF, S7-1500C sowie S7-1500pro CPUs, ET200SP CPUs sowie der SIMATIC S7-1500 SW Controller gemeint.

Die vorliegende Dokumentation enthält Abbildungen zu den beschriebenen Geräten. Die Abbildungen können vom gelieferten Gerät in Einzelheiten abweichen.

Beachten Sie außerdem die folgendermaßen gekennzeichneten Hinweise:

Hinweis

Ein Hinweis enthält wichtige Informationen zum Produkt, zur Handhabung des Produkts oder zu dem Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Siehe auch

PRODIS (<http://www.siemens.com/simatic-tech-doku-portal>)

Katalog (<http://mall.industry.siemens.com>)

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 unter (<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 unter (<http://www.siemens.com/industrialsecurity>).

Siemens Industry Online Support

Aktuelle Informationen erhalten Sie schnell und einfach zu folgenden Themen:

- **Produkt-Support**

Alle Informationen und umfangreiches Know-how rund um Ihr Produkt, Technische Daten, FAQs, Zertifikate, Downloads und Handbücher.

- **Anwendungsbeispiele**

Tools und Beispiele zur Lösung Ihrer Automatisierungsaufgabe – außerdem Funktionsbausteine, Performance-Aussagen und Videos.

- **Services**

Informationen zu Industry Services, Field Services, Technical Support, Ersatzteilen und Trainingsangeboten.

- **Foren**

Für Antworten und Lösungen rund um die Automatisierungstechnik.

- **mySupport**

Ihr persönlicher Arbeitsbereich im Siemens Industry Online Support für Benachrichtigungen, Support-Anfragen und konfigurierbare Dokumente.

Diese Informationen bietet Ihnen der Siemens Industry Online Support im Internet (<http://www.siemens.com/automation/service&support>).

Industry Mall

Die Industry Mall ist das Katalog- und Bestellsystem der Siemens AG für Automatisierungs- und Antriebslösungen auf Basis von Totally Integrated Automation (TIA) und Totally Integrated Power (TIP).

Kataloge zu allen Produkten der Automatisierungs- und Antriebstechnik finden Sie im Internet (<https://mall.industry.siemens.com>).

Inhaltsverzeichnis

	Vorwort	3
1	Wegweiser Dokumentation	11
2	Produktübersicht.....	16
3	Kommunikationsdienste	21
3.1	Kommunikationsmöglichkeiten im Überblick	21
3.2	Kommunikationsprotokolle und verwendete Portnummern bei Ethernet-Kommunikation	24
3.3	Verbindungsressourcen im Überblick	30
3.4	Einrichten einer Verbindung.....	31
3.5	Datenkonsistenz.....	35
3.6	Secure Communication.....	38
3.6.1	Grundlagen zu Secure Communication	38
3.6.2	Vertraulichkeit durch Verschlüsselung.....	40
3.6.3	Authentizität und Integrität durch Signaturen.....	43
3.6.4	Verwalten von Zertifikaten mit STEP 7	47
3.6.5	Beispiele zum Verwalten von Zertifikaten	51
3.6.6	Beispiel: HTTP over TLS	57
3.7	SNMP	61
3.7.1	SNMP deaktivieren	61
3.7.2	Beispiel: SNMP für eine CPU 1516-3 PN/DP deaktivieren	62
4	PG-Kommunikation.....	65
5	HMI-Kommunikation	68
6	Open User Communication	70
6.1	Open User Communication im Überblick.....	70
6.2	Protokolle für Open User Communication	71
6.3	Anweisungen für Open User Communication.....	73
6.4	Open User Communication mit Adressierung über Domainnamen.....	77
6.5	Open User Communication über TCP, ISO-on-TCP, UDP und ISO einrichten	80
6.6	Kommunikation über FDL einrichten.....	87
6.7	Kommunikation über Modbus TCP einrichten	90
6.8	Kommunikation über E-Mail einrichten	93
6.9	Kommunikation über FTP einrichten.....	94
6.10	Auf- und Abbau von Kommunikationsbeziehungen.....	97

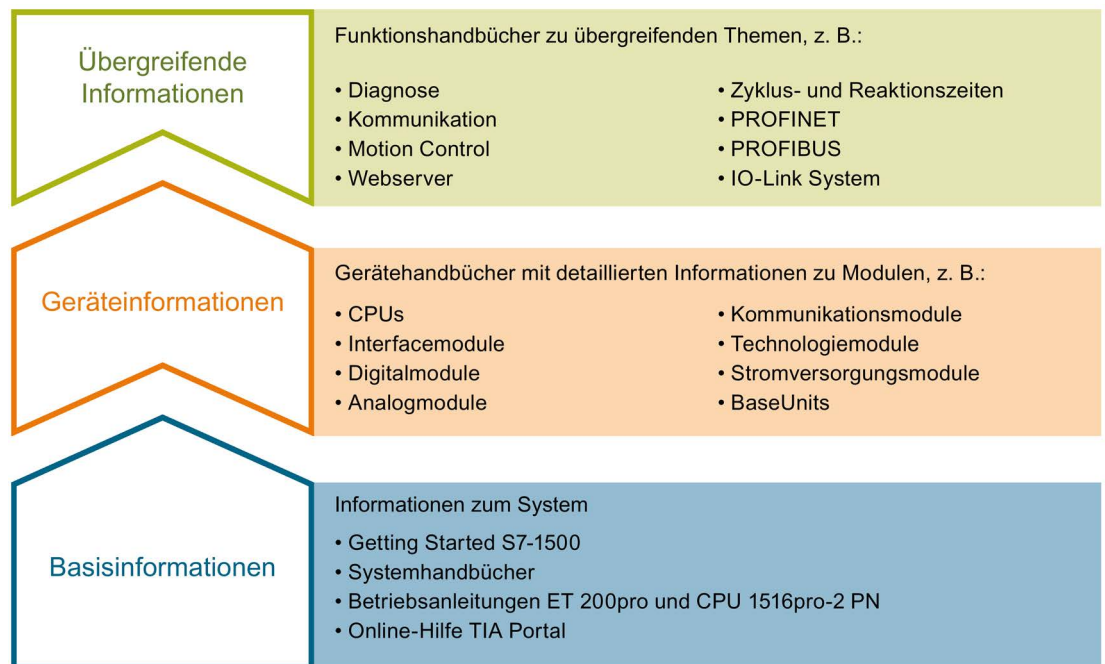
6.11	Secure Open User Communication	98
6.11.1	Secure OUC von einer S7-1500 CPU als TLS-Client zu einem Fremd-PLC (TLS-Server)	98
6.11.2	Secure OUC von einer S7-1500 CPU als TLS-Server zu einem Fremd-PLC (TLS-Client)	101
6.11.3	Secure OUC zwischen zwei S7-1500 CPUs	105
6.11.4	Secure OUC über CP-Schnittstelle	109
6.11.5	Secure OUC mit Modbus TCP	115
6.11.6	Secure OUC über E-Mail	116
7	S7-Kommunikation	121
8	Punkt-zu-Punkt-Kopplung	130
9	OPC UA-Kommunikation	135
9.1	Wissenswertes zu OPC UA	135
9.1.1	OPC UA und Industrie 4.0	135
9.1.2	Aufbau der Beschreibung	135
9.1.3	Allgemeine Eigenschaften von OPC UA	137
9.1.4	Von der klassischen OPC-Schnittstelle zu OPC UA	139
9.1.5	Der OPC UA-Server der S7-1500 CPUs	139
9.1.6	Adressierung von Knoten	141
9.1.7	Mapping von Datentypen	145
9.1.8	Endpunkte der OPC UA-Server	147
9.1.9	Verhalten des OPC UA-Servers im Betrieb	149
9.1.10	Wissenswertes zu OPC UA-Clients	150
9.2	Security bei OPC UA	155
9.2.1	Security-Einstellungen	155
9.2.2	Zertifikate gemäß X.509 der ITU	156
9.2.3	Zertifikate bei OPC UA	159
9.2.4	Selbst-signierte Zertifikate erzeugen	160
9.2.5	PKI-Schlüsselpaare und Zertifikate selbst erzeugen	162
9.2.6	Nachrichten gesichert übertragen	165

9.3	S7-1500 CPU als OPC UA-Server nutzen.....	168
9.3.1	Zugriff auf PLC-Variablen projektieren	168
9.3.1.1	Schreib- und Leserechte verwalten	168
9.3.1.2	Schreib- und Leserechte für kompletten DB verwalten	170
9.3.1.3	Zugriffsmöglichkeiten auf Daten des OPC UA-Servers.....	171
9.3.1.4	XML-Datei mit den freigegebenen PLC-Variablen exportieren	172
9.3.2	OPC UA-Server der S7-1500 CPU konfigurieren	173
9.3.2.1	OPC UA-Server aktivieren	173
9.3.2.2	Zugang zum OPC UA-Server	175
9.3.2.3	Allgemeine Einstellungen des OPC UA-Servers	177
9.3.2.4	Einstellungen des Servers für Subscriptions	178
9.3.2.5	Handling der Client- und Server-Zertifikate	180
9.3.2.6	Handling der Client-Zertifikate der S7-1500 CPU.....	185
9.3.2.7	Server-Zertifikate mit STEP 7 erzeugen	187
9.3.2.8	Authentifizierung des Benutzers	190
9.3.2.9	Benutzer und Rollen mit OPC UA-Funktionsrechten.....	192
9.3.2.10	Lizenzen für den OPC UA-Server.....	193
9.3.3	Methoden auf dem OPC UA-Server bereitstellen.....	194
9.3.3.1	Server-Methoden	194
9.3.4	OPC UA-Server-Anweisungen für die Implementierung von Methoden.....	199
9.3.4.1	OPC-UA-ServerMethodPre	199
9.3.4.2	OPC-UA-ServerMethodPost.....	202
9.3.4.3	Beispielprogramm zum Bereitstellen einer Methode für OPC UA-Clients.....	205
9.3.4.4	Fehlercodes	209
9.3.5	OPC UA-Informationsmodelle nutzen.....	218
9.3.5.1	OPC UA Companion Spezifikationen verwenden.....	218
9.3.5.2	Schreib- und Leserechte für CPU-Variablen koordinieren	231
9.3.5.3	Hinweise zu Mengengerüsten bei Nutzung von Server-Schnittstellen	233
10	Routing.....	235
10.1	S7-Routing	235
10.2	Datensatz-Routing	240
11	Verbindungsressourcen	242
11.1	Verbindungsressourcen einer Station.....	242
11.2	Belegung von Verbindungsressourcen.....	246
11.3	Anzeige der Verbindungsressourcen.....	250
12	Diagnose von Verbindungen	254
13	Industrial Ethernet Security mit CP 1543-1	258
13.1	Firewall.....	260
13.2	Logging	261
13.3	NTP-Client.....	261
13.4	SNMP	262
13.5	VPN.....	262
	Glossar	263
	Index	275

Wegweiser Dokumentation

Die Dokumentation für das Automatisierungssystem SIMATIC S7-1500, für die auf SIMATIC S7-1500 basierende CPU 1516pro-2 PN und die Dezentralen Peripheriesysteme SIMATIC ET 200MP, ET 200SP und ET 200AL gliedert sich in drei Bereiche.

Die Aufteilung bietet Ihnen die Möglichkeit, gezielt auf die gewünschten Inhalte zuzugreifen.



Basisinformationen

Systemhandbücher und Getting Started beschreiben ausführlich die Projektierung, Montage, Verdrahtung und Inbetriebnahme der Systeme SIMATIC S7-1500, ET 200MP, ET 200SP und ET 200AL, für CPU 1516pro-2 PN nutzen Sie die entsprechenden Betriebsanleitungen. Die Online-Hilfe von STEP 7 unterstützt Sie bei der Projektierung und Programmierung.

Geräteinformationen

Gerätehandbücher enthalten eine kompakte Beschreibung der modulspezifischen Informationen wie Eigenschaften, Anschlussbilder, Kennlinien, Technische Daten.

Übergreifende Informationen

In den Funktionshandbüchern finden Sie ausführliche Beschreibungen zu übergreifenden Themen, z. B. Diagnose, Kommunikation, Motion Control, Webserver, OPC UA.

Die Dokumentation finden Sie zum kostenlosen Download im Internet (<https://support.industry.siemens.com/cs/ww/de/view/109742705>).

Änderungen und Ergänzungen zu den Handbüchern werden in Produktinformationen dokumentiert.

Sie finden die Produktinformationen im Internet:

- S7-1500/ET 200MP (<https://support.industry.siemens.com/cs/de/de/view/68052815>)
- ET 200SP (<https://support.industry.siemens.com/cs/de/de/view/73021864>)
- ET 200AL (<https://support.industry.siemens.com/cs/de/de/view/99494757>)

Manual Collections

Die Manual Collections beinhalten die vollständige Dokumentation zu den Systemen zusammengefasst in einer Datei.

Sie finden die Manual Collections im Internet:

- S7-1500/ET 200MP (<https://support.industry.siemens.com/cs/ww/de/view/86140384>)
- ET 200SP (<https://support.industry.siemens.com/cs/ww/de/view/84133942>)
- ET 200AL (<https://support.industry.siemens.com/cs/ww/de/view/95242965>)

"mySupport"

Mit "mySupport", Ihrem persönlichen Arbeitsbereich, machen Sie das Beste aus Ihrem Industry Online Support.

In "mySupport" können Sie Filter, Favoriten und Tags ablegen, CAx-Daten anfordern und sich im Bereich Dokumentation Ihre persönliche Bibliothek zusammenstellen. Des Weiteren sind in Support-Anfragen Ihre Daten bereits vorausgefüllt und Sie können sich jederzeit einen Überblick über Ihre laufenden Anfragen verschaffen.

Um die volle Funktionalität von "mySupport" zu nutzen, müssen Sie sich einmalig registrieren.

Sie finden "mySupport" im Internet (<https://support.industry.siemens.com/My/ww/de/>).

"mySupport" - Dokumentation

In "mySupport" haben Sie im Bereich Dokumentation die Möglichkeit ganze Handbücher oder nur Teile daraus zu Ihrem eigenen Handbuch zu kombinieren.

Sie können das Handbuch als PDF-Datei oder in einem nachbearbeitbaren Format exportieren.

Sie finden "mySupport" - Dokumentation im Internet (<http://support.industry.siemens.com/My/ww/de/documentation>).

"mySupport" - CAx-Daten

In "mySupport" haben Sie im Bereich CAx-Daten die Möglichkeit auf aktuelle Produktdaten für Ihr CAx- oder CAe-System zuzugreifen.

Mit wenigen Klicks konfigurieren Sie Ihr eigenes Download-Paket.

Sie können dabei wählen:

- Produktbilder, 2D-Maßbilder, 3D-Modelle, Geräteschaltpläne, EPLAN-Makrodateien
- Handbücher, Kennlinien, Bedienungsanleitungen, Zertifikate
- Produktstammdaten

Sie finden "mySupport" - CAx-Daten im Internet
(<http://support.industry.siemens.com/my/ww/de/CAxOnline>).

Anwendungsbeispiele

Die Anwendungsbeispiele unterstützen Sie mit verschiedenen Tools und Beispielen bei der Lösung Ihrer Automatisierungsaufgaben. Dabei werden Lösungen im Zusammenspiel mehrerer Komponenten im System dargestellt - losgelöst von der Fokussierung auf einzelne Produkte.

Sie finden die Anwendungsbeispiele im Internet
(<https://support.industry.siemens.com/sc/ww/de/sc/2054>).

TIA Selection Tool

Mit dem TIA Selection Tool können Sie Geräte für Totally Integrated Automation (TIA) auswählen, konfigurieren und bestellen.

Es ist der Nachfolger des SIMATIC Selection Tools und fasst die bereits bekannten Konfiguratoren für die Automatisierungstechnik in einem Werkzeug zusammen.

Mit dem TIA Selection Tool erzeugen Sie aus Ihrer Produktauswahl oder Produktkonfiguration eine vollständige Bestellliste.

Sie finden das TIA Selection Tool im Internet
(<http://w3.siemens.com/mcms/topics/de/simatic/tia-selection-tool>).

SIMATIC Automation Tool

Mit dem SIMATIC Automation Tool können Sie unabhängig vom TIA Portal gleichzeitig an verschiedenen SIMATIC S7-Stationen Inbetriebsetzungs- und Servicetätigkeiten als Massenoperation ausführen.

Das SIMATIC Automation Tool bietet eine Vielzahl von Funktionen:

- Scannen eines PROFINET/Ethernet Anlagennetzes und Identifikation aller verbundenen CPUs
- Adresszuweisung (IP, Subnetz, Gateway) und Stationsname (PROFINET Device) zu einer CPU
- Übertragung des Datums und der auf UTC-Zeit umgerechneten PG/PC-Zeit auf die Baugruppe
- Programm-Download auf CPU
- Betriebsartenumstellung RUN/STOP
- CPU-Lokalisierung mittels LED-Blinken
- Auslesen von CPU-Fehlerinformation
- Lesen des CPU Diagnosepuffers
- Rücksetzen auf Werkseinstellungen
- Firmwareaktualisierung der CPU und angeschlossener Module

Sie finden das SIMATIC Automation Tool im Internet (<https://support.industry.siemens.com/cs/ww/de/view/98161300>).

PRONETA

Mit SIEMENS PRONETA (PROFINET Netzwerk-Analyse) analysieren Sie im Rahmen der Inbetriebnahme das Anlagennetz. PRONETA verfügt über zwei Kernfunktionen:

- Die Topologie-Übersicht scannt selbsttätig das PROFINET und alle angeschlossenen Komponenten.
- Der IO-Check ist ein schneller Test der Verdrahtung und des Modulausbaus einer Anlage.

Sie finden SIEMENS PRONETA im Internet (<https://support.industry.siemens.com/cs/ww/de/view/67460624>).

SINETPLAN

SINETPLAN, der Siemens Network Planner, unterstützt Sie als Planer von Automatisierungsanlagen und -netzwerken auf Basis von PROFINET. Das Tool erleichtert Ihnen bereits in der Planungsphase die professionelle und vorausschauende Dimensionierung Ihrer PROFINET-Installation. Weiterhin unterstützt Sie SINETPLAN bei der Netzwerkoptimierung und hilft Ihnen, Netzwerkressourcen bestmöglich auszuschöpfen und Reserven einzuplanen. So vermeiden Sie Probleme bei der Inbetriebnahme oder Ausfälle im Produktivbetrieb schon im Vorfeld eines geplanten Einsatzes. Dies erhöht die Verfügbarkeit der Produktion und trägt zur Verbesserung der Betriebssicherheit bei.

Die Vorteile auf einen Blick

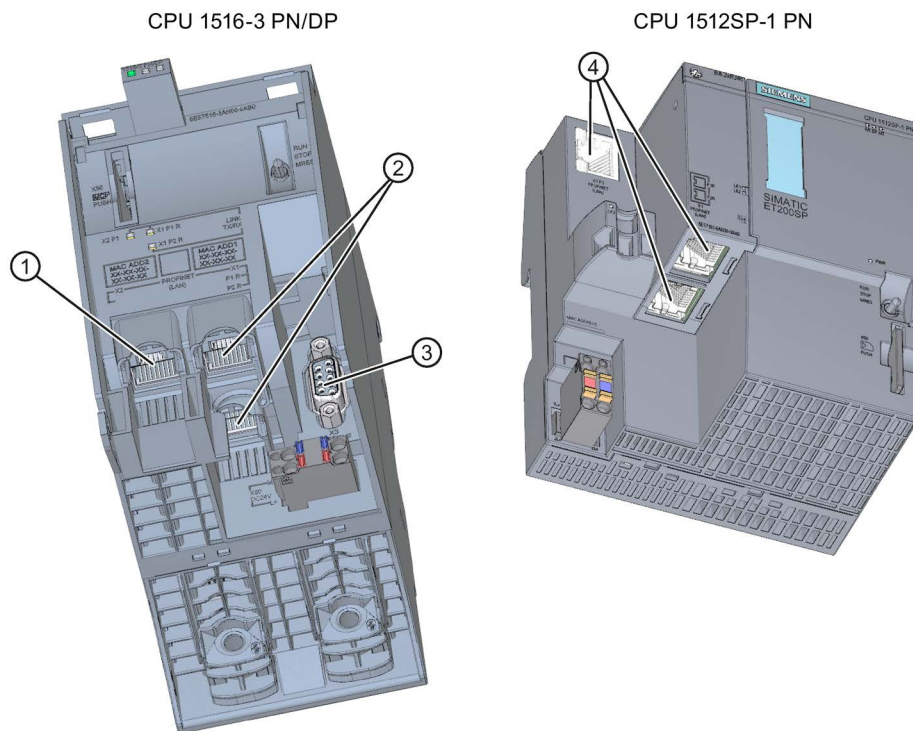
- Netzwerkoptimierung durch portgranulare Berechnung der Netzwerklast
- höhere Produktionsverfügbarkeit durch Onlinescan und Verifizierung bestehender Anlagen
- Transparenz vor Inbetriebnahme durch Import und Simulierung vorhandener STEP7 Projekte
- Effizienz durch langfristige Sicherung vorhandener Investitionen und optimale Ausschöpfung der Ressourcen

Sie finden SINETPLAN im Internet (<https://www.siemens.com/sinetplan>).

CPUs, Kommunikationsmodule und -prozessoren und PC-Systeme der Systeme S7-1500, ET 200MP, ET 200SP, ET 200pro und ET 200AL bieten Ihnen Schnittstellen für die Kommunikation über PROFINET, PROFIBUS und Punkt-zu-Punkt-Kopplung.

CPUs, Kommunikationsmodule und Kommunikationsprozessoren

PROFINET- und PROFIBUS DP-Schnittstellen sind in die CPUs S7-1500 integriert. Zum Beispiel verfügt die CPU 1516-3 PN/DP über zwei PROFINET- und eine PROFIBUS DP-Schnittstelle. Weitere PROFINET- und PROFIBUS DP-Schnittstellen stehen über Kommunikationsmodule (CM) und Kommunikationsprozessoren (CP) zur Verfügung.



- ① PROFINET-Schnittstelle (X2) mit 1 Port
- ② PROFINET-Schnittstelle (X1) mit 2-Port-Switch
- ③ PROFIBUS DP-Schnittstelle (X3)
- ④ PROFINET-Schnittstelle (X1) mit 3-Port-Switch

Bild 2-1 Schnittstellen der CPU 1516-3 PN/DP und des CPU 1512SP-1 PN

Schnittstellen von Kommunikationsmodulen

Schnittstellen von Kommunikationsmodulen (CM) erweitern die Schnittstellen von CPUs (z. B. erweitert das Kommunikationsmodul CM 1542-5 das Automatisierungssystem S7-1500 um eine PROFIBUS-Schnittstelle).

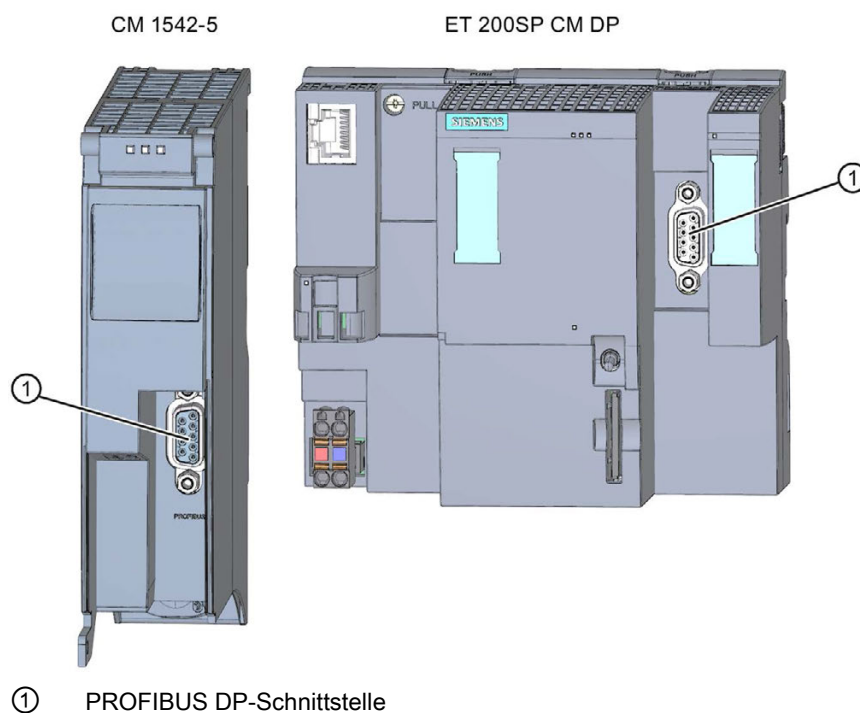
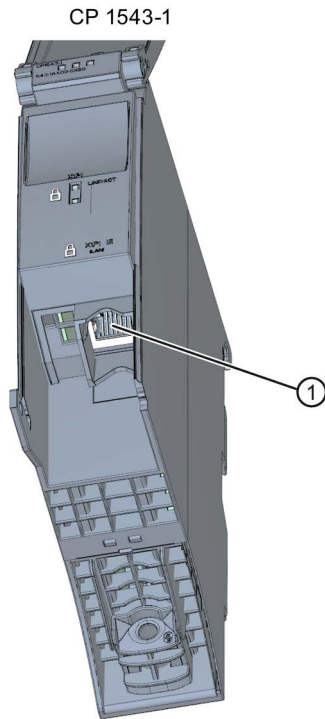


Bild 2-2 PROFIBUS DP-Schnittstelle des CM 1542-5 und des CM DP

Schnittstellen von Kommunikationsprozessoren

Schnittstellen von Kommunikationsprozessoren (CP) bieten Ihnen zusätzliche Funktionalität als die integrierten Schnittstellen der CPUs. CPs decken spezielle Anwendungsfälle ab, z. B. bietet der CP 1543-1 über seine Industrial Ethernet-Schnittstelle Security-Funktionen zur Absicherung von Industrial Ethernet-Netzwerken.



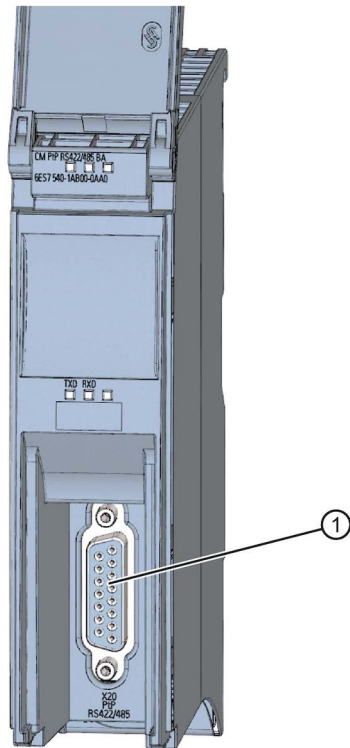
① Industrial Ethernet-Schnittstelle

Bild 2-3 Industrial Ethernet-Schnittstelle des CP 1543-1

Schnittstellen von Kommunikationsmodulen für Punkt-zu-Punkt-Kopplung

Die Kommunikationsmodule für Punkt-zu-Punkt-Kopplung bieten Ihnen Kommunikation über ihre RS232-, RS422- und RS485-Schnittstelle, z. B. Freeport- oder Modbus-Kommunikation.

CM PtP RS422/485 BA

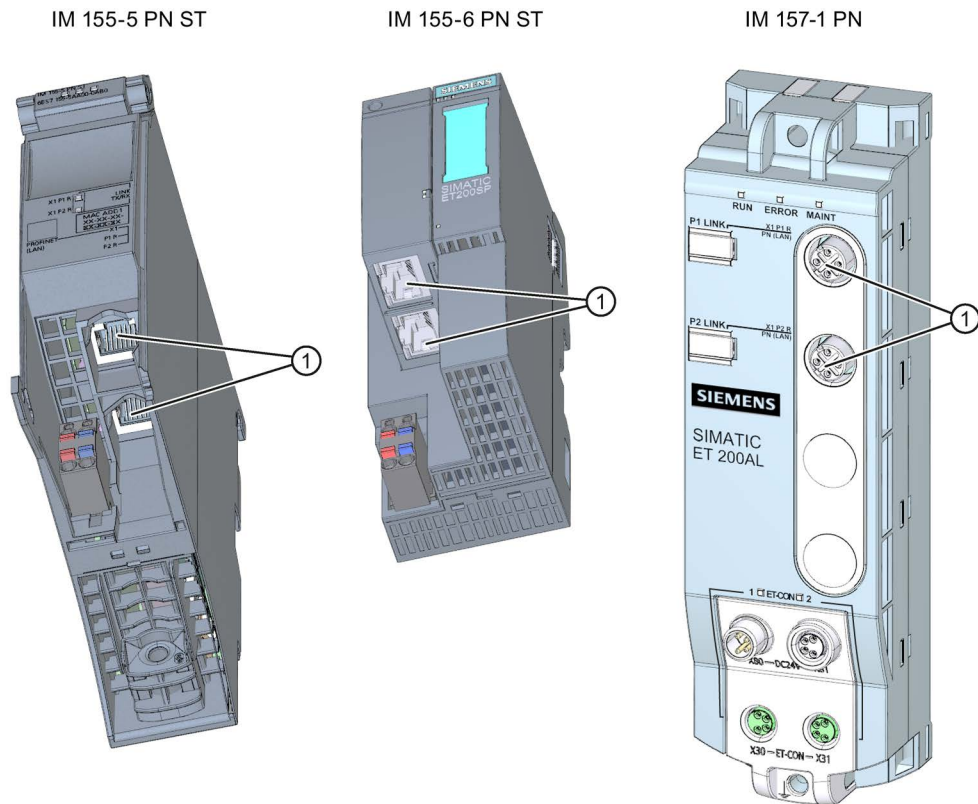


① Schnittstelle für Punkt-zu-Punkt-Kopplung

Bild 2-4 Beispiel für Schnittstelle für Punkt-zu-Punkt-Kopplung am CM PtP RS422/485 BA

Schnittstellen von Interfacemodulen

PROFINET- und PROFIBUS DP-Schnittstellen von Interfacemodulen (IM) in ET 200MP, ET 200SP und ET 200AL dienen der Anbindung der Dezentralen Peripherie ET 200MP, ET 200SP und ET 200AL an PROFINET bzw. PROFIBUS des überlagerten IO-Controllers bzw. DP-Masters.



① PROFINET-Schnittstelle mit 2-Port-Switch

Bild 2-5 PROFINET-Schnittstellen IM 155-5 PN ST (ET 200MP), IM 155-6 PN ST (ET 200SP), und IM 157-1 PN (ET 200AL)

Kommunikationsdienste

Die nachfolgend beschriebenen Kommunikationsdienste nutzen die Schnittstellen und Kommunikationsmechanismen, die Ihnen das System über CPUs, Kommunikationsmodule und -prozessoren bietet.

Kommunikationsdienste

3.1 Kommunikationsmöglichkeiten im Überblick

Übersicht über die Kommunikationsmöglichkeiten

Für Ihre Automatisierungsaufgabe stehen Ihnen folgende Kommunikationsmöglichkeiten zur Verfügung.

Tabelle 3- 1 Möglichkeiten der Kommunikation

Möglichkeiten der Kommunikation	Funktionalität	Über Schnittstelle:		
		PN/IE ¹	DP	serielle
PG-Kommunikation	Zur Inbetriebnahme, Test, Diagnose	X	X	-
HMI-Kommunikation	Zum Bedienen und Beobachten	X	X	-
Offene Kommunikation über TCP/IP	Datenaustausch über PROFINET/Industrial Ethernet mit TCP/IP Anweisungen: <ul style="list-style-type: none"> • TSEND_C/TRCV_C • TSEND/TRCV • TCON • T_DISCON 	X	-	-
Offene Kommunikation über ISO-on-TCP	Datenaustausch über PROFINET/Industrial Ethernet mit ISO-on-TCP Anweisungen: <ul style="list-style-type: none"> • TSEND_C/TRCV_C • TSEND/TRCV • TCON • T_DISCON 	X	-	-
Offene Kommunikation über UDP	Datenaustausch über PROFINET/Industrial Ethernet mit UDP Anweisungen: <ul style="list-style-type: none"> • TSEND_C/TRCV_C • TUSEND/TURCV/TRCV • TCON • T_DISCON 	X	-	-

3.1 Kommunikationsmöglichkeiten im Überblick

Möglichkeiten der Kommunikation	Funktionalität	Über Schnittstelle:		
		PN/IE ¹	DP	serielle
Offene Kommunikation über ISO (nur CPs mit PROFINET/Industrial Ethernet-Schnittstelle)	Datenaustausch über PROFINET/Industrial Ethernet mit ISO-Protokoll Anweisungen: <ul style="list-style-type: none"> • TSEND_C/TRCV_C • TSEND/TRCV • TCON • T_DISCON 	X	-	-
Offene Kommunikation über FDL (nur CM 1542-5 ab Firmwarestand V2.0)	Datenaustausch über PROFIBUS mit Protokoll FDL Anweisungen: <ul style="list-style-type: none"> • TSEND_C/TRCV_C • TSEND/TRCV • TUSEND/TURCV • TCON • T_DISCON 	-	X	-
OPC UA-Server (nur über interne PROFINET-Schnittstellen der CPU)	Datenaustausch mit OPC UA-Clients	X	-	-
Kommunikation über Modbus TCP	Datenaustausch über PROFINET mit Protokoll Modbus TCP Anweisungen: <ul style="list-style-type: none"> • MB_CLIENT • MB_SERVER 	X	-	-
E-Mail	Prozessmeldungen über E-Mail versenden Anweisung: <ul style="list-style-type: none"> • TMAIL_C 	X	-	-
FTP (nur CPs mit PROFINET/Industrial Ethernet-Schnittstelle)	Dateiverwaltung und Dateizugriff über FTP (File Transfer Protocol); CP kann FTP-Client und FTP-Server sein Anweisung: <ul style="list-style-type: none"> • FTP_CMD 	X	-	-
Fetch/Write (nur CPs mit PROFINET/Industrial Ethernet-Schnittstelle)	Serverdienste über TCP/IP, ISO-on-TCP und ISO Über spezielle Anweisungen für Fetch/Write	X	-	-
S7-Kommunikation	Datenaustausch über PROFINET/PROFIBUS mit S7-Protokoll. Anweisungen: <ul style="list-style-type: none"> • PUT/GET • BSEND/BRCV • USEND/URCV 	X	X	-
Serielle Punkt-zu-Punkt-Kopplung	Datenaustausch über Punkt-zu-Punkt mit Freeport-, 3964(R)-, USS- oder Modbus-Protokoll Über spezielle Anweisungen für PtP, USS bzw. Modbus RTU	-	-	X

Möglichkeiten der Kommunikation	Funktionalität	Über Schnittstelle:		
		PN/IE ¹	DP	serielle
Webserver	Datenaustausch über HTTP(S), z. B. zur Diagnose	X	-	-
SNMP (Simple Network Management Protocol)	Zur Überwachung und Fehlererkennung von IP-Netzwerken, ggf. Parametrierung der IP-Netzkomponenten über Standardprotokoll SNMP	X	-	-
Uhrzeitsynchronisation	Über PN/IE-Schnittstelle: CPU ist NTP-Client (Network Time Protocol)	X	-	-
	Über DP-Schnittstelle: CPU/CM/CP ist Uhrzeitmaster oder Uhrzeit-Slave	-	X	-

¹ IE - Industrial Ethernet

Weitere Informationen

- Anwendungsbeispiel: CPU-CPU Kommunikation mit SIMATIC Controllern (Kompendium)
Das Anwendungsbeispiel finden Sie im Internet
(<https://support.industry.siemens.com/cs/ww/de/view/20982954>).
- Wie Sie bei der S7-1500 die Fetch/Write-Kommunikation über einen CP1543-1 konfigurieren, finden Sie in diesem FAQ
(<https://support.industry.siemens.com/cs/ww/de/view/102420020>).
- Weitere Informationen zu den Fetch/Write-Diensten finden Sie in der Online-Hilfe STEP 7.
- Weitere Informationen zur PtP-Kopplung finden Sie im Funktionshandbuch CM PtP - Konfigurationen für Punkt-zu-Punkt-Kopplungen
(<http://support.automation.siemens.com/WW/view/de/59057093>).
- Die Beschreibung der Webserverfunktionalität finden Sie im Funktionshandbuch Webserver (<http://support.automation.siemens.com/WW/view/de/59193560>).
- Informationen zum Standardprotokoll SNMP finden Sie auf den Service & Support Seiten im Internet (<http://support.automation.siemens.com/WW/view/de/15166742>).
- Informationen zur Uhrzeitsynchronisation finden Sie in diesem FAQ
(<https://support.industry.siemens.com/cs/ww/de/view/86535497>).

3.2 Kommunikationsprotokolle und verwendete Portnummern bei Ethernet-Kommunikation

Dieser Abschnitt gibt einen Überblick über die unterstützten Protokolle die verwendeten Portnummern bei Kommunikation über PN/IE-Schnittstellen. Für jedes Protokoll sind die Adressparameter, die betroffene Kommunikationsschicht sowie die Kommunikationsrolle und Kommunikationsrichtung angegeben.

Diese Informationen ermöglichen Ihnen, Security-Maßnahmen zum Schutz des Automatisierungssystems auf die verwendeten Protokolle abzustimmen (z. B. Firewall). Da sich Security-Maßnahmen auf Ethernet- bzw. PROFINET-Netze beschränken, sind in den Tabellen keine PROFIBUS-Protokolle aufgeführt.

Hinweis

Verwendete Portnummern

Die angegebenen Portnummern sind die von der S7-1500 CPU standardmäßig verwendeten Portnummern. Viele Kommunikationsprotokolle bzw. Implementierungen erlauben es Ihnen, andere Portnummern zu verwenden.

Die folgenden Tabellen zeigen die verschiedenen Schichten und Protokolle, die Einsatz finden.

Die folgende Tabelle zeigt die von den S7-1500 CPUs, ET 200SP CPUs und der CPU 1516pro-2 PN unterstützten Protokolle. Die S7-1500 Software Controller unterstützen ebenfalls die in der folgenden Tabelle aufgeführten Protokolle für die Ethernet-Schnittstellen, die dem Software Controller zugewiesen sind.

Tabelle 3-2 Schichten und Protokolle der S7-1500 CPUs und Software Controller (über PROFINET-Schnittstelle der CPU)

Protokoll	Portnummer	(2) Link-Layer-Schicht (4) Transport-schicht	Funktion	Beschreibung
PROFINET-Protokolle				
DCP Discovery and configuration protocol	Nicht relevant	(2) Ethertype 0x8892 (PROFINET)	Erreichbare Teil- nehmer, PROFINET Dis- covery and confi- guration	DCP wird von PROFINET verwendet, um PROFINET-Geräte zu ermitteln und Grundein- stellungen zu ermöglichen.
LLDP Link Layer Discovery protocol	Nicht relevant	(2) Ethertype 0x88CC (LLDP)	PROFINET Link Layer Discovery protocol	LLDP wird von PROFINET verwendet, um Nach- barschaftsbeziehungen zwischen PROFINET- Geräten zu ermitteln und zu verwalten. LLDP verwendet die spezielle Multicast-MAC- Adresse: 01-80-C2-00-00-0E
MRP Media Redun- dancy Protocol	Nicht relevant	(2) Ethertype 0x88E3 (IEC 62493-2- 2010)	PROFINET me- dium redundancy	MRP ermöglicht die Steuerung von redundanten Übertragungswegen durch eine Ringtopologie. MRP verwendet normkonforme Multi- cast-MAC-Adressen.

3.2 Kommunikationsprotokolle und verwendete Portnummern bei Ethernet-Kommunikation

Protokoll	Portnummer	(2) Link-Layer-Schicht (4) Transportschicht	Funktion	Beschreibung
PTCP Precision Transparent Clock Protocol	Nicht relevant	(2) Ethertype 0x8892 (PROFINET)	PROFINET send clock and time synchronisation, based on IEEE 1588	PTC ermöglicht eine Zeitverzögerungsmessung zwischen RJ45 Ports und damit die Sendetakt-Synchronisation und Zeitsynchronisation. PTCP verwendet normkonforme Multicast-MAC-Adressen.
PROFINET IO data	Nicht relevant	(2) Ethertype 0x8892 (PROFINET)	PROFINET Cyclical IO data transfer	Die PROFINET-IO Telegramme werden verwendet, um IO-Daten zyklisch zwischen PROFINET IO-Controller und IO-Devices über Ethernet zu übertragen.
PROFINET Context Manager	34964	(4) UDP	PROFINET connection less RPC	Der PROFINET Context Manager stellt einen Endpoint-Mapper zur Verfügung, um eine Applikationsbeziehung (PROFINET AR) herzustellen.
Verbindungsorientierte Kommunikationsprotokolle				
SMTP Simple mail transfer protocol	25	(4) TCP	Simple mail transfer protocol	SMTP wird zum Senden von E-Mails verwendet.
SMTPS (SMTP over TLS)	465	(4) TCP	Secure SMTP	SMTPS wird zum Senden von E-Mails über gesicherte Verbindungen verwendet.
SMTP mit STARTTLS	25 587	(4) TCP	Simple mail transfer protocol mit dem SMTP-Befehl "STARTTLS"	SMTP mit STARTTLS wird zum Senden von E-Mails über gesicherte Verbindungen verwendet.
HTTP Hypertext transfer protocol	80	(4) TCP	Hypertext transfer protocol	HTTP wird verwendet zur Kommunikation mit dem CPU-internen Webserver.
ISO-on-TCP (gemäß RFC 1006)	102	(4) TCP	ISO-on-TCP protocol	ISO-on-TCP (gemäß RFC 1006) dient zum nachrichtenorientierten Datenaustausch an entfernte CPU oder Software Controller S7-Kommunikation mit ES, HMI, OPC Server, usw.
NTP Network time protocol	123	(4) UDP	Network time protocol	NTP wird zur Synchronisation der Systemzeit der CPU mit der Uhrzeit eines NTP-Servers genutzt.
SNMP Simple network management protocol	161 162 (trap)	(4) UDP	Simple network management protocol	SNMP ermöglicht das Auslesen und Setzen von Netzwerk-Management-Daten (SNMP managed Objects) durch SNMP-Manager.
HTTPS Secure Hypertext transfer protocol	443	(4) TCP	Secure Hypertext transfer protocol	HTTPS wird verwendet, um mit dem CPU-internen Webserver über Secure Socket Layer (SSL) zu kommunizieren.

3.2 Kommunikationsprotokolle und verwendete Portnummern bei Ethernet-Kommunikation

Protokoll	Portnummer	(2) Link-Layer-Schicht (4) Transport-schicht	Funktion	Beschreibung
Modbus TCP Modbus Transmission Control Protocol	502	(4) TCP	Modbus/TCP protocol	Modbus/TCP wird durch MB_CLIENT/MB_SERVER Anweisungen im Anwenderprogramm genutzt.
OPC UA Open Platform Communications Unified Architecture	4840	(4) TCP	Basiert auf TCP/IP-Protokoll	Kommunikationsstandard mit Reichweite von der Enterprise-Ebene bis auf die Feldebene.
OUC ¹ Open User Communication und Secure OUC	1 ... 1999 bedingt nutzbar ² 2000 ... 5000 empfohlen 5001 ... 49151 bedingt nutzbar ²	(4) TCP (4) UDP	Open User Communication (TCP/UDP) Secure Open User Communication (TLS)	OUC-Anweisungen ermöglichen Verbindungsaufbau, Verbindungsabbau und Datentransfer basierend auf dem Socket Layer.
IGMPv2 Internet Group Management Protocol	Nicht relevant	(3) Vermittlungsschicht	Internet Group Management Protocol	Netzwerkprotokoll zur Organisation von Multicast-Kreisen.
Reserved	49152 ... 65535	(4) TCP (4) UDP	-	Dynamischer Port-Bereich, der für den aktiven Verbindungsendpunkt verwendet wird, wenn die Applikation die lokale Portnummer nicht bestimmt.

¹ Hinweis: Die offene Kommunikation liefert einen direkten Zugang zum UDP/TCP für den Benutzer. Der Anwender ist verantwortlich die Port-Einschränkungen/Definitionen der IANA (Internet Assigned Numbers Authority) zu berücksichtigen.

² Verwenden Sie keine Ports für OUC, die bereits durch andere Protokolle belegt sind.

3.2 Kommunikationsprotokolle und verwendete Portnummern bei Ethernet-Kommunikation

Die folgende Tabelle zeigt die Protokolle, die vom S7-1500 Software Controller über die Windows zugewiesenen Ethernet-Schnittstellen unterstützt werden.

Tabelle 3- 3 Schichten und Protokolle der S7-1500 Software Controller (über Ethernet-Schnittstelle der Windows-Seite)

Protokoll	Portnummer	(2) Link-Layer-Schicht (4) Transportschicht	Funktion	Beschreibung
PROFINET-Protokolle				
DCP Discovery and configuration protocol	Nicht relevant	(2) Ether-type 0x8892 (PROFINET)	Erreichbare Teilnehmer, PROFINET Discovery and configuration	DCP wird von PROFINET verwendet, um PROFINET-Geräte zu ermitteln und Grundeinstellungen zu ermöglichen.
Verbindungsorientierte Kommunikationsprotokolle				
SMTP Simple mail transfer protocol	25	(4) TCP	Simple mail transfer protocol	SMTP wird zum Senden von E-Mails verwendet.
HTTP Hypertext transfer protocol	Einstellbar ¹	(4) TCP	Hypertext transfer protocol	HTTP wird verwendet zur Kommunikation mit CPU-internem Webserver. Die Portnummer können Sie zur Vermeidung von Konflikten mit anderen Webservern unter Windows anpassen. Wenn Sie Webserverzugang nutzen wollen, müssen Sie den Port in der Windows Firewall freischalten.
ISO-on-TCP (gemäß RFC 1006)	102	(4) TCP	ISO-on-TCP protocol	ISO-on-TCP (gemäß RFC 1006) zur S7-Kommunikation mit PG/PC oder HMI.
OUC ² Open User Communication und Secure OUC	1 ... 1999 bedingt nutzbar ^{3,4}	(4) TCP (4) UDP	Open User Communication (TCP/UDP) Secure Open User Communication (TLS)	OUC-Anweisungen ermöglichen Verbindungsaufbau, Verbindungsabbau und Datentransfer basierend auf dem Socket Layer. Wenn Sie OUC nutzen wollen, müssen Sie die Ports in der Windows Firewall freischalten.
	2000 ... 5000 empfohlen ⁴			
	5001 ... 49151 bedingt nutzbar ^{3,4}			

Protokoll	Portnummer	(2) Link-Layer-Schicht (4) Transportschicht	Funktion	Beschreibung
IGMPv2 Internet Group Management Protocol	Nicht relevant	(3) Vermittlungsschicht	Internet Group Management Protocol	Netzwerkprotokoll zur Organisation von Multicast-Kreisen.
Reserved	49152 ... 65535	(4) TCP (4) UDP	-	Dynamischer Port-Bereich, der für den aktiven Verbindungsendpunkt verwendet wird, wenn die Applikation die lokale Portnummer nicht bestimmt. Wenn Sie diese Kommunikation nutzen wollen, müssen Sie die Ports in der Windows Firewall freischalten.

- ¹ Voreinstellung bei Windows zugewiesenen Schnittstellen: 81
- ² Hinweis: Die offene Benutzerkommunikation liefert einen direkten Zugang zum UDP/TCP für den Benutzer. Der Anwender ist verantwortlich die Port-Einschränkungen/Definitionen der IANA (Internet Assigned Numbers Authority) zu berücksichtigen.
- ³ Verwenden Sie keine Ports für OUC, die bereits durch andere Protokolle belegt sind.
- ⁴ Verwenden Sie keine Ports für OUC, die bereits durch andere Windows-Anwendungen belegt sind.

3.2 Kommunikationsprotokolle und verwendete Portnummern bei Ethernet-Kommunikation

Die folgende Tabelle zeigt die Protokolle, die zusätzlich zu den in den Tabellen oben genannten Protokollen von den S7-1500 Kommunikationsmodulen (z. B. CP 1543-1) unterstützt werden.

Tabelle 3- 4 Schichten und Protokolle von S7-1500 Kommunikationsmodulen

Protokoll	Portnummer	(2) Link-Layer-Schicht (4) Transport-schicht	Funktion	Beschreibung
PROFINET/Industrial Ethernet-Protokolle				
Verbindungsorientierte Kommunikationsprotokolle				
FTP File transfer protocol	20 (data) 21 (control)	(4) TCP	File transfer protocol	FTP wird zur Übertragung von Dateien (nur in Verbindung mit CP 1543-1) verwendet.
secureFTP File transfer protocol	20 (data) 21 (control)	(4) TCP	File transfer protocol	SecureFTP wird zur Übertragung von Dateien über eine TLS-Verbindung (nur in Verbindung mit CP 1543-1) verwendet.
DHCP Dynamic Host Configuration Protocol	68	(4) UDP	Dynamic Host Configuration protocol	DHCP wird verwendet, um im Hochlauf der IE-Schnittstelle die IP Address Suite von einem DHCP Server zu beziehen.
Gesichertes NTPv3 Network time protocol	123	(4) UDP	Network time protocol	Gesichertes NTP wird verwendet, um die CP 1543-1 interne Systemuhr auf einen NTP-Server abzustimmen.
SNMP Simple network management protocol	161 162 (trap)	(4) UDP	Simple network management protocol	SNMPv3 erlaubt dem CP 1543-1 Netzwerk-Management-Daten (MIBs) von SNMPv3-Agent mit Authentifikation zu lesen.

Besonderheit S7-1500 MFP:

Port 111: Die S7-1500 MFP nutzt den Port 111 zum NFS-Sevice für interne Kommunikation.

3.3 Verbindungsressourcen im Überblick

Verbindungsressourcen

Einige Kommunikationsdienste benötigen Verbindungen. Verbindungen belegen Ressourcen in den beteiligten CPUs, CPs und CMs (z. B. Speicherbereiche im Betriebssystem der CPU). In den meisten Fällen wird für eine Verbindung eine Ressource pro CPU/CP/CM belegt. Bei HMI-Kommunikation werden pro HMI-Verbindung bis zu 3 Verbindungsressourcen benötigt.

Die zur Verfügung stehenden Verbindungsressourcen sind abhängig von der eingesetzten CPU, den CPs und CMs und dürfen eine definierte Obergrenze für das Automatisierungssystem nicht überschreiten.

Verfügbare Verbindungsressourcen in einer Station

Die maximale Anzahl der Ressourcen einer Station wird durch die CPU bestimmt.

Jede CPU bringt reservierte Verbindungsressourcen für PG-, HMI- und Webserver-Kommunikation mit. Daneben gibt es verfügbare Ressourcen, die für SNMP, E-Mail-Verbindungen, HMI- und S7-Kommunikation sowie für offene Kommunikation genutzt werden können.

Wann werden Verbindungsressourcen belegt?

Der Zeitpunkt für die Belegung der Verbindungsressourcen hängt davon ab, wie die Verbindung eingerichtet wird, automatisch, programmiert oder projiziert (siehe Kapitel Einrichten einer Verbindung (Seite 31)).

Weitere Informationen

Nähere Informationen zur Belegung von Verbindungsressourcen und zur Anzeige von Verbindungsressourcen in STEP 7 finden Sie im Kapitel Verbindungsressourcen (Seite 242).

3.4 Einrichten einer Verbindung

Automatische Verbindung

STEP 7 richtet eine Verbindung automatisch ein (z. B. PG- oder HMI-Verbindung), sofern Sie die PG/PC-Schnittstelle physikalisch mit einer Schnittstelle der CPU verbunden haben und in STEP 7, im Dialog "Online verbinden" die Schnittstellen-Zuordnung vorgenommen haben.

Programmiertes Einrichten der Verbindung

Die programmierte Verbindung richten Sie im Programmierer von STEP 7 im Kontext einer CPU durch Parametrierung von Anweisungen für Kommunikation, z. B. TSEND_C ein.

Bei der Festlegung der Verbindungsparameter (im Inspektorfenster, in den Eigenschaften der Anweisung) werden Sie durch die komfortable Bedienoberfläche unterstützt.

The image shows the TSEND_C function block in a Ladder Logic diagram and its corresponding configuration window.

Function Block Diagram:

- Block:** TSEND_C (FB1030)
- Inputs:**
 - EN: Connected to a power rail.
 - REQ: Connected to a variable labeled **%DB2** with the value "TSEND_C_DB".
 - CONT: Connected to a variable labeled **TRUE**.
 - CONNECT: Connected to a variable labeled **%DB3** with the value "PLC_1_Send_DB".
 - DATA: Connected to a variable labeled "Data_block_1".
 - Con_Data_1: Connected to a variable labeled "Con_Data_1".
- Outputs:**
 - ENO: Connected to a power rail.
 - DONE: Connected to a variable labeled "Data_block_1".
 - BUSY: Connected to a variable labeled "Data_block_1".
 - ERROR: Connected to a variable labeled "Data_block_1".
 - STATUS: Connected to a variable labeled "Data_block_1".

Configuration Window (TSEND_C [FB1030]):

- General Tab (Allgemein):**
 - Verbindung...** (checked)
 - Bausteinpara...** (checked)
 - Übersicht ü...**
- Verbindungsparameter:**
 - Allgemein:**
 - Lokal:**
 - Endpunkt: PLC_1 [CPU 1516-3 PN/DP]
 - Schnittstelle: PLC_1, PROFINET-Schnittstelle_1[X1]
 - Subnetz: PN/IE_1
 - Adresse: 192.168.0.2
 - Partner:**
 - Endpunkt: PLC_2 [CPU 1516-3 PN/DP]
 - Schnittstelle: PLC_2, PROFINET-Schnittstelle_1[X1]
 - Subnetz: PN/IE_1
 - Adresse: 192.168.0.1
 - Verbindungstyp:** TCP
 - Konfigurationsart:** Programmbausteine verwenden
 - Verbindungs-ID (dez):** 1
 - Verbindungsdaten:**
 - Lokal: PLC_1_Send_DB
 - Partner: PLC_2_Receive_DB
 - Aktiver Verbindungsaufbau:**
 - Lokal: ☒ Aktivierter Verbindungsaufbau
 - Partner: ☐ Aktivierter Verbindungsaufbau

Bild 3-1 Programmiertes Einrichten

Projektiertes Einrichten der Verbindung

Die projektierte Verbindung richten Sie in der Netzansicht des Hardware- und Netzwerkeditors von STEP 7 im Kontext einer CPU oder eines Software-Controllers ein.

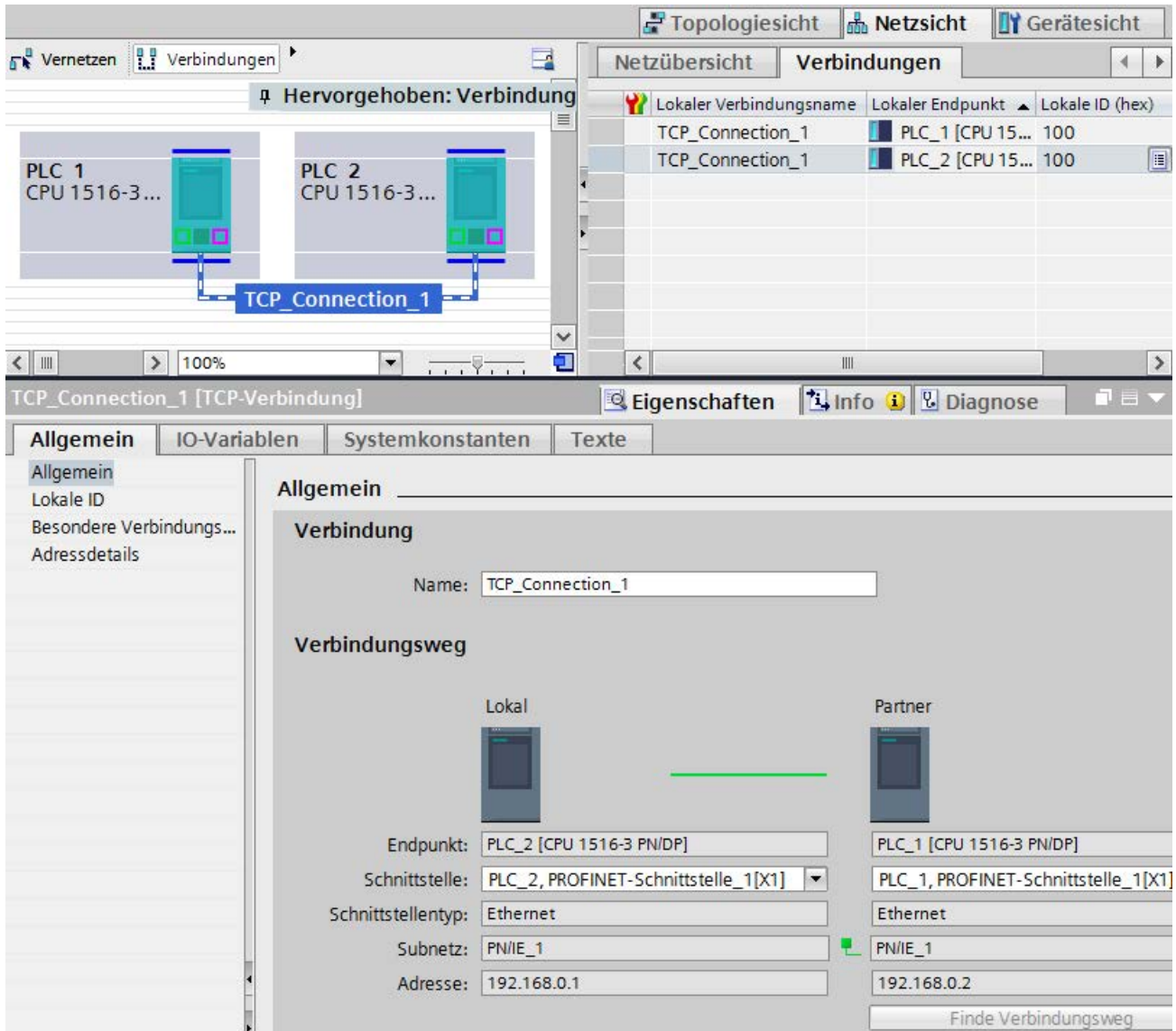


Bild 3-2 Projektiertes Einrichten

Auswirkungen auf die Verbindungsressourcen der CPU

Sie können sich oftmals alternativ für eine projektierte oder programmierte Verbindung entscheiden. Das programmierte Einrichten ermöglicht die Freigabe von Verbindungsressourcen nach der Datenübertragung. Programmierte Verbindungen sind wie geroutete Verbindungen nicht garantiert, d. h. sie werden nur aufgebaut, wenn Ressourcen frei sind. Beim projektierten Einrichten steht die Ressource nach dem Download der Konfiguration bis zur erneuten Änderung der Konfiguration zur Verfügung. Für den Verbindungsaufbau über projektierte Verbindungen sind daher entsprechende Ressourcen reserviert. Die Tabelle "Verbindungsressourcen" im Inspektorfenster der CPU zeigt eine Übersicht der bereits belegten und noch verfügbaren Verbindungsressourcen an.

Wie richte ich welche Verbindung ein?

Tabelle 3- 5 Einrichten der Verbindung

Verbindung	Automatisch	Programmiertes Einrichten	Projektiertes Einrichten
PG-Verbindung	X	-	-
HMI-Verbindung	X	-	X
Offene Kommunikation über TCP/IP-Verbindung	-	X	X
Offene Kommunikation über ISO-on-TCP-Verbindung	-	X	X
Offene Kommunikation über UDP-Verbindung	-	X	X
Offene Kommunikation über ISO-Verbindung	-	X	X
Offene Kommunikation über FDL-Verbindung	-	X	X
Kommunikation über Modbus TCP-Verbindung	-	X	-
E-Mail-Verbindung	-	X	-
FTP-Verbindung	-	X	-
S7-Verbindung*	-	-	X

* Beachten Sie, daß Sie bei einer S7-1500 CPU mit Firmwarestand kleiner V2.0 die Nutzung von PUT/GET-Kommunikation in den Eigenschaften der CPU freigeben müssen. Weitere Informationen dazu finden Sie in der Online-Hilfe von STEP 7.

Weitere Informationen

Weitere Informationen zur Belegung von Verbindungsressourcen und zur Anzeige von Verbindungsressourcen in STEP 7 finden Sie im Kapitel Verbindungsressourcen (Seite 242).

3.5 Datenkonsistenz

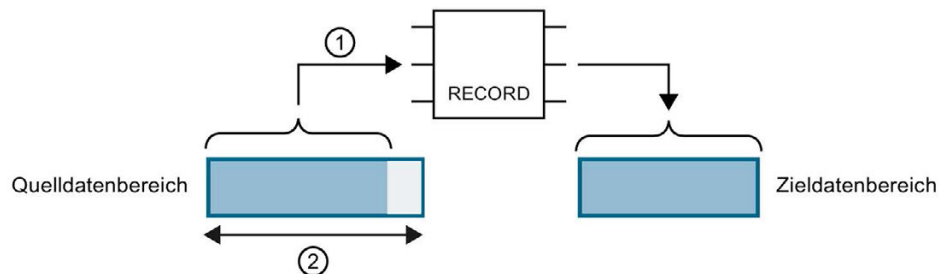
Definition

Für die Übertragung von Daten ist die Datenkonsistenz eine wichtige Eigenschaft, die Sie bei der Projektierung einer Kommunikationsaufgabe berücksichtigen müssen. Geschieht das nicht, kann es zu Fehlfunktionen kommen.

Ein Datenbereich, der nicht durch konkurrierende Prozesse verändert werden kann, wird als konsistenter Datenbereich bezeichnet. Das heißt, ein in sich zusammengehöriger Datenbereich, der größer ist als die Maximalgröße des konsistenten Datenbereichs, kann zu einem Zeitpunkt teilweise aus neuen und aus alten Daten bestehen.

Eine Inkonsistenz kann entstehen, wenn eine Anweisung für Kommunikation z. B. durch einen Prozessalarm-OB mit höherer Priorität unterbrochen wird. Dadurch wird auch die Übertragung des Datenbereichs unterbrochen. Verändert das Anwenderprogramm in diesem OB jetzt die Daten, die noch nicht von der Kommunikationsanweisung verarbeitet wurden, stammen die übertragenen Daten aus unterschiedlichen Zeitpunkten.

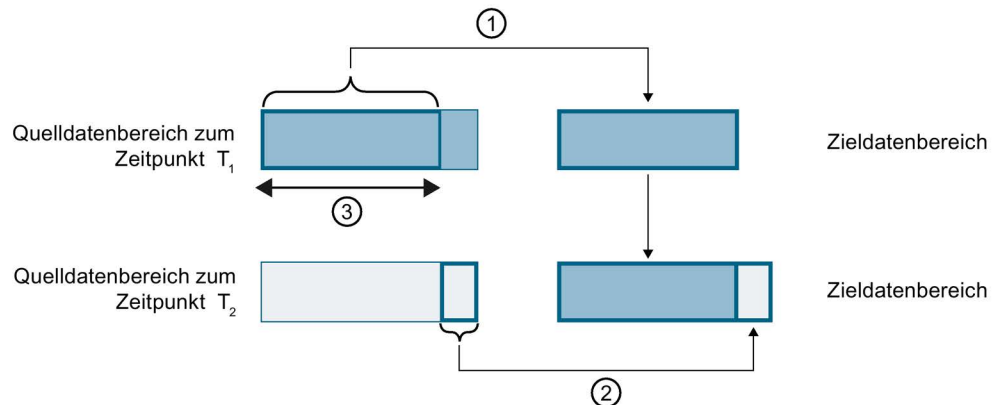
Das folgende Bild zeigt einen Datenbereich, der kleiner ist als die Maximalgröße des konsistenten Datenbereichs. In diesem Fall wird bei der Übertragung des Datenbereichs sichergestellt, dass während des Datenzugriffs keine Unterbrechung durch das Anwenderprogramm erfolgt und damit die Daten nicht geändert werden.



- ① Der Quelldatenbereich ist kleiner als die Maximalgröße des konsistenten Datenbereichs (②). Die Anweisung überträgt die Daten zusammenhängend in den Zielfatenbereich.
- ② Maximalgröße konsistenter Datenbereich

Bild 3-3 Konsistente Übertragung von Daten

Das folgende Bild zeigt einen Datenbereich, der größer ist als die Maximalgröße des konsistenten Datenbereichs. In diesem Fall können die Daten während einer Unterbrechung der Datenübertragung verändert werden. Eine Unterbrechung entsteht z.B. auch, wenn der Datenbereich in mehreren Teilen übertragen werden muss. Werden die Daten während der Unterbrechung verändert, dann stammen die übertragenen Daten aus unterschiedlichen Zeitpunkten.

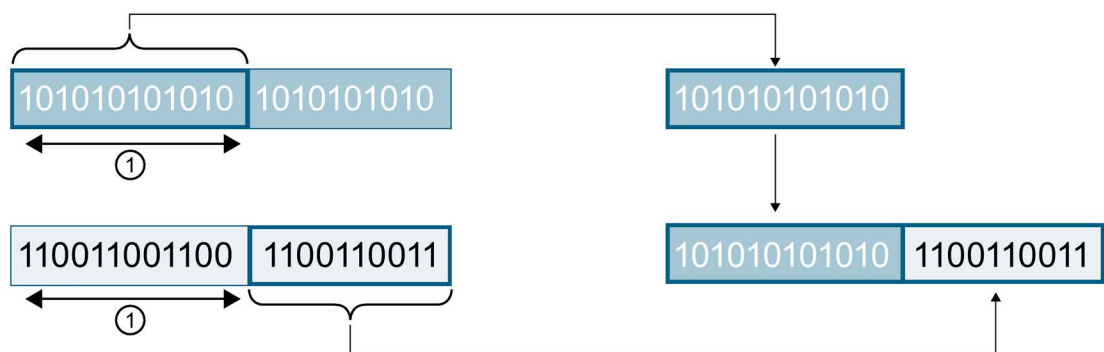


- ① Der Quelldatenbereich ist größer als die Maximalgröße des konsistenten Datenbereichs (③). Zum Zeitpunkt T₁ überträgt die Anweisung nur so viel Daten vom Quelldatenbereich in den Zieldatenbereich, wie in den konsistenten Datenbereich reinpassen.
- ② Zum Zeitpunkt T₂ überträgt die Anweisung den Rest des Quelldatenbereichs in den Zieldatenbereich. Nach der Übertragung liegen im Zieldatenbereich Daten aus unterschiedlichen Zeitpunkten. Wenn sich die Daten im Quelldatenbereich in der Zwischenzeit geändert haben, kann eine Inkonsistenz entstehen.
- ③ Maximalgröße konsistenter Datenbereich

Bild 3-4 Übertragung von Daten größer dem maximalen Konsistenzbereich

Beispiel für eine Inkonsistenz

Das folgende Bild zeigt ein Beispiel für Veränderung von Daten während der Übertragung. Im Zieldatenbereich liegen Daten aus unterschiedlichen Zeitpunkten.



- ① Maximalgröße konsistenter Datenbereich

Bild 3-5 Beispiel: Veränderung von Daten während der Übertragung

Systemspezifische maximale Datenkonsistenz für S7-1500:

Eine Inkonsistenz tritt nicht auf, wenn die systemspezifische Maximalgröße der konsistenten Daten eingehalten wird. Bei S7-1500 werden die Kommunikationsdaten in Blöcken bis maximal 512 Byte während des Programmzyklus konsistent in/aus dem Anwenderspeicher kopiert. Für alle größeren Datenbereiche wird keine Datenkonsistenz garantiert. Ist eine definierte Datenkonsistenz gefordert, so dürfen die Kommunikationsdaten im Anwenderprogramm der CPU nicht größer als 512 Byte sein. Auf diese Datenbereiche können Sie dann z. B. von einem HMI-Gerät mit Lesen/Schreiben von Variablen konsistent zugreifen.

Wenn mehr Daten als die systemspezifische Maximalgröße konsistent übertragen werden sollen, dann müssen Sie selbst durch entsprechende Maßnahmen im Anwenderprogramm die Datenkonsistenz sicherstellen.

Datenkonsistenz sicherstellen**Einsatz von Anweisungen für Zugriff auf gemeinsame Daten:**

Existieren im Anwenderprogramm Kommunikationsanweisungen, welche auf gemeinsame Daten zugreifen, z. B. TSEND/TRCV, können Sie den Zugriff auf diesen Datenbereich z. B. über den Parameter "DONE" selbst koordinieren. Die Datenkonsistenz der Datenbereiche, die mit einer Anweisung für Kommunikation übertragen werden, kann deshalb im Anwenderprogramm sichergestellt werden.

Hinweis**Maßnahmen im Anwenderprogramm**

Um Datenkonsistenz zu erreichen, können Sie die zu übertragenden Daten auf einen separaten Datenbereich (z. B. globaler Datenbaustein) umkopieren. Während das Anwenderprogramm weiterhin mit den Originaldaten arbeitet, können Sie die im separaten Datenbereich gespeicherten Daten konsistent mit der Kommunikationsanweisung übertragen.

Verwenden Sie für das Umkopieren nicht unterbrechbare Anweisungen, wie UMOVE_BLK oder UFILL_BLK. Diese Anweisungen gewährleisten eine Datenkonsistenz bis 16 KByte.

Einsatz von Anweisungen PUT/GET bzw. Schreiben/Lesen über HMI-Kommunikation:

Bei S7-Kommunikation mit den Anweisungen PUT/GET bzw. Schreiben/Lesen über HMI-Kommunikation müssen Sie bereits bei der Programmierung bzw. Projektierung die Größe der konsistenten Datenbereiche berücksichtigen. Im Anwenderprogramm einer S7-1500 als Server ist keine Anweisung vorhanden, die die Datenübertragung im Anwenderprogramm koordinieren kann. Die über PUT/GET-Anweisungen ausgetauschten Daten aktualisiert die S7-1500 während der Laufzeit des Anwenderprogramms. Es gibt keinen Zeitpunkt innerhalb der Bearbeitung des zyklischen Anwenderprogramms, an dem die Daten konsistent ausgetauscht werden. Die Länge des zu übertragenden Datenbereichs sollte kleiner sein als 512 Bytes.

Weitere Informationen

- Die max. Anzahl konsistenter Daten finden Sie auch in den Gerätehandbüchern der Kommunikationsmodule in den Technischen Daten.
- Weitere Informationen zur Datenkonsistenz finden Sie in der Beschreibung der Anweisungen in der Online-Hilfe STEP 7.

3.6 Secure Communication

3.6.1 Grundlagen zu Secure Communication

Für STEP 7 (TIA Portal) ab V14 und für S7-1500 CPUs ab Firmware V2.0 sind die Möglichkeiten zur sicheren Kommunikation, im Folgenden als "Secure Communication" bezeichnet, erheblich erweitert worden.

Einleitung

Das Attribut "secure" wird für die Kennzeichnung von Kommunikationsmechanismen verwendet, die auf einer Public Key Infrastructure (PKI) aufbauen (z. B. RFC 5280 für Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile). Mit Public Key Infrastructure (PKI) ist ein System gemeint, das digitale Zertifikate ausstellen, verteilen und prüfen kann. Die ausgestellten digitalen Zertifikate werden innerhalb der PKI zur Absicherung von rechnergestützter Kommunikation verwendet. Wenn eine PKI ein asymmetrisches Schlüsselverfahren nutzt, dann können die Nachrichten in einem Netzwerk digital signiert und verschlüsselt werden.

Komponenten, die Sie in STEP 7 (TIA Portal) für secure Communication projiziert haben, verwenden ein asymmetrisches Schlüsselverfahren mit öffentlichem Schlüssel (Public Key) und privatem Schlüssel (Private Key). Als Verschlüsselungsprotokoll wird TLS (Transport Layer Security) eingesetzt. TLS ist Nachfolger für das Protokoll SSL (Secure Sockets Layer).

Ziele für Secure Communication

Secure Communication wird angewendet, um folgende Ziele zu erreichen:

- Vertraulichkeit
d. h. die Daten sind für nicht autorisierten Lauscher geheim bzw. nicht lesbar.
- Integrität
d. h. die Nachricht, die beim Empfänger eintrifft ist dieselbe, unveränderte Nachricht, die der Sender geschickt hat. Die Nachricht wurde auf dem Transportweg nicht verändert.
- Endpunkt Authentifizierung
d. h. der Kommunikationspartner als Endpunkt ist genau derjenige, der er vorgibt zu sein und der erreicht werden soll. Die Identität des Partners ist geprüft.

Waren diese Ziele in der Vergangenheit hauptsächlich für die IT-Welt und für die vernetzten Computer von Belang, so sind heutzutage auch im industriellen Umfeld Maschinen und Steuerungen mit schützenswerten Daten durch ihre Vernetzung genauso gefährdet und stellen hohe Anforderungen an einen sicheren Datenaustausch.

Bisher und weiterhin gängig ist der Schutz der Automatisierungszelle mit Hilfe des Zellschutzkonzepts per Firewall, oder per Verbindung über VPN, z. B. mit dem Security Modul.

Zunehmend besteht aber der Kommunikationsbedarf, auch Daten an externe Rechner in verschlüsselter Form über Intranet oder öffentliche Netze zu übertragen

Gemeinsame Prinzipien von Secure Communication

Unabhängig vom Kontext basiert Secure Communication auf dem Konzept der Public Key Infrastructure (PKI) und beinhaltet folgende Komponenten:

- Ein asymmetrisches Verschlüsselungsverfahren. Dieses Verfahren ermöglicht Folgendes:
 - Verschlüsselung oder Entschlüsselung der Nachrichten mit Hilfe von öffentlichen oder privaten Schlüsseln.
 - Überprüfung von Signaturen an Nachrichten und Zertifikaten.

Die Nachrichten/Zertifikate werden vom Sender/Zertifikatsinhaber mit ihrem privaten Schlüssel signiert. Der Empfänger/Prüfer überprüft die Signatur mit dem öffentlichen Schlüssel des Senders/Zertifikatsinhabers.

- Transport und Speicherung der öffentlichen Schlüssel mit Hilfe von X.509-Zertifikaten:
 - X.509-Zertifikate sind digital signierte Daten, mit der die Echtheit von öffentlichen Schlüsseln in Bezug auf die gebundene Identität überprüft werden kann.
 - X.509-Zertifikate können Informationen enthalten, die die Benutzung der öffentlichen Schlüssel genauer charakterisieren bzw. einschränken. Zum Beispiel ab wann ein öffentlicher Schlüssel in einem Zertifikat gültig ist und ab wann er ungültig wird.
 - X.509-Zertifikate enthalten abgesichert die Informationen über den Herausgeber des Zertifikats.

Die folgenden Ausführungen geben einen Überblick über diese Grundkonzepte, die z.B. für den Umgang mit Zertifikaten in STEP 7 (TIA Portal) oder für die Programmierung der Kommunikationsanweisungen für secure Open User Communication (sOUC) erforderlich sind.

Secure Communication bei STEP 7

STEP 7 ab V14 stellt die jeweils notwendige PKI zur Verfügung, die für die Projektierung und den Betrieb einer Secure Communication erforderlich ist.

Beispiele:

- Das Hypertext Transfer Protokoll (HTTP) wird mit Hilfe des Protokolls TLS (Transport Layer Security) zum Hypertext Transfer Protokoll Secure (HTTPS). Weil HTTPS eine Kombination aus HTTP und TLS ist, wird es im entsprechenden RFC "HTTP over TLS" genannt. Die Verwendung von HTTPS erkennt man im Browser daran, dass in der Aufrufzeile des Browsers das URL-Schema "https://" statt "http://" verwendet wird. Die meisten Browser heben eine solchermaßen abgesicherte Verbindung zusätzlich optisch hervor.
- Open User Communication wird zur secure Open User Communication. Das zugrunde liegende Protokoll ist ebenfalls TLS.
- E-Mail Provider bieten ebenfalls Zugang über das Protokoll "Secure SMTP over TLS" an, um die Sicherheit des E-Mail-Verkehrs zu erhöhen.

Das folgende Bild zeigt das Protokoll TLS im Kontext der Kommunikationsschichten.

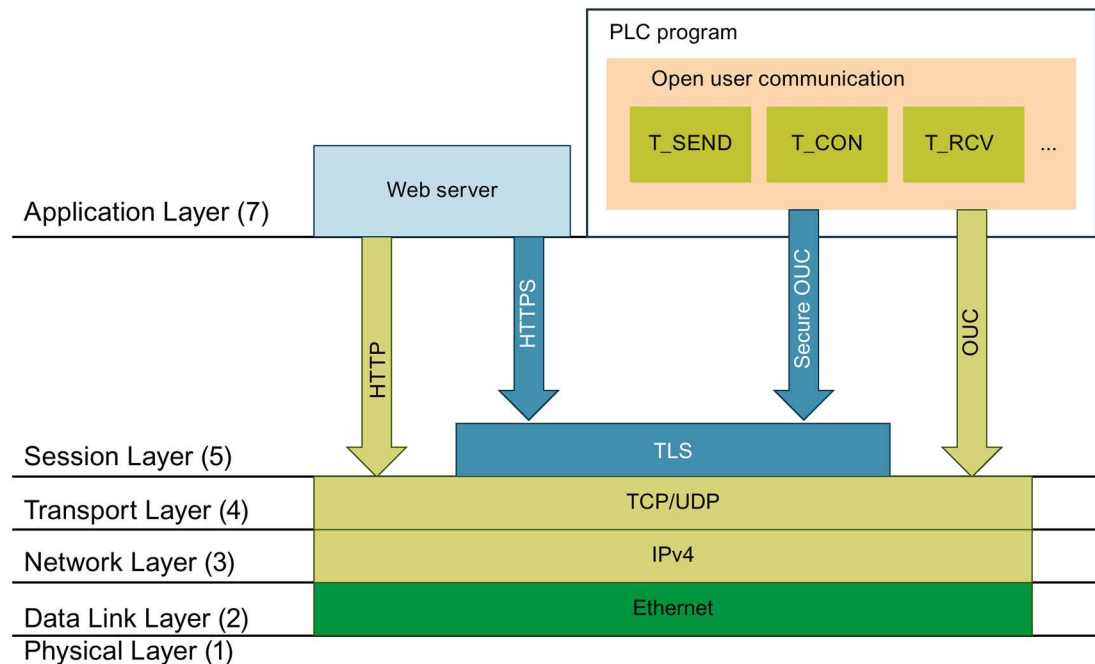


Bild 3-6 Protokoll TLS im Kontext der Kommunikationsschichten

Secure Communication bei OPC UA

In den S7-1500 CPUs ist ab Firmware V2.0 ein OPC UA-Server implementiert. OPC UA Security umfasst ebenfalls Authentifizierung, Verschlüsselung und Datenintegrität über digitale X.509-Zertifikate und nutzt ebenfalls eine Public Key Infrastructure (PKI). Entsprechend den Anforderungen der Anwendung können Sie verschiedene Security-Stufen für die Endpunkt-Security wählen. Die Beschreibung der OPC UA Server-Funktionalität finden Sie im Kapitel OPC UA-Kommunikation (Seite 135).

3.6.2 Vertraulichkeit durch Verschlüsselung

Ein wichtiger Beitrag zur Datensicherheit ist die Verschlüsselung der Nachrichten. Wenn verschlüsselte Nachrichten auf dem Transportweg von einem Dritten abgefangen werden, kann dieser potentielle Lauscher nichts mit diesen Nachrichten anfangen.

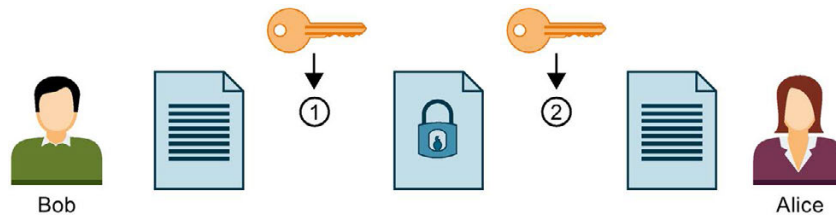
Es gibt eine Vielzahl mathematischer Verfahren (Algorithmen) zur Verschlüsselung von Nachrichten.

Allen Algorithmen gemeinsam ist, dass sie einen Parameter "Schlüssel" verarbeiten, um Nachrichten zu verschlüsseln bzw. zu entschlüsseln.

- Algorithmus + Schlüssel + Nachricht => verschlüsselte Nachricht
- Verschlüsselte Nachricht + Schlüssel + Algorithmus => (entschlüsselte) Nachricht

Symmetrische Verschlüsselung

Wesentlich beim symmetrischen Verschlüsselungsverfahren ist, dass beide Kommunikationspartner zur Verschlüsselung und Entschlüsselung von Nachrichten denselben Schlüssel anwenden, wie in folgendem Bild dargestellt ist: Bob verwendet denselben Schlüssel zum Verschlüsseln wie Alice zum Entschlüsseln. Allgemein sagt man auch, dass beide Seiten als Geheimnis den geheimen Schlüssel teilen, mit dem sie eine Nachricht verschlüsseln oder entschlüsseln können.



- ① Bob verschlüsselt seine Nachricht mit dem symmetrischen Schlüssel
- ② Alice entschlüsselt die verschlüsselte Nachricht mit dem symmetrischen Schlüssel

Bild 3-7 Symmetrische Verschlüsselung

Anschaulich ist das Verfahren mit einem Aktenkoffer vergleichbar, für den Absender und Empfänger jeweils den gleichen Schlüssel haben, um ihn verschließen bzw. öffnen zu können.

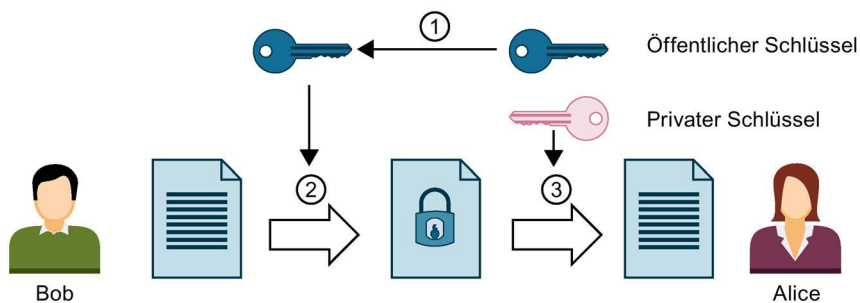
- Vorteil: Symmetrische Verschlüsselungsalgorithmen (z. B. AES, Advanced Encryption Algorithm) arbeiten schnell.
- Nachteile: Wie kommt der Schlüssel zu einem Empfänger, ohne dass er in falsche Hände gerät? Dies ist ein Schlüssel-Verteilungsproblem. Außerdem lässt sich bei einer genügend großer Anzahl abgefangener Nachrichten der Schlüssel erraten, daher muss der Schlüssel ausreichend oft neu vereinbart werden.

Bei einer Vielzahl von Kommunikationspartnern ist außerdem auch eine Vielzahl von Schlüsseln zu verteilen.

Asymmetrische Verschlüsselung

Das asymmetrische Schlüsselverfahren arbeitet mit einem Schlüsselpaar, das aus einem öffentlichen Schlüssel und einem privaten Schlüssel besteht. Im Zusammenhang mit einer PKI wird es auch Public-Key-Verfahren oder nur PKI-Verfahren genannt: Ein Kommunikationspartner, im Bild unten Alice, besitzt einen privaten Schlüssel und einen öffentlichen Schlüssel. Der öffentliche Schlüssel wird der Öffentlichkeit, also jedem potenziellen Kommunikationspartner, zur Verfügung gestellt. Jeder, der den öffentlichen Schlüssel hat, kann Nachrichten für Alice verschlüsseln. Im Bild unten ist das Bob.

Der private Schlüssel von Alice, der von ihr geheim gehalten werden muss, wird von Alice dazu benutzt, um eine an sie adressierte verschlüsselte Nachricht zu entschlüsseln.



- ① Alice stellt Bob ihren öffentlichen Schlüssel zur Verfügung. Dazu sind keine Vorsichtsmaßnahmen erforderlich: Jeder darf den öffentlichen Schlüssel für Nachrichten an Alice nutzen, wenn er sich sicher ist, dass es tatsächlich der öffentliche Schlüssel von Alice ist.
- ② Bob verschlüsselt seine Nachricht mit dem öffentlichen Schlüssel von Alice.
- ③ Alice entschlüsselt die verschlüsselte Nachricht von Bob mit ihrem privaten Schlüssel. Da nur Alice den privaten Schlüssel besitzt und ihn niemals aus der Hand gibt, kann auch nur sie diese Nachricht entschlüsseln. Mit dem privaten Schlüssel kann sie jede Nachricht entschlüsseln, die mit ihrem öffentlichen Schlüssel verschlüsselt wurde - nicht nur die von Bob!

Bild 3-8 Asymmetrische Verschlüsselung

Anschaulich ist das Verfahren mit einem Briefkasten vergleichbar, in den jeder eine Nachricht hineinwerfen kann, aber nur derjenige, der den Schlüssel für den Briefkasten besitzt, kann die Nachricht wieder herausholen.

- Vorteile: Eine mit öffentlichem Schlüssel verschlüsselte Nachricht kann nur vom Inhaber des privaten Schlüssels entschlüsselt werden. Da ein anderer (privater) Schlüssel zum Entschlüsseln benutzt werden muss, ist es auch wesentlich schwerer, den Schlüssel zur Entschlüsselung aus der Menge der verschlüsselten Nachrichten zu erraten. Dadurch müssen die öffentlichen Schlüssel nicht streng geheim aufbewahrt werden, wie es bei symmetrischen Schlüsseln nötig ist.

Ein weiterer Vorteil besteht in der einfacheren Verteilung von öffentlichen Schlüsseln. Beim asymmetrischen Verfahren ist kein besonders gesicherter Kanal erforderlich zur Übertragung der öffentlichen Schlüssel vom Empfänger zum Sender, der die Nachrichten verschlüsselt. Bei der Verwaltung der Schlüssel gibt es also weniger Aufwand als im symmetrischen Verschlüsselungsverfahren.

- Nachteile: Rechenintensiver Algorithmus (z. B. RSA, benannt nach den drei Mathematikern Rivest, Shamir und Adleman), daher geringere Performance im Vergleich zur symmetrischen Verschlüsselung.

Verschlüsselungsverfahren in der Praxis

In der Praxis wie z. B. beim Webserver der CPU und bei der Secure Open User Communication wird das TLS-Protokoll unterhalb der jeweiligen Anwendungsschicht genutzt. Anwendungsschichten sind z. B. HTTP oder SMTP wie im vorhergehenden Abschnitt gezeigt.

TLS (Transport Layer Security) nutzt eine Kombination aus asymmetrischer Verschlüsselung und symmetrischer Verschlüsselung (hybrides Verschlüsselungsverfahren) zur sicheren Datenübertragung z. B. im Internet und nutzt folgende Unterprotokolle:

- TLS Handshake Protocol, zuständig für Authentifizierung der Kommunikationspartner sowie Aushandeln der später für die Datenübertragung zu nutzenden Algorithmen und Schlüssel auf Basis asymmetrischer Verschlüsselungsverfahren.
- TLS Record Protocol, zuständig für Verschlüsselung der Nutzdaten mittels symmetrischer Verschlüsselungsverfahren und Datenaustausch.

Sowohl die asymmetrische als auch die symmetrische Verschlüsselung gelten als sichere Verschlüsselungsverfahren - es gibt bezüglich der Verfahren keinen prinzipiellen Unterschied hinsichtlich der Sicherheit. Der Grad der Sicherheit hängt ab von den Parametern wie z. B. von der gewählten Schlüssellänge.

Missbrauch von Verschlüsselung

Man sieht einem öffentlichen Schlüssel als eine Bitfolge nicht an, welcher Identität dieser öffentliche Schlüssel zugeordnet ist. Ein Betrüger könnte seinen öffentlichen Schlüssel zur Verfügung stellen und behaupten, er sei eine ganz andere Person. Wenn ein Dritter diesen Schlüssel im Glauben nutzt, er hätte den gewünschten Kommunikationspartner adressiert, landen möglicherweise vertrauliche Informationen bei dem Betrüger. Der Betrüger entschlüsselt dann mit seinem privaten Schlüssel die gar nicht für ihn bestimmte Nachricht und vertrauliche Informationen geraten dann in die falschen Hände.

Um solchen Missbrauch zu verhindern, muss bei den Kommunikationspartnern das Vertrauen geschaffen werden, dass sie es mit dem gewünschten Kommunikationspartner zu tun haben. Um dieses Vertrauen herzustellen, nutzt man in einer PKI digitale Zertifikate.

3.6.3 Authentizität und Integrität durch Signaturen

Angriffe von Programmen, welche die Kommunikation zwischen Server und Client abfangen und agieren, als wären sie selbst Client oder Server, nennt man "Man-In-The-Middle Attacks". Wenn die falsche Identität dieser Programme nicht erkannt wird, können sie z. B. wichtige Informationen über das S7-Programm erhalten oder Werte in der CPU setzen und damit eine Maschine oder Anlagen angreifen. Zur Vermeidung solcher Angriffe werden digitale Zertifikate verwendet.

Secure Communication verwendet digitale Zertifikate, die dem Standard X.509 der International Telecommunication Union (ITU) entsprechen. Damit lässt sich die Identität eines Programms, eines Rechners oder einer Organisation prüfen (authentifizieren).

Wie Zertifikate Vertrauen schaffen

Die wesentliche Aufgabe von X.509-Zertifikaten ist es, eine Identität mit den Daten eines Zertifikatsinhabers (z. B. E-Mail-Adresse, Rechnername) an den öffentlichen Schlüssel der Identität zu binden. Identitäten können Personen, Rechner oder Maschinen sein.

Zertifikate werden von Zertifizierungsstellen (Certificate Authority, CA) oder dem Inhaber des Zertifikates selber ausgegeben. PKI-Systeme legen fest, wie die Nutzer den Zertifizierungsstellen und den von ihnen herausgegebenen Zertifikaten trauen können.

Der Weg zum Zertifikat:

1. Wer ein Zertifikat haben möchte, reicht über eine bei der Zertifizierungsstelle angeschlossene Registrierungsstelle einen Zertifikatsantrag ein.
2. Die Zertifizierungsstelle bewertet Antrag und Antragsteller anhand festgelegter Kriterien.
3. Wenn sich die Identität des Antragstellers eindeutig feststellen lässt, beglaubigt die Zertifizierungsstelle diese Identität durch Ausstellen eines signierten Zertifikats. Aus dem Antragsteller ist nun der Zertifikatsinhaber geworden.

Im folgenden Bild ist der Sachverhalt vereinfacht dargestellt. Nicht gezeigt ist die Möglichkeit, wie Alice die digitale Signatur prüfen kann.

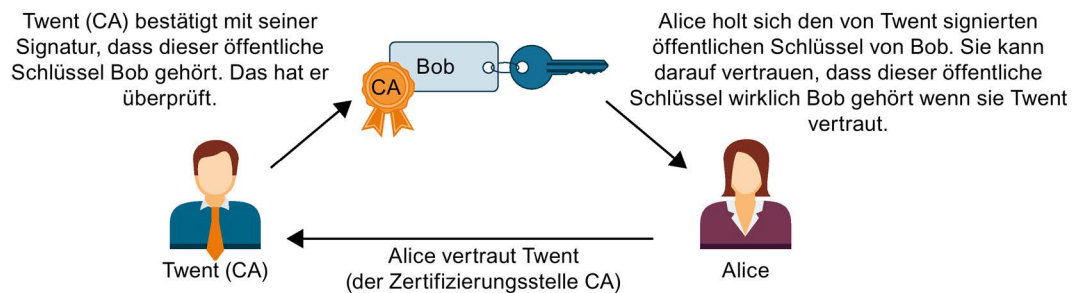


Bild 3-9 Signieren eines Zertifikates durch eine Zertifizierungsstelle

Selbstsignierte Zertifikate

Selbstsignierte Zertifikate sind Zertifikate, dessen Signatur vom Zertifikateinhaber stammt und nicht von einer unabhängigen Zertifizierungsstelle.

Beispiele:

- Sie können ein Zertifikat erstellen und selbst signieren, um zum Beispiel Nachrichten zu einem Kommunikationspartner zu verschlüsseln. Im Beispiel oben könnte Bob selber (statt Twent) sein Zertifikat mit seinem privaten Schlüssel signieren. Alice kann mit Hilfe von Bobs öffentlichem Schlüssel prüfen, dass Signatur und öffentlicher Schlüssel von Bob zusammenpassen. Für eine einfache Anlagen-interne Kommunikation, die verschlüsselt ablaufen soll, ist das ausreichend.
- Bei einem Stammzertifikat handelt es sich z. B. um ein von der Zertifizierungsstelle (Aussteller) selbstsigniertes Zertifikat, welches den öffentlichen Schlüssel der Zertifizierungsstelle enthält.

Besonderheiten von selbstsignierten Zertifikaten

Die Attribute "CN" (Common Name of Subject) für den Zertifikatsinhaber und "Issuer" (Aussteller) von selbstsignierten Zertifikaten sind identisch: Sie haben Ihr Zertifikat ja selbst signiert. Das Feld "CA" (Certificate Authority) für die Zertifizierungsstelle muss auf "False" stehen; das selbstsignierte Zertifikat soll ja nicht dazu benutzt werden, andere Zertifikate zu signieren.

Selbstsignierte Zertifikate sind nicht in eine PKI-Hierarchie eingebettet.

Inhalte von Zertifikaten

Ein Zertifikat nach dem Standard X.509 V3, der Standard, der auch von STEP 7 bzw. den S7-1500 CPUs genutzt wird, besteht im Wesentlichen aus folgenden Teilen:

- Öffentlicher Schlüssel
- Angaben über den Zertifikatsinhaber (d. h. den Schlüsselinhaber); das ist z. B. der Common Name (CN) of Subject
- Attribute wie Seriennummer und Gültigkeitsdauer
- Digitale Signatur (Beglaubigung) der Zertifizierungsstelle (CA), dass die Angaben stimmen.

Daneben gibt es Erweiterungen, z. B.

- Angabe, für welchen Verwendungszweck der öffentliche Schlüssel verwendet werden darf (Key Usage), z. B. zum Signieren oder zur Schlüssel-Verschlüsselung. Wenn Sie mit STEP 7 ein neues Zertifikat erstellen, z. B. im Kontext Secure Open User Communication, wählen Sie aus der Liste der möglichen Verwendungszwecke den treffenden Eintrag aus der Liste aus, z. B. "TLS".
- Angabe eines "Alternativen Namens des Zertifikatsinhabers" ("SAN", Subject Alternative Name), der z. B. bei der sicheren Kommunikation mit Webservern (HTTP over TLS) dazu genutzt wird, um sicherzustellen, dass das Zertifikat auch dem Webserver gehört, der im URL der Adresszeile des Web-Browsers angegeben wurde.

Wie Signaturen erzeugt und verifiziert werden

Die technische Voraussetzung, dass Zertifikate geprüft werden können, liefert die asymmetrische Schlüsselerzeugung: Am Beispiel des Zertifikats "MyCert" werden die Prozesse "Signieren" und "Signatur prüfen" gezeigt.

Signatur erzeugen:

1. Der Aussteller des Zertifikates "MyCert" erzeugt aus den Daten des Zertifikats mit einer bestimmten Hash-Funktion (z.B. SHA-1, Secure Hash Algorithm) einen Hash-Wert.

Der Hash-Wert ist eine Bitfolge mit konstanter Länge. Die stets gleiche Länge des Hash-Wertes bietet den Vorteil, dass das Signieren des Hash-Wertes immer die gleiche Zeit beansprucht.

2. Aus dem so erzeugten Hash-Wert erzeugt der Aussteller des Zertifikats mit Hilfe des privaten Schlüssels eine digitale Signatur. Häufig wird dazu das RSA-Signaturverfahren benutzt.
3. Die digitale Signatur wird im Zertifikat gespeichert. Dadurch ist das Zertifikat signiert.

Signatur verifizieren:

1. Der Prüfer des Zertifikats "MyCert" besorgt sich das Zertifikat des Ausstellers und damit den öffentlichen Schlüssel.
2. Mit demselben Hash-Algorithmus, der bei der Signierung verwendet wurde (z.B. SHA-1), wird aus den Daten des Zertifikats erneut ein Hash-Wert gebildet.
3. Dieser Hash-Wert wird dann verglichen mit dem Hash-Wert, der mit Hilfe des öffentlichen Schlüssel des Zertifikate-Ausstellers und dem Signaturalgorithmus zur Prüfung der Signatur ermittelt wird.
4. Wenn die Prüfung der Signatur ein positives Ergebnis liefert, ist sowohl die Identität des Zertifikatsinhabers als auch die Integrität, d. h. die Echtheit und Unverfälschtheit des Zertifikate-Inhalts nachgewiesen. Jeder, der den öffentlichen Schlüssel, d. h. das Zertifikat der Zertifizierungsstelle hat, kann die Signatur prüfen und so erkennen, dass das Zertifikat tatsächlich von der Zertifizierungsstelle signiert wurde.

Im folgenden Bild ist gezeigt, wie Alice mit Hilfe des öffentlichen Schlüssels des Zertifikats von Twent (verkörpert die Zertifizierungsstelle CA) die Signatur an Bobs öffentlichem Schlüssel verifiziert. Voraussetzung für die Prüfung ist also die Verfügbarkeit des Zertifikats der Zertifizierungsstelle zum Prüfungszeitpunkt. Die Validierung selbst läuft automatisch in der TLS-Session ab.

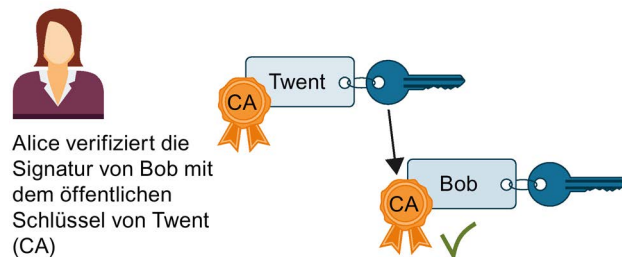


Bild 3-10 Verifizierung eines Zertifikats über den öffentlichen Schlüssel des Zertifikates einer Zertifizierungsstelle

Signatur von Nachrichten

Die oben beschriebene Methode zum Signieren und Verifizieren nutzt die TLS-Session auch zum Signieren und Verifizieren von Nachrichten:

Wenn von einer Nachricht ein Hash-Wert erzeugt wird und dieser Hash-Wert mit dem privaten Schlüssel des Senders signiert und an die originale Nachricht angehängt wird, ist der Empfänger der Nachricht in der Lage, die Integrität (Unversehrtheit) der Nachricht zu erkennen. Der Empfänger entschlüsselt den Hash-Wert mit dem öffentlichen Schlüssel des Senders, bildet aus der empfangenen Nachricht selbst den Hash-Wert und vergleicht beide Werte. Wenn die Werte unterschiedlich sind, wurde die Nachricht auf dem Transportweg verfälscht.

Kette von Zertifikaten bis zum Stammzertifikat

Die Zertifikate einer PKI sind häufig hierarchisch organisiert: An der Spitze der Hierarchie stehen Stammzertifikate, auch Wurzelzertifikate oder Root-Zertifikate genannt. Das sind Zertifikate, die nicht durch eine übergeordnete Zertifizierungsstelle beglaubigt werden. Zertifikatsinhaber und Zertifikatsaussteller von Stammzertifikaten sind identisch. Stammzertifikate genießen absolutes Vertrauen, sie sind der "Anker" des Vertrauens und müssen deshalb beim Empfänger als vertrauenswürdige Zertifikate bekannt sein. Sie werden in einem für vertrauenswürdige Zertifikate vorgesehenen Bereich gespeichert.

Die Funktion von Stammzertifikaten kann je nach PKI z. B. darin bestehen, Zertifikate von untergeordneten Zertifizierungsstellen, sogenannte Zwischenzertifikate, zu signieren. Damit überträgt sich das Vertrauen vom Stammzertifikat auf das Zwischenzertifikat. Ein Zwischenzertifikat kann genauso gut ein Zertifikat signieren wie ein Stammzertifikat, daher werden beide auch "CA-Zertifikate" genannt.

Diese Hierarchie lässt sich über mehrere Zwischenzertifikate weiterführen bis zum End-Entity Zertifikat. Das End-Entity Zertifikat ist das Zertifikat des Benutzers, der identifiziert werden soll.

Bei der Validierung wird die Hierarchie in umgekehrter Richtung durchlaufen: Wie oben beschrieben wird der Zertifikatsaussteller ermittelt, mit seinem öffentlichen Schlüssel die Signatur geprüft, dann das Zertifikat des übergeordneten Zertifikatsausstellers ermittelt bis die Vertrauenskette bis zum Stammzertifikat durchlaufen ist.

Fazit: Die Kette von Zwischenzertifikaten bis zum Stammzertifikat, der Zertifikate-Pfad, muss in jedem Gerät vorhanden sein, das ein End-Entity-Zertifikat vom Kommunikationspartner validieren soll, unabhängig davon, welche Art von Secure Communication Sie projektieren.

3.6.4 Verwalten von Zertifikaten mit STEP 7

STEP 7 ab Version V14 zusammen mit den S7-1500-CPU's ab FW Version 2.0 unterstützen die Internet-PKI (RFC 5280) soweit, dass eine S7-1500-CPU in der Lage ist, mit Geräten zu kommunizieren, die ebenfalls die Internet PKI unterstützen.

Die Nutzung von X.509-Zertifikaten z. B. zur Prüfung von Zertifikaten wie in den vorangegangenen Abschnitten beschrieben ist eine Konsequenz hieraus.

STEP 7 ab V14 nutzt eine PKI ähnlich der Internet PKI. Nicht unterstützt werden z. B. Certificate Revocation Lists (CRLs).

Zertifikate erstellen oder zuweisen

Für Geräte mit Security-Eigenschaften wie z. B. eine S7-1500-CPU ab Firmware V2.0 erstellen Sie in STEP 7-Zertifikate für verschiedene Anwendungen.

Folgende Bereiche im Inspektorfenster der CPU erlauben das Erstellen neuer oder das Auswählen vorhandener Zertifikate:

- "Schutz & Security > Zertifikatsmanager" - für die Erzeugung bzw. Zuweisung aller Arten von Zertifikaten; voreingestellt für das Erstellen von Zertifikaten sind TLS-Zertifikate für Secure Open User Communication.
- "Webserver > Server-Security" - für die Erzeugung bzw. Zuweisung von Webserver-Zertifikaten.
- "OPC UA > Server > Security" - für die Erzeugung bzw. Zuweisung von OPC UA-Zertifikaten

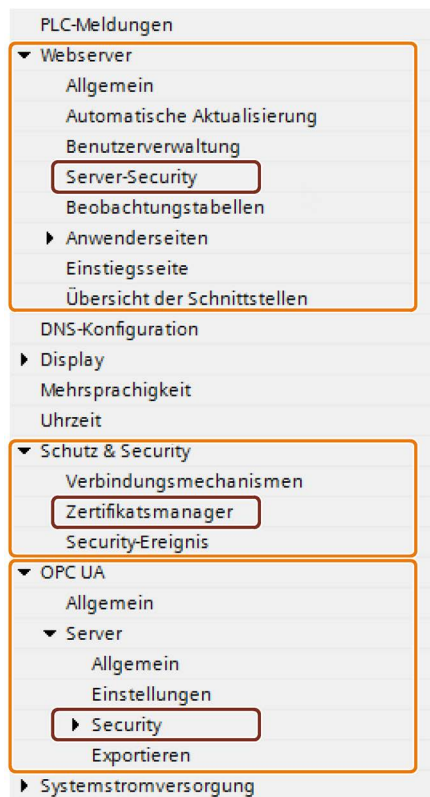


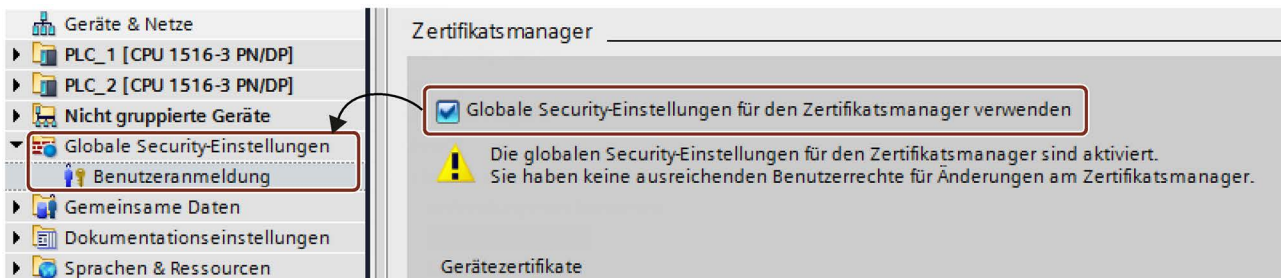
Bild 3-11 Security-Einstellungen für eine S7-1500 CPU in STEP 7

Besonderheit des Bereichs "Schutz & Security > Zertifikatsmanager"

Nur in diesem Bereich des Inspektorfensters schalten Sie zwischen dem globalen, d. h. projektweitem und lokalen, d. h. gerätespezifischem Zertifikatsmanager um (Option "Globale Security-Einstellungen für den Zertifikatsmanager verwenden"). Die Option entscheidet darüber, ob Sie Zugriff auf alle Zertifikate im Projekt haben oder nicht.

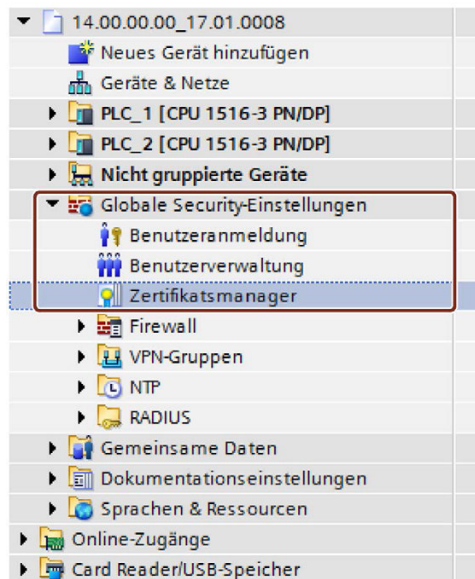
- Wenn Sie den Zertifikatsmanager in den globalen Security-Einstellungen **nicht** verwenden, haben Sie nur Zugriff auf den lokalen Zertifikatsspeicher der CPU. Sie haben keinen Zugriff z. B. auf importierte Zertifikate oder Stammzertifikate. Ohne diese Zertifikate ist nur eine eingeschränkte Funktionalität verfügbar; Sie können z. B. nur selbstsignierte Zertifikate erzeugen.
- Wenn Sie den Zertifikatsmanager in den globalen Security-Einstellungen verwenden und Sie z. B. als Administrator angemeldet sind, haben Sie Zugriff auf den globalen, projektweiten Zertifikatsspeicher. Sie können z. B. der CPU importierte Zertifikate zuweisen oder Zertifikate erstellen, die von der Projekt-CA (Zertifizierungsstelle des Projekts) ausgestellt und signiert sind.

Das folgende Bild zeigt, wie nach dem Aktivieren der Option "Globale Security-Einstellungen für den Zertifikatemananger verwenden" im Inspektorfenster der CPU die "Globalen Security-Einstellungen" in der Projektnavigation erscheinen.



Wenn Sie in der Projektnavigation auf "Benutzeranmeldung" unterhalb der Globalen Security-Einstellungen doppelklicken und sich anmelden, erscheint dort auch unter anderem eine Zeile "Zertifikatsmanager".

Mit einem Doppelklick auf die Zeile "Zertifikatsmanager" erhalten Sie Zugang zu allen Zertifikaten im Projekt, aufgeteilt in die Register "CA" (Zertifizierungsstellen), "Gerätezertifikate" und "Vertrauenswürdige Zertifikate und Stammzertifizierungsstellen".



Private Schlüssel

STEP 7 erzeugt private Schlüssel beim Erzeugen von Gerätezertifikaten bzw. Server-Zertifikaten (End-Entity-Zertifikate). Wo der private Schlüssel verschlüsselt gespeichert wird, hängt von der Verwendung der globalen Security-Einstellungen für den Zertifikatsmanager ab:

- Wenn Sie die globalen Security-Einstellungen verwenden, dann wird der private Schlüssel im globalen (projektweiten) Zertifikatsspeicher verschlüsselt gespeichert.
- Wenn Sie die globalen Security-Einstellungen nicht verwenden, dann wird der private Schlüssel im lokalen (CPU-spezifischen) Zertifikatsspeicher verschlüsselt gespeichert.

Angezeigt wird das Vorhandensein des privaten Schlüssels, der z. B. für die Entschlüsselung von Daten notwendig ist, in der Spalte "Privater Schlüssel" im Register "Gerätezertifikate" des Zertifikatsmanagers in den globalen Security-Einstellungen.

Beim Laden der Hardware-Konfiguration wird das Gerätezertifikat, der öffentliche Schlüssel sowie der private Schlüssel in die CPU geladen.

ACHTUNG

Die Option "Globale Security-Einstellungen für den Zertifikatsmanager verwenden" beeinflusst die bisher verwendeten privaten Schlüssel: Wenn Sie bereits Zertifikate erstellt haben ohne Verwendung des Zertifikatsmanagers in den globalen Security-Einstellungen und dann die Option zur Verwendung des Zertifikatsmanagers umstellen, dann gehen die privaten Schlüssel verloren und die Zertifikats-ID kann sich ändern! Eine Warnung macht Sie auf diesen Sachverhalt aufmerksam. Legen Sie daher zu Beginn der Projektierung fest, welche Option zum Zertifikatsmanagement erforderlich ist.

3.6.5 Beispiele zum Verwalten von Zertifikaten

Wie in den vorangegangenen Abschnitten erläutert, sind Zertifikate für jede Art von Secure Communication erforderlich. Im Folgenden wird beispielhaft gezeigt, wie Sie mit STEP 7 die Zertifikate handhaben, damit die Voraussetzungen für Secure Open User Communication gegeben sind.

Dabei wird im Folgenden unterschieden, um welche Geräte es sich bei den beteiligten Kommunikationspartnern handelt. Die jeweiligen Schritte zum Versorgen der Kommunikationsteilnehmer mit den benötigten Zertifikaten sind jeweils beschrieben. Vorausgesetzt wird immer eine S7-1500 CPU bzw. ein S7-1500 Software Controller ab Firmware Version 2.0.

Allgemein gilt:

Während des Aufbaus einer sicheren Verbindung ("Handshake") übermitteln die Kommunikationspartner in der Regel nur ihre End-Entity-Zertifikate (Gerätezertifikate).

Daher müssen sich die zur Prüfung des übermittelten Gerätezertifikats notwendigen CA-Zertifikate im Zertifikatsspeicher des jeweiligen Kommunikationspartners befinden.

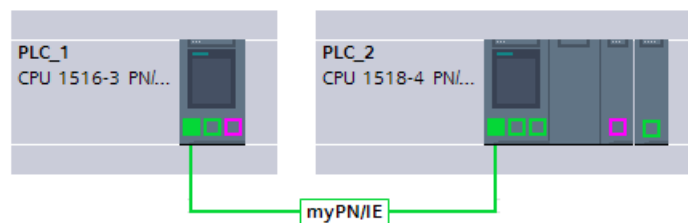
Secure Open User Communication zwischen zwei S7-1500 CPUs

Zwei S7-1500-CPU's PLC_1 und PLC_2 sollen miteinander Daten austauschen über Secure Open User Communication.

Die erforderlichen Gerätezertifikate erzeugen Sie mit STEP 7 und weisen Sie den CPUs zu wie im Folgenden beschrieben.

Es werden STEP 7-Projekt-Zertifizierungsstellen (CA des Projekts) verwendet, um die Gerätezertifikate zu signieren.

Die Zertifikate sind im Anwenderprogramm (Kommunikationsanweisung TCON in Verbindung mit dem zugehörigen Sytemdatentyp, z. B. TCON_IPV4_SEC) über ihre Zertifikats-ID zu referenzieren. Die Zertifikats-ID vergibt STEP 7 automatisch beim Erzeugen oder beim Anlegen von Zertifikaten.



Vorgehen

STEP 7 lädt automatisch die erforderlichen CA-Zertifikate zusammen mit der Hardware-Konfiguration in die beteiligten CPUs, so dass die Voraussetzungen für die Zertifikatsprüfung für beide CPUs gegeben sind. Sie müssen also nur die Gerätezertifikate für die jeweilige CPU erzeugen - alles Übrige erledigt STEP 7 für Sie.

1. Markieren Sie PLC_1 und aktivieren Sie im Bereich "Schutz & Security" die Option "Globale Security-Einstellungen für den Zertifikatsmanager verwenden".
2. Melden Sie sich in der Projektnavigation im Bereich "Globale Security-Einstellungen" als Benutzer an. Bei einem neuen Projekt ist bei der erstmaligen Anmeldung die Rolle "Administrator" vorgesehen.
3. Kehren Sie zurück zur PLC-1 in den Bereich "Schutz & Security". Klicken Sie in der Tabelle "Gerätezertifikate" in eine leere Zeile der Spalte "Zertifikatsinhaber", um ein neues Zertifikat hinzuzufügen.
4. Klicken Sie in der Klappliste zur Auswahl eines Zertifikats auf die Schaltfläche "Hinzufügen".

Der Dialog "Neues Zertifikat erzeugen" wird geöffnet.

5. Belassen Sie die Voreinstellungen in diesem Dialog; sie sind auf die Verwendung für Secure Open User Communication zugeschnitten (Verwendung: TLS).

Tipp: Ergänzen Sie den voreingestellten Namen des Zertifikatsinhabers, in diesem Fall den CPU-Namen. Belassen Sie zur besseren Unterscheidung den voreingestellten CPU-Namen für den Fall, dass Sie viele Gerätezertifikate verwalten müssen.

Beispiel: PLC_1/TLS wird zu PLC_1-SecOUC-Chassis17FactoryState.

6. Übersetzen Sie die Konfiguration.

Das Gerätezertifikat und das CA-Zertifikat sind Bestandteil der Konfiguration.

7. Wiederholen Sie die beschriebenen Schritte für PLC_2.

Im nächsten Schritt müssen Sie die Anwenderprogramme für den Datenaustausch erstellen und die Konfigurationen zusammen mit dem Programm laden.

Selbstsignierte Zertifikate statt CA-Zertifikate verwenden

Beim Anlegen von Gerätezertifikaten können Sie die Option "Selbstsigniert" wählen. Selbstsignierte Zertifikate können Sie erstellen, ohne für die globalen Security-Einstellungen angemeldet zu sein. Diese Vorgehensweise wird nicht empfohlen, da die so erstellten Zertifikate nicht im globalen Zertifikatsspeicher vorhanden sind und daher nicht direkt einer Partner-CPU zugewiesen werden können.

Wie oben beschrieben sollten Sie sorgfältig den Namen des Zertifikateinhabers wählen, um das richtige Zertifikat zweifelsfrei einem Gerät zuordnen zu können.

Für selbstsignierte Zertifikate ist keine Prüfung mit den CA-Zertifikaten des STEP 7-Projekts möglich. Um selbstsignierte Zertifikate prüfen zu können, müssen Sie für jede CPU das selbstsignierte Zertifikat des Kommunikationspartners in die Liste der vertrauenswürdigen Partnergeräte aufnehmen. Dazu müssen Sie die Option "Globale Security-Einstellungen für den Zertifikatsmanager verwenden" aktiviert haben und in den globalen Security-Einstellungen als Benutzer angemeldet sein.

Um das selbstsignierte Zertifikat vom Kommunikationspartner der CPU hinzuzufügen, gehen sie folgendermaßen vor:

1. Markieren Sie PLC_1 und navigieren Sie zur Tabelle "Zertifikate von Partnergeräten" im Bereich "Schutz & Security".
2. Klicken Sie in eine leere Zeile der Spalte "Zertifikatsinhaber", um die Klappliste zum Hinzufügen oder Auswählen von Zertifikaten zu öffnen.
3. Wählen Sie aus der Klappliste das selbstsignierte Zertifikat des Kommunikationspartners und bestätigen Sie die Auswahl.

Im nächsten Schritt müssen Sie die Anwenderprogramme für den Datenaustausch erstellen und die Konfigurationen zusammen mit dem Programm laden.

Secure Open User Communication zwischen S7-1500 CPU als TLS-Client und Fremdgerät als TLS-Server

Zwei Geräte sollen miteinander Daten austauschen über eine TLS-Verbindung bzw. TLS-Sitzung, z. .B zum Austausch von Rezepturen, Produktionsdaten oder Qualitätsdaten:

- Eine S7-1500-CPU (PLC_1) als TLS-Client; die CPU nutzt Secure Open User Communication
- Ein Fremdgerät (z. B. ein Manufacturing Execution System (MES) als TLS-Server

Die S7-1500 CPU baut als TLS-Client die TLS-Verbindung/Sitzung zum MES-System auf.



- ① TLS-Client
② TLS-Server

Zur Authentifizierung des TLS-Servers benötigt die S7-1500 CPU die CA-Zertifikate des MES-Systems: Das Stammzertifikat und ggf. die Zwischenzertifikate zur Prüfung des Zertifikatspfads.

Diese Zertifikate müssen Sie in den globalen Zertifikatespeicher der S7-1500 CPU importieren.

Um Zertifikate des Kommunikationspartners zu importieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie in der Projektnavigation in den globalen Security-Einstellungen den Zertifikatsmanager.
2. Wählen Sie für das zu importierende Zertifikat die passende Tabelle (Vertrauenswürdige Zertifikate und Stammzertifizierungsstellen).

3. Öffnen Sie in der Tabelle mit Rechtsklick das Kontextmenü. Klicken Sie auf "Importieren" und importieren Sie das benötigte Zertifikat bzw. die benötigten CA-Zertifikate.

Das Zertifikat erhält durch den Import eine Zertifikats-ID und kann im nächsten Schritt einer Baugruppe zugewiesen werden.

4. Markieren Sie PLC_1 und navigieren Sie zur Tabelle "Zertifikate von Partnergeräten" im Bereich "Schutz & Security".
5. Klicken Sie in eine leere Zeile der Spalte "Zertifikatsinhaber", um die importierten Zertifikate hinzuzufügen.
6. Wählen Sie aus der Klappliste die benötigten CA-Zertifikate des Kommunikationspartners und bestätigen Sie die Auswahl.

Optional kann das MES-System zur Authentifizierung der CPU (d. h. des TLS-Clients) ebenfalls ein Gerätezertifikat der CPU anfordern. Dem MES-System müssen in diesem Fall die CA-Zertifikate der CPU zur Verfügung gestellt werden. Voraussetzung für den Import der Zertifikate ins MES-System ist ein vorhergehender Export der CA-Zertifikate aus dem STEP 7-Projekt der CPU. Gehen sie folgendermaßen vor:

1. Öffnen Sie in der Projektnavigation in den globalen Security-Einstellungen den Zertifikatsmanager.
2. Wählen Sie für das zu exportierende Zertifikat die passende Tabelle (CA-Zertifikate).
3. Öffnen Sie bei selektiertem Zertifikat mit Rechtsklick das Kontextmenü.
4. Klicken Sie auf "Exportieren".
5. Wählen Sie das Exportformat des Zertifikats.

Im nächsten Schritt müssen Sie die Anwenderprogramme für den Datenaustausch erstellen und die Konfigurationen zusammen mit dem Programm laden.

Secure Open User Communication zwischen S7-1500 CPU als TLS-Server und Fremdgerät als TLS-Client

Wenn die S7-1500 CPU als TLS-Server agiert und das Fremdgerät, z. B. ein ERP-System (Enterprise Resource Planning System) die TLS-Verbindung/Sitzung aufbaut, benötigen Sie folgende Zertifikate:

- Für die S7-1500 CPU erzeugen Sie ein Gerätezertifikat (Server-Zertifikat) mit privatem Schlüssel und laden es mit der Hardware-Konfiguration in die S7-1500 CPU. Sie verwenden beim Erzeugen des Server-Zertifikats die Option "Von Zertifizierungsstelle signiert".

Der private Schlüssel wird für den Schlüsselaustausch benötigt, wie im Bild für das Beispiel "HTTP over TLS" erläutert.

- Für das ERP-System müssen Sie das CA-Zertifikat des STEP 7-Projekts exportieren und in das ERP-System importieren/laden. Mit dem CA-Zertifikat prüft das ERP-System das Serverzertifikat der S7-1500, das von der CPU an das ERP-System während des Aufbaus der TLS-Verbindung/Sitzung übermittelt wird.



- ① TLS-Server
- ② TLS-Client

Bild 3-12 Secure OUC zwischen einer S7-1500 CPU und einem ERP-System

Die Beschreibung der erforderlichen Handlungsschritte entnehmen Sie dem vorangegangenen Abschnitten.

Secure Open User Communication zu einem Mailserver (SMTP over TLS)

Eine S7-1500 CPU kann mit der Kommunikationsanweisung TMAIL-C eine sichere Verbindung zu einem E-Mail-Server aufbauen.

Die Systemdatentypen TMail_V4_SEC und TMail_QDN_SEC ermöglichen Ihnen, den Partnerport des E-Mail-Servers zu bestimmen und so den E-Mail-Server über "SMTP over TLS" zu erreichen.



Bild 3-13 Secure OUC zwischen einer S7-1500 CPU und einem Mail Server

Notwendige Voraussetzung für eine sichere E-Mail-Verbindung ist ein Import des Stammzertifikats und der Zwischenzertifikate vom Mailserver (Provider) in den globalen Zertifikatsspeicher der S7-1500 CPU. Mithilfe dieser Zertifikate kann die CPU das Serverzertifikat prüfen, das beim Aufbau der TLS-Verbindung/Sitzung vom Mail-Server gesendet wird.

Um Zertifikate des Mailservers zu importieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie in der Projektnavigation in den globalen Security-Einstellungen den Zertifikatsmanager.
2. Wählen Sie für das zu importierende Zertifikat die passende Tabelle (Vertrauenswürdige Zertifikate und Stammzertifizierungsstellen).
3. Öffnen Sie in der Tabelle mit Rechtsklick das Kontextmenü. Klicken Sie auf "Importieren" und importieren Sie das benötigte Zertifikat bzw. die benötigten CA-Zertifikate.
Das Zertifikat erhält durch den Import eine Zertifikats-ID und kann im nächsten Schritt einer Baugruppe zugewiesen werden.
4. Markieren Sie PLC_1 und navigieren Sie zur Tabelle "Zertifikate von Partnergeräten" im Bereich "Schutz & Security".
5. Klicken Sie in eine leere Zeile der Spalte "Zertifikatsinhaber", um die importierten Zertifikate hinzuzufügen.
6. Wählen Sie aus der Klappliste die benötigten CA-Zertifikate des Kommunikationspartners und bestätigen Sie die Auswahl.

Im nächsten Schritt müssen Sie die Anwenderprogramme für die E-Mail-Client-Funktion der CPU erstellen und die Konfigurationen zusammen mit dem Programm laden.

3.6.6 Beispiel: HTTP over TLS

Im Folgenden wird gezeigt, wie die beschriebenen Mechanismen genutzt werden, um eine Secure Communication zwischen einem Webbrowser und dem Webserver einer S7-1500 CPU aufzubauen.

Zunächst sind die Änderungen für die Option "Zugriff nur über HTTPS" in STEP 7 beschrieben. Ab STEP 7 V14 haben Sie die Möglichkeit, Einfluss auf das Server-Zertifikat des Webserver einer S7-1500-CPU ab Firmware V2.0 zu nehmen: Das Server-Zertifikat wird ab diesen Versionen mit STEP 7 erzeugt.

Außerdem wird gezeigt, welche Prozesse beim Aufruf einer Webseite des Webserver der CPU mit einem Webbrowser eines PCs über eine verschlüsselte HTTPS-Verbindung ablaufen.

Umgang mit Webserver-Zertifikaten für S7-1500 CPUs ab FW V2.0

Für S7-1500 CPUs mit einem Firmware-Stand kleiner V2.0 haben Sie beim Einstellen der Webserver-Eigenschaften ohne Voraussetzungen die Option "Zugriff nur über HTTPS zulassen" wählen können.

Mit der Hantierung von Zertifikaten sind Sie bei diesen CPUs nicht in Berührung gekommen; die erforderlichen Zertifikate für den Webserver erzeugt die CPU automatisch.

Bei S7-1500 CPUs ab Firmware V2.0 erzeugt STEP 7 das Server-Zertifikat (End-Entity-Zertifikat) für die CPU. In den Eigenschaften der CPU (Webserver > Server Security) weisen Sie dem Webserver ein Server-Zertifikat zu.

Da immer ein Server-Zertifikatsname voreingestellt ist, ändert sich an der einfachen Projektierung vom Webserver nichts: Sie aktivieren den Webserver und aktivieren die Option "Zugriff nur über HTTPS zulassen" - STEP 7 generiert beim Übersetzen ein Server-Zertifikat mit dem voreingestellten Namen.

Unabhängig davon, ob Sie den Zertifikatsmanager in den globalen Security-Einstellungen verwenden oder nicht: STEP 7 hat alle Informationen, um das Server-Zertifikat erzeugen zu können.

Zusätzlich haben Sie die Möglichkeit, die Eigenschaften des Server-Zertifikats zu bestimmen, z. B. den Namen oder die Gültigkeitsdauer.

Laden des Webserver-Zertifikats

Mit dem Laden der Hardware-Konfiguration in die CPU wird das von STEP 7 erzeugte Server-Zertifikat automatisch mitgeladen.

- Wenn Sie den Zertifikatsmanager in den globalen Security-Einstellungen verwenden, signiert die Zertifizierungsstelle des Projekts (CA-Zertifikat) das Server-Zertifikat des Webserver. Beim Laden wird das CA-Zertifikat des Projekts automatisch mitgeladen.
- Wenn Sie den Zertifikatsmanager in den globalen Security-Einstellungen nicht verwenden, erzeugt STEP 7 das Server-Zertifikat als selbstsigniertes Zertifikat.

Wenn Sie den Webserver der CPU über die IP-Adresse der CPU adressieren, dann ist mit jeder Änderung der IP-Adresse einer Ethernet-Schnittstelle der CPU ein neues Serverzertifikat (End-Entity-Zertifikat) zu erzeugen und zu laden. Der Grund ist, dass sich mit der IP-Adresse die Identität der CPU ändert - und die muss nach den Regeln der PKI beglaubigt (signiert) werden.

Sie können dieses Problem vermeiden, indem Sie die CPU über einen Domainnamen adressieren statt mit ihrer IP-Adresse, z. B. "myconveyer-cpu.room13.myfactory.com". Dazu müssen Sie die Domainnamen der CPUs über einen DNS-Server verwalten.

Webbrowser mit CA-Zertifikat des Webserver versorgen

Im Webbrowser sollte der Anwender, der per HTTPS auf die Webseiten der CPU zugreift, das CA-Zertifikat der CPU installieren. Wenn kein Zertifikat installiert ist, wird nämlich eine Warnung angezeigt, mit der Empfehlung, die Seite nicht zu benutzen. Um die Seite zu sehen, muss der Anwender dann explizit eine "Ausnahme hinzufügen".

Das gültige Wurzelzertifikat (Certification Authority) erhält der Anwender als Download auf der Webseite "Intro" des CPU-Webserver unter "Zertifikat herunterladen".

Eine andere Möglichkeit bietet STEP 7: Exportieren Sie das CA-Zertifikat des Projekts mit dem Zertifikatsmanager in den Globalen Security-Einstellungen in STEP 7. Anschließend importieren Sie das CA-Zertifikat in den Browser.

Ablauf der Secure Communication

Das folgende Bild zeigt vereinfacht den prinzipiellen Ablauf des Aufbaus der Kommunikation ("Handshake") mit dem Schwerpunkt auf dem Aushandeln der Schlüssel, die zum Datenaustausch (hier über HTTP over TLS) verwendet werden.

Der Ablauf ist aber prinzipiell übertragbar auf alle Kommunikationsmöglichkeiten, die auf der Nutzung von TLS basieren, also auch für Secure Open User Communication (siehe Grundlagen zur Secure Communication).

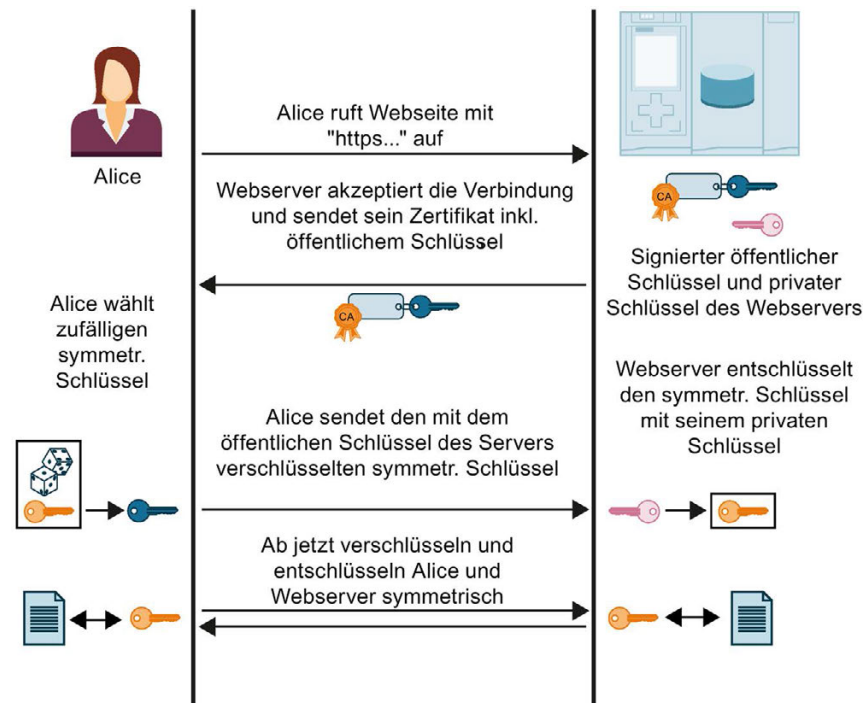


Bild 3-14 Handshake bei https

Nicht dargestellt im Bild sind die Aktionen, die auf Seiten von Alice (Browser) ablaufen, um das vom Webserver gesendete Zertifikat zu prüfen. Vom positiven Ausgang der Prüfung hängt es ab, ob Alice dem übermittelten Webserver-Zertifikat und damit der Identität des Webserver trauen kann und sich auf den Datenaustausch einlassen darf.

Die Schritte zum Prüfen der Authentizität des Webserver im Einzelnen:

1. Alice muss die öffentlichen Schlüssel aller beteiligten Zertifizierungsstellen kennen, d. h. Alice benötigt die gesamte Zertifikate-Kette zum Prüfen des Webserver-Zertifikats (d. h. des End-Entity-Zertifikats des Webserver):

Üblicherweise hat Alice in ihrem Zertifikatespeicher das benötigte Stammzertifikat. Mit der Installation eines Webbrowsers werden z. B. eine Reihe von vertrauenswürdigen Stammzertifikaten mitinstalliert. Wenn sie das Stammzertifikat nicht hat, muss sie es von der Zertifizierungsstelle herunterladen und im Zertifikatespeicher des Browsers installieren. Die Zertifizierungsstelle kann auch das Gerät sein, auf dem sich der Webserver befindet.

Um die Zwischenzertifikate zu erhalten, gibt es folgende Möglichkeiten:

- Der Server selber sendet zusammen mit seinem End-Entity-Zertifikat die erforderlichen Zwischenzertifikate an Alice, und zwar als signierte Nachricht, damit Alice die Integrität der Zertifikate-Kette prüfen kann.
- In den Zertifikaten befinden sich oft die URLs vom jeweiligen Zertifikate-Aussteller. Über diese URLs kann Alice die erforderlichen Zwischenzertifikate laden.

Wenn Sie Zertifikate in STEP 7 hantieren, wird immer davon ausgegangen, dass Sie die erforderlichen Zwischenzertifikate und das Stammzertifikat in das Projekt importiert und der Baugruppe zugeordnet haben.

2. Mit den öffentlichen Schlüsseln der Zertifikate validiert Alice die Signaturen der Zertifikate-Kette.
3. Der symmetrische Schlüssel muss erzeugt und an den Webserver übermittelt werden.
4. Wenn der Webserver mit seinem Domainnamen adressiert wird, verifiziert Alice gemäß den in RFC 2818 festgelegten PKI-Regeln die Identität des Webserver: Das kann sie, weil die URL des Webserver, in diesem Fall der "Fully Qualified Domain Name" (FQDN), im End-Entity-Zertifikat des Webserver hinterlegt ist. Wenn der Zertifikatseintrag im Feld "Subject Alternative Name" mit dem Eintrag in der Adresszeile des Browsers übereinstimmt, ist alles in Ordnung.

Weiter geht es mit dem Datenaustausch mit dem symmetrischen Schlüssel, wie im Bild oben gezeigt.

3.7 SNMP

3.7.1 SNMP deaktivieren

Das Netzwerk-Management-Protokoll SNMP (Simple Network Management Protocol) ist ein Protokoll, das verschiedene Dienste und Tools zur Erkennung und Diagnose der Netzwerktopologie nutzen.

Welche SNMP-Anfragen die S7-1500 CPUs und die S7-1200 CPUs entgegennehmen können, finden Sie beschrieben in diesem FAQ

(<https://support.industry.siemens.com/cs/ww/de/view/79993228>).

SNMP nutzt das Transportprotokoll UDP. SNMP kennt zwei Netzkomponenten, den SNMP-Manager und den SNMP-Client. Der SNMP-Manager überwacht die Netzwerkknoten. Die SNMP-Clients sammeln in den einzelnen Netzwerkknoten verschiedene netzwerkspezifische Informationen und legen Sie in strukturierter Form in der MIB (Management Information Base) ab. Mit Hilfe dieser Daten können verschiedene Dienste und Tools eine ausführliche Netzwerkd Diagnose durchführen.

Unter bestimmten Voraussetzungen ist es sinnvoll, SNMP zu deaktivieren. Beispiele:

- Die Sicherheitsrichtlinien in Ihrem Netzwerk lassen den Einsatz von SNMP nicht zu.
- Sie verwenden eine eigene SNMP-Lösung, z. B. über eigene Kommunikationanweisungen.

Wenn Sie SNMP für ein Gerät deaktivieren, dann stehen Ihnen verschiedene Möglichkeiten zur Diagnose der Netzwerktopologie (z.B. über das PRONETA-Tool oder über den Webserver der CPU) nicht mehr zur Verfügung.

SNMP deaktivieren

Um SNMP für eine der integrierten Schnittstellen einer S7-1500 CPU zu deaktivieren, gehen Sie folgendermaßen vor:

1. Legen Sie in STEP 7 einem Datenbaustein an, der die Struktur des Datensatzes B071_H enthält.
 - Die folgende Tabelle zeigt die Struktur des Datensatzes B071_H.

Byte	Element	Kodierung	Erläuterung
0-1	BlockID	F003 _H	Header
2-3	BlockLenght	8	Die Datensatzlänge wird ab dem Byte 4 "Version" gezählt.
4	Version	01 _H	
5	Subversion	00 _H	
6-7	Reserviert	-	-
8-11	SNMP-Steuerung	Deaktivieren/Aktivieren von SNMP	Wenn Sie SNMP deaktivieren wollen, dann tragen den Wert 0 ein. Wenn Sie SNMP aktivieren wollen, dann tragen Sie den Wert 1 ein.

2. Übertragen Sie den Datensatz B071_H im Anlauf-OB (OB100) mit der Anweisung WRREC (Datensatz Schreiben) an die CPU.
Nutzen Sie hierzu die Hardware-ID einer integrierten Schnittstelle der CPU.

3.7.2 Beispiel: SNMP für eine CPU 1516-3 PN/DP deaktivieren

Aufgabe

Weil die Sicherheitsrichtlinien in Ihrem Netzwerk kein SNMP zulassen, wollen Sie für eine CPU 1516-3 PN/DP das SNMP deaktivieren.

Voraussetzungen

- CPU 1516-3 PN/DP mit Firmwarestand V2.0
- STEP 7 V14

Lösung

Legen Sie zuerst einen Datenbaustein an, der die Struktur des Datensatzes B071_H enthält. Das folgende Bild zeigt den Datenbaustein "Deactivate SNMP". Der Datenbaustein "Deactivate SNMP" enthält neben dem Datensatz B071_H weitere Variablen, die Sie zum Übertragen des Datensatzes verwenden. Die Variable "snmp_deactivate" dient zum Anstoßen des Auftrags für WRREC. Legen Sie diese Variable in den remanenten Speicherbereich, damit der Wert im Anlauf-OB (OB100) auch verfügbar ist.

Deactivate SNMP									
	Name	Data type	Start value	R...	A...	W...	V...	Comment	
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	snmp_deactivate	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Tag for deactivation	
3	snmp_record	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data record 16#B071	
4	BlockID	UInt	16#F003	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
5	BlockLenght	UInt	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Version	USInt	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Subversion	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Reserved	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
9	SNMPControl	UDInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
10	snmp_done	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
11	snmp_error	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
12	snmp_status	DWord	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

Bild 3-15 Beispiel: Datenbaustein zum Deaktivieren von SNMP

Übertragen Sie den Datensatz B071_H im Anlauf-OB (OB100) mit der Anweisung WRREC (Datensatz Schreiben) an die CPU 1516-3 PN/DP.

Im folgenden Programmcode wird der Datensatz B071_H mit der Anweisung WRREC in einer REPEAT UNTIL Schleife übertragen.

```

ORGANIZATION_BLOCK "Startup"
TITLE = "Complete Restart"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
BEGIN
  REPEAT
    "WRREC_DB_1"
    (REQ := "Deactivate SNMP".snmp_deactivate,
    //Transfer data record
    INDEX:=16#B071,
    //Data record number for SNMP deactivation
    ID:="Local~PROFINET_interface_1",
    //any integrated PROFINET Interface
    DONE => "Deactivate SNMP".snmp_done,
    ERROR => "Deactivate SNMP".snmp_error,
    STATUS => "Deactivate SNMP".snmp_status,
    RECORD := "Deactivate SNMP".snmp_record)
    //Data record
  UNTIL "Deactivate SNMP".snmp_done OR "Deactivate SNMP".snmp_error
  END_REPEAT;
END_ORGANIZATION_BLOCK

```


Programmcode verwenden

Den vollständigen Programmcode finden Sie hier.

Um den Programmcode in Ihr Projekt zu übernehmen, gehen Sie folgendermaßen vor:

1. Kopieren Sie den gesamten Programmcode in die Zwischenablage mit Strg+A, Strg+C.
2. Öffnen Sie einen Texteditor (z. B. "Editor").
3. Fügen Sie den Inhalt der Zwischenablage in den Texteditor ein mit Strg+V.
4. Speichern Sie das Dokument als scl-Datei ab, z. B. SNMP_DEACT.scl.
5. Öffnen Sie Ihr Projekt in STEP 7.
6. Importieren Sie scl-Datei als externe Quelle.
Weitere Informationen zum Importieren von externen Quellen finden Sie in der Onlinehilfe von STEP 7.
7. Erzeugen Sie den Anlauf-OB und die Datenbausteine. (Rechtsklick auf scl-Datei, Kontextmenü: "Baustein aus Quelle generieren")

SNMP wieder aktivieren

Mit kleinen Änderungen können Sie den oben verwendeten Programmcode zum Aktivieren von SNMP verwenden.

Weisen Sie im Anwenderprogramm der Variablen "Deactivate SNMP".snmp_record.SNMPControl den Wert "1" zu:

```
"Deactivate SNMP".snmp_record.SNMPControl := 1;
```

Im nächsten Anlauf der CPU wird SNMP wieder aktiviert.

PG-Kommunikation

Eigenschaften

Über die PG-Kommunikation tauscht die CPU oder ein anderes kommunikationsfähiges Modul Daten mit einer Engineering Station (z. B. PG, PC) aus. Der Datenaustausch ist über PROFIBUS- und PROFINET-Subnetze möglich. Der Übergang zwischen S7-Subnetzen wird ebenfalls unterstützt.

Mit der PG-Kommunikation stehen Ihnen Funktionen zur Verfügung, die Sie zum Laden von Programmen und Konfigurationsdaten, zum Durchführen von Tests und zum Auswerten von Diagnoseinformationen benötigen. Diese Funktionen sind im Betriebssystem des kommunikationsfähigen Moduls integriert.

Ein PG/PC kann mit einer CPU online verbunden sein. Das PG/PC kann maximal 4 Online-Verbindungen (z. B. zu 4 CPUs) parallel betreiben.

Voraussetzungen

- Das PG/PC ist physikalisch mit dem kommunikationsfähigen Modul verbunden.
- Wenn das kommunikationsfähige Modul über S7-Routing erreicht werden soll, muss die Hardware-Konfiguration in die beteiligten Stationen (S7-Router und Endpunkt) geladen sein.

Vorgehen zum online Verbinden

Für die PG-Kommunikation müssen Sie eine Online-Verbindung mit der CPU herstellen:

1. Markieren Sie in STEP 7 in der Projektnavigation die CPU.
2. Wählen Sie den Menübefehl "Online > Online verbinden".

3. Nehmen Sie im Dialog "Online verbinden" die folgenden Einstellungen für Ihre Online-Verbindung vor:
- Wählen Sie in der Klappliste "Typ der PG/PC-Schnittstelle" den Schnittstellentyp (z. B. PN/IE)
 - Wählen Sie in der Klappliste "PG/PC-Schnittstelle" diejenige PG/PC-Schnittstelle (z. B. Ind. Ethernet-Karte), über die Sie die Online-Verbindung herstellen wollen.
 - Wählen Sie in der Klappliste "Verbindung mit Schnittstelle/Subnetz" die Schnittstelle oder das S7-Subnetz aus, mit dem das PG/PC physikalisch verbunden.
 - Falls das kommunikationsfähige Modul über ein S7-Router (Gateway) erreichbar ist, wählen Sie in der Klappliste "1. Gateway" denjenigen S7-Router aus, der die betroffenen Subnetze miteinander verbindet.

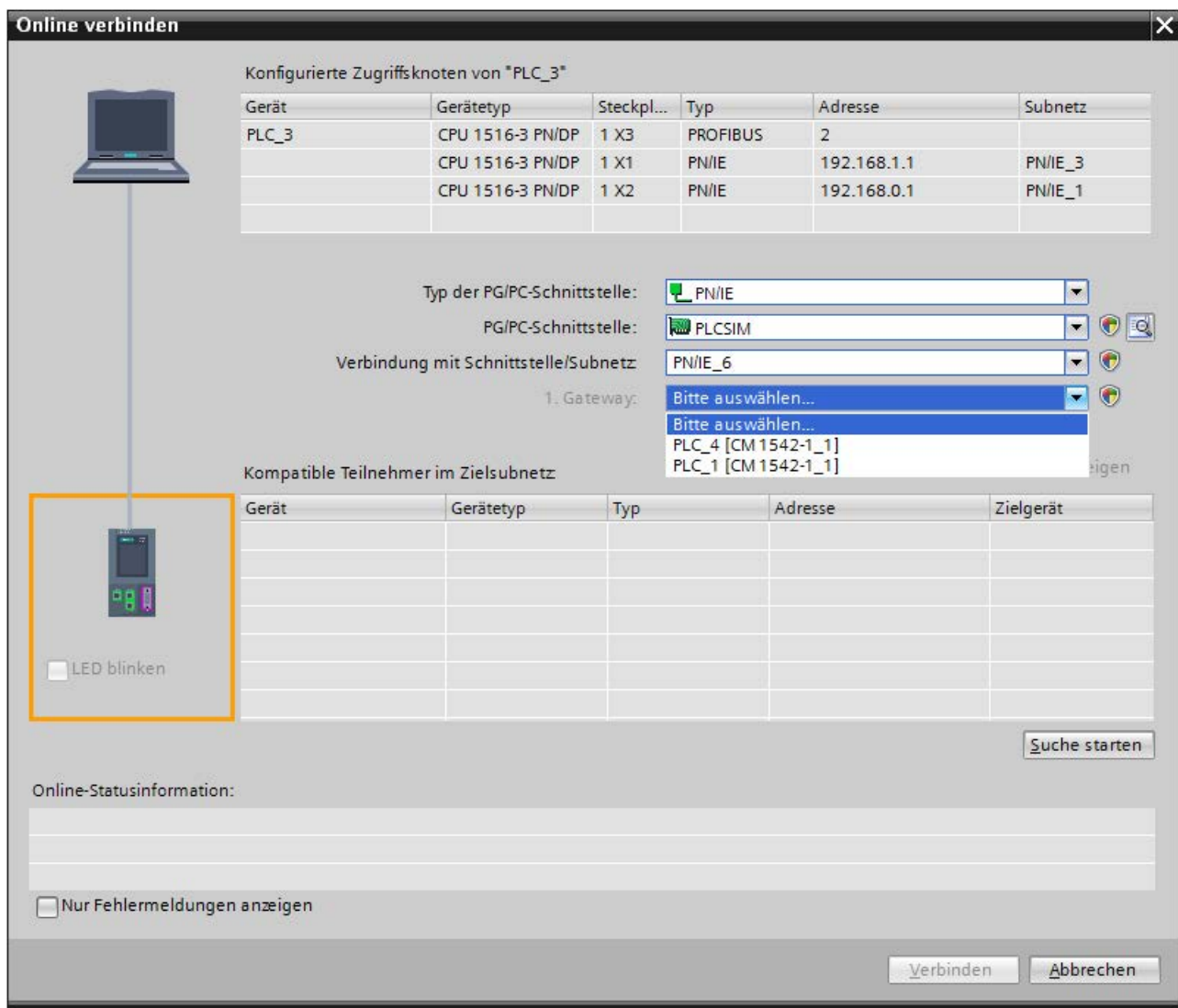


Bild 4-1 PG-Kommunikation einrichten

4. Klicken Sie auf "Suche starten".
Nach kurzer Zeit erscheinen in der Tabelle "Kompatible Teilnehmer im Zielsubnetz" alle Geräte, die Sie mit PG-Kommunikation ansprechen können.
5. Wählen Sie in der Tabelle "Kompatible Teilnehmer im Zielsubnetz" die entsprechende CPU aus und bestätigen Sie mit "Verbinden".

Weitere Informationen

Weitere Informationen zum "Online verbinden" finden Sie in der Online-Hilfe STEP 7.

HMI-Kommunikation

Eigenschaften

Über die HMI-Kommunikation tauschen ein oder mehrere HMI-Geräte (z. B. HMI Basic/Comfort/Mobile Panel) Daten zum Bedienen und Beobachten mit einer CPU über die PROFINET- oder PROFIBUS DP-Schnittstelle aus. Der Datenaustausch erfolgt über HMI-Verbindungen.

Wenn Sie mehrere HMI-Verbindungen zu einer CPU einrichten möchten, verwenden Sie z. B.:

- die PROFINET- und PROFIBUS DP-Schnittstellen der CPU
- CPs und CMs mit den entsprechenden Schnittstellen

Vorgehen zum Einrichten von HMI-Kommunikation

Sobald Sie eine Variable, z. B. eine Variable aus einem globalen Datenbaustein, per Drag & Drop in ein HMI-Bild oder in die HMI-Variablentabelle hineinziehen, richtet STEP 7 automatisch eine HMI-Verbindung ein. Alternativ können Sie die HMI-Verbindung auch selbst einrichten.

Um eine HMI-Verbindung einzurichten, gehen Sie folgendermaßen vor.

1. Konfigurieren Sie in der Netzsicht des Hardware- und Netzwerkeditors von STEP 7 das HMI-Gerät in einer vorhandenen Konfiguration mit der CPU.
2. Wählen Sie die Schaltfläche "Verbindungen" und aus der Klappliste "HMI-Verbindung".
3. Ziehen Sie per Drag & Drop eine Linie zwischen den Endpunkten der Verbindung (HMI-Gerät und CPU). Die Endpunkte sind farblich hervorgehoben. Falls noch kein zugehöriges S7-Subnetz vorhanden ist, wird dieses automatisch angelegt.

4. Wählen Sie im Register "Verbindungen" die Zeile der HMI-Verbindung.

Im Bereich "Allgemein", im Register "Eigenschaften" sehen Sie die Eigenschaften der HMI-Verbindung, die Sie z. T. ändern können.

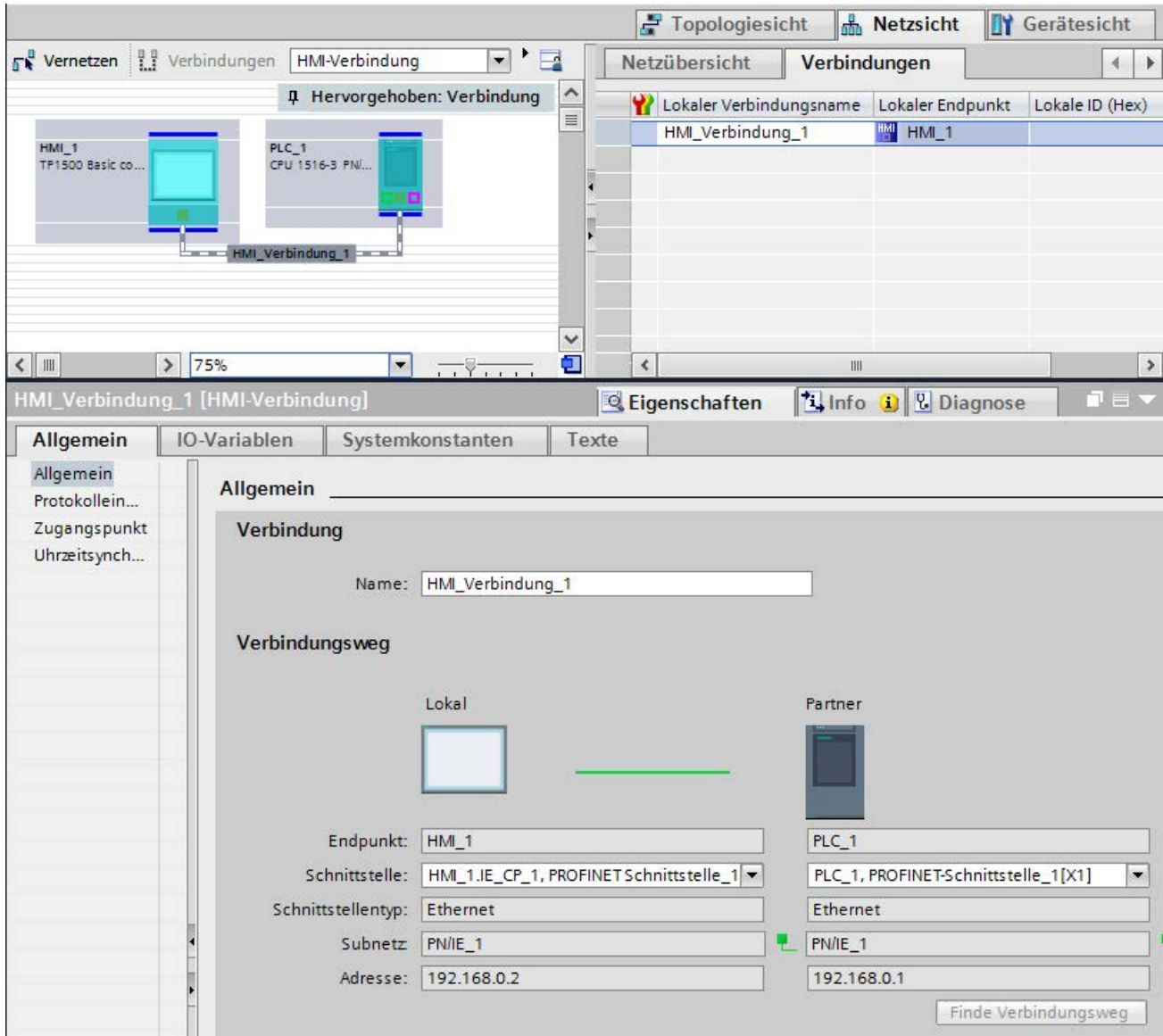


Bild 5-1 HMI-Kommunikation einrichten

5. Laden Sie die Hardware-Konfiguration in die CPU.
6. Laden Sie die Hardware-Konfiguration in das HMI-Gerät.

Weitere Informationen

Informationen zum S7-Routing für HMI-Verbindungen finden Sie im Kapitel S7-Routing (Seite 235).

Weitere Informationen zum Einrichten von HMI-Verbindungen finden Sie in der Online-Hilfe STEP 7.

Open User Communication

6.1 Open User Communication im Überblick

Merkmale von Open User Communication

Über Open User Communication, auch "Offene Kommunikation" genannt, tauscht die CPU-Daten mit einem weiteren, kommunikationsfähigen Gerät aus. Die Open User Communication zeichnet sich durch folgende Merkmale aus:

- Offener Standard (Kommunikationspartner können zwei SIMATIC CPUs oder eine SIMATIC CPU und ein geeignetes Fremdgerät sein)
- Kommunikation über unterschiedliche Protokolle (in STEP 7 als "Verbindungstypen" bezeichnet)
- Hohe Flexibilität hinsichtlich der zu übertragenden Datenstrukturen; ermöglicht damit offenen Datenaustausch mit beliebigen Kommunikationsteilnehmern, sofern diese die zur Verfügung gestellten Verbindungstypen unterstützen
- Secure Communication: Zum Schutz Ihres Automatisierungssystems können Sie Daten über Open User Communication gesichert austauschen. Bei der Secure Open User Communication werden die signiert und verschlüsselt gesendet.
- Open User Communication ist in verschiedenen Automatisierungssystemen möglich, siehe technische Daten der jeweiligen Gerätehandbücher.
Beispiele:
 - Integrierte PROFINET/Ind. Ethernet-Schnittstellen von CPUs (S7-1500, ET 200SP CPU, S7-1500 Software Controller, CPU 1516pro-2 PN)
 - PROFINET/Ind. Ethernet-Schnittstellen von Kommunikationsmodulen (z. B. CP 1543-1, CM 1542-1)

Informationen zu Secure Communication finden Sie im Kapitel Secure Communication (Seite 38).

6.2 Protokolle für Open User Communication

Protokolle für Open User Communication

Für die offene Kommunikation stehen folgende Protokolle zur Verfügung:

Tabelle 6- 1 Transportprotokolle für offene Kommunikation

Transportprotokoll	Über Schnittstelle
TCP gemäß RFC 793	PROFINET/Industrial Ethernet
ISO-on-TCP gemäß RFC 1006 (Class 4)	PROFINET/Industrial Ethernet
ISO gemäß ISO/IEC 8073	Industrial Ethernet (nur CP 1543-1)
UDP gemäß RFC 768	PROFINET/Industrial Ethernet
FDL	PROFIBUS

Tabelle 6- 2 Applikationsprotokolle für offene Kommunikation

Applikationsprotokoll	Genutztes Transportprotokoll
Modbus TCP	TCP gemäß RFC 793
E-Mail	TCP gemäß RFC 793
FTP	TCP gemäß RFC 793

TCP, ISO-on-TCP, ISO, UDP

Diese Protokolle (außer UDP) bauen vor der Datenübertragung eine Transportverbindung zum Kommunikationspartner auf. Verbindungsorientierte Protokolle werden eingesetzt, wenn es bei der Datenübertragung besonders auf Sicherheit vor Datenverlust ankommt.

Bei UDP ist möglich:

- Unicast an einen oder Broadcast an alle Teilnehmer am PROFINET über die PROFINET-Schnittstelle der CPU oder die Industrial Ethernet-Schnittstelle des CP 1543-1
- Multicast an alle Empfänger eines Multicast-Kreises über die PROFINET-Schnittstelle der CPU* oder die PROFINET/Industrial Ethernet-Schnittstelle des CP 1543-1

* ab Firmwarestand V2.0, die PROFINET-Schnittstelle der CPU unterstützt maximal 5 Multicast-Kreise

Maximale Nutzdatenlängen UDP: Welche maximale Nutzdatenlänge für UDP unterstützt wird, finden Sie in den Technischen Daten der jeweiligen Gerätehandbücher beschrieben.

Protokoll zur Kommunikation via PROFIBUS: FDL

Die Datenübertragung über eine FDL-Verbindung (Fieldbus Data Link) ist geeignet für die Übertragung zusammenhängender Datenblöcke zu einem Kommunikationspartner am PROFIBUS, der das Senden bzw. Empfangen entsprechend des FDL-Dienstes SDA (Send Data with Acknowledge) nach EN 50170, Vol 2. unterstützt. Beide Partner sind gleichberechtigt, d. h. jeder Partner kann ereignisabhängig den Sende- und Empfangsvorgang anstoßen.

Entsprechend des FDL-Dienstes SDN (Send Data with No Acknowledge) nach EN 50170, Vol 2. sind bei FDL möglich:

- Broadcast an alle Teilnehmer am PROFIBUS über die PROFIBUS-Schnittstelle des CM 1542-5
- Multicast an alle Empfänger eines Multicast-Kreises über die PROFIBUS-Schnittstelle des CM 1542-5

Modbus TCP

Das Modbus-Protokoll ist ein Kommunikationsprotokoll mit Linientopologie auf Basis einer Master/Slave-Architektur. In der Übertragungsart Modbus TCP (Transmission Control Protocol) werden die Daten als TCP/IP-Pakete übertragen.

Die Kommunikation wird ausschließlich über entsprechende Anweisungen im Anwenderprogramm gesteuert.

E-Mail und FTP

Über E-Mail ist z. B. das Versenden von Datenbausteininhalten (z. B. Prozessdaten) als Anhang möglich.

Die FTP-Verbindung (FTP = File-Transfer-Funktionen) verwenden Sie für die Übertragung von Dateien zu und von den S7-Geräten.

Client-seitig wird die Kommunikation über entsprechende Anweisungen im Anwenderprogramm gesteuert.

Anwendungsbeispiel: MQTT Publisher für die SIMATIC S7-1500 CPU

Das "Message Queue Telemetry Transport" (MQTT) ist ein einfaches Protokoll auf TCP/IP-Ebene. Es eignet sich für den Nachrichtenaustausch zwischen Geräten mit geringer Funktionalität und für die Übertragung über unzuverlässige Netze.

Das Anwendungsbeispiel stellt Ihnen einen Funktionsbaustein zur Verfügung, mit dem Sie das MQTT-Protokoll in die SIMATIC S7-1500 implementieren können.

Das Anwendungsbeispiel finden Sie im Internet (<https://support.industry.siemens.com/cs/ww/de/view/109748872>).

6.3 Anweisungen für Open User Communication

Einleitung

Sie richten die Open User Communication über die entsprechende Verbindung (z. B. TCP-Verbindung) wie folgt ein:

- durch Programmieren in den Anwenderprogrammen der Kommunikationspartner oder
- durch Projektieren der Verbindung in STEP 7 im Hardware- und Netzwerkkeditor

Unabhängig vom Einrichten der Verbindung durch Programmierung oder Projektierung sind in den Anwenderprogrammen beider Kommunikationspartner immer Anweisungen zum Senden und Empfangen der Daten notwendig.

Einrichten der Verbindung über das Anwenderprogramm

Beim programmierten Einrichten der Verbindung wird der Verbindungsauf- und -abbau über Anweisungen im Anwenderprogramm realisiert.

In bestimmten Anwendungsbereichen ist es vorteilhaft, die Kommunikationsverbindungen nicht statisch einzurichten, mittels Projektierung in der Hardware-Konfiguration, sondern über das Anwenderprogramm. Sie können die Verbindungen über eine spezifische Applikation programmgesteuert und damit bei Bedarf einrichten. Das programmierte Einrichten ermöglicht außerdem die Freigabe von Verbindungsressourcen nach der Datenübertragung.

Für jede Kommunikationsverbindung ist eine Datenstruktur notwendig, die die Parameter für den Aufbau der Verbindung enthält (z. B. Systemdatentyp "TCON_IP_v4" für TCP).

Die Systemdatentypen (SDT) werden vom System zur Verfügung gestellt und haben eine vordefinierte Struktur, die nicht änderbar ist.

Die verschiedenen Protokolle haben jeweils eigene Datenstrukturen (siehe folgende Tabelle). Die Parameter werden in einem Datenbaustein ("Verbindungsbeschreibungs-DB") z. B. des Systemdatentyps TCON_IP_v4 gespeichert.

Sie haben zwei Möglichkeiten, den DB mit der Datenstruktur vorzugeben:

- Empfehlung: Datenbaustein bei der Parametrierung der Verbindung in den Eigenschaften im Programmeditor automatisch anlegen lassen, mit Hilfe der Verbindungsparametrierung bei den Anweisungen TSEND_C, TRCV_C und TCON
- Datenbaustein manuell erstellen, parametrieren und direkt an die Anweisung schreiben
Notwendig für:
 - Secure OUC
 - Verbindung über DNS
 - E-Mail
 - FTP

Im "Verbindungsbeschreibungs-DB" können Sie die Verbindungsparameter modifizieren.

Wie Sie die Anweisung TCON programmieren, um zwischen zwei S7-1500 CPUs eine Verbindung für die Open User Communication einzurichten, finden Sie in diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/58875807>).

Protokolle, Systemdatentypen und einsetzbare Anweisungen für programmiertes Einrichten

Die folgende Tabelle zeigt Ihnen die Protokolle der Open User Communication und die dazu passenden Systemdatentypen und Anweisungen.

Tabelle 6- 3 Anweisungen bei programmiertem Einrichten der Verbindung

Protokoll	Systemdatentyp	Anweisungen
TCP	<ul style="list-style-type: none"> • TCON_QDN • TCON_IP_v4 	Verbindung herstellen und Daten senden/empfangen über:
ISO-on-TCP	<ul style="list-style-type: none"> • TCON_IP_RFC 	<ul style="list-style-type: none"> • TSEND_C/TRCV_C oder • TCON, TSEND/TRCV oder
ISO gemäß ISO/IEC 8073 (Class 4)	<ul style="list-style-type: none"> • TCON_ISOnative¹ • TCON_Configured 	<ul style="list-style-type: none"> • TCON, TUSEND/TURCV (Abbau der Verbindung über TDISCON möglich)
UDP	<ul style="list-style-type: none"> • TCON_IP_v4 • TADDR_Param • TADDR_SEND_QDN • TADDR_RCV_IP 	Verbindung herstellen und Daten senden/empfangen über: <ul style="list-style-type: none"> • TSEND_C/TRCV_C • TUSEND/TURCV/TRCV (Abbau der Verbindung über TDISCON möglich)
FDL ¹	<ul style="list-style-type: none"> • TCON_FDL 	Verbindung herstellen und Daten senden/empfangen über: <ul style="list-style-type: none"> • TSEND_C/TRCV_C oder • TCON, TSEND/TRCV oder • TCON, TUSEND/TURCV (Abbau der Verbindung über TDISCON möglich)
Modbus TCP	<ul style="list-style-type: none"> • TCON_IP_v4 • TCON_QDN 	<ul style="list-style-type: none"> • MB_CLIENT • MB_SERVER
E-Mail	<ul style="list-style-type: none"> • TMAIL_v4 • TMAIL_v6 • TMAIL_FQDN 	<ul style="list-style-type: none"> • TMAIL_C
FTP ²	<ul style="list-style-type: none"> • FTP_CONNECT_IPV4³ • FTP_CONNECT_IPV6³ • FTP_CONNECT_NAME³ 	<ul style="list-style-type: none"> • FTP_CMD

¹ dieses Protokoll ist nur über das CM 1542-5 verwendbar

² dieses Protokoll ist nur über den CP 1543-1 verwendbar

³ anwenderdefinierter Datentyp

Die folgende Tabelle zeigt Ihnen die verschiedenen Verbindungen der Secure Open User Communication und die dazu passenden Systemdatentypen und Anweisungen.

Secure OUC-Verbindung	Systemdatentyp	Anweisungen
Gesicherte TCP-Verbindung von einer S7-1500 CPU als TLS-Client zu einem Fremd-PLC (TLS-Server)	• TCON_QDN_SEC	• TSEND_C/TRCV_C • TCON
Gesicherte TCP-Verbindung von einer S7-1500 CPU als TLS-Server zu einem Fremd-PLC (TLS-Client)		
Gesicherte TCP-Verbindung zwischen zwei S7-1500 Stationen	• TCON_IP_V4_SEC ¹	
Gesicherte Verbindung zu einem Mailserver ²	• TMAIL_V4_SEC • TMAIL_QDN_SEC	• TMAIL_C (ab V5.0)
Gesicherte Modbus TCP-Verbindung	• TCON_IP_V4_SEC ¹	• MB_Client • MB_Server
	• TCON_QDN_SEC	

¹ Auch über CP 1543-1 möglich

² Gesicherte Verbindung zu einem Mailserver auch möglich mit CP1543-1 und TMAIL_C (V4.0)

Einrichten der Verbindung über Verbindungsprojektierung

Beim Einrichten über die Verbindungsprojektierung werden die Adressparameter der Verbindung im Hardware- und Netzwerkeitor von STEP 7 festgelegt.

Für das Senden und Empfangen der Daten nutzen Sie die gleichen Anweisungen, wie beim programmierten Einrichten von Verbindungen:

Tabelle 6- 4 Anweisungen zum Senden/Empfangen bei projektierten Verbindungen

Protokoll	Senden/Empfangen bei projektierten Verbindungen
Einsetzbare Anweisungen:	
TCP	Daten senden/empfangen über: • TSEND_C/TRCV_C oder • TSEND/TRCV oder • TUSEND/TURCV
ISO-on-TCP	
ISO gemäß ISO/IEC 8073 (Class 4)	
UDP	Daten senden/empfangen über: • TSEND_C/TRCV_C oder • TUSEND/TURCV
FDL	Daten senden/empfangen über: • TSEND_C/TRCV_C oder • TSEND/TRCV oder • TUSEND/TURCV
Modbus TCP	Nicht unterstützt
E-Mail	Nicht unterstützt
FTP	Nicht unterstützt

Weitere Anweisungen für offene Kommunikation

Die folgenden Anweisungen können Sie sowohl bei im Anwenderprogramm eingerichtet Verbindungen, als auch bei projektierten Verbindungen einsetzen:

- T_RESET: Verbindung abbauen und aufbauen
- T_DIAG: Verbindung überprüfen

Basisbeispiele für Open User Communication

Der Siemens Online Support bietet Ihnen Funktionsbausteine (FBs) an, die Ihnen die Handhabung der Anweisungen der Open User Communication erleichtern. Die Funktionsbausteine mit zugehörigen Beispielen finden Sie im Internet (<https://support.industry.siemens.com/cs/ww/de/view/109747710>).

Weitere Informationen

In der Online-Hilfe STEP 7 finden Sie beschrieben:

- die Anwender- und Systemdatentypen
- die Anweisungen für offene Kommunikation
- die Verbindungsparameter

Informationen zur Belegung und Freigabe von Verbindungsressourcen finden Sie im Kapitel Belegung von Verbindungsressourcen (Seite 246).

Siehe auch

Secure Open User Communication (Seite 98)

6.4 Open User Communication mit Adressierung über Domainnamen

S7-1500 CPUs, ET 200SP CPUs und die CPU 1516pro-2 PN unterstützen ab Firmwarestand V2.0 die Open User Communication mit Adressierung über ein Domain Name System (DNS). In der CPU ist ein DNS-Client integriert. Bei der Kommunikation über DNS verwenden Sie Domainnamen als Alias für IP-Adressen zur Adressierung von Kommunikationspartnern. Die Adressierung der Kommunikationspartner über Domainnamen ist für offene Kommunikation über TCP und UDP möglich.

Als Voraussetzung für die Kommunikation über DNS muss sich in Ihrem Netz mindestens ein DNS-Server befinden.

Der S7-1500 Software Controller unterstützt Kommunikation über DNS für alle Schnittstellen, die dem Software-Controller zugeordnet sind.

Kommunikation über DNS einrichten

Damit eine CPU eine Verbindung zu einem Kommunikationspartner über dessen Domainnamen aufbauen kann, muss der DNS-Client der CPU die IPv4-Adresse von mindestens einem DNS-Server kennen. Die CPU unterstützt bis zu 4 verschiedene DNS-Server.

Um die Kommunikation über Domainnamen für eine S7-1500 CPU einzurichten, gehen Sie folgendermaßen vor:

1. Selektieren Sie die CPU in der Netzansicht von STEP 7.
2. Navigieren Sie im Inspektorenfenster zu "Eigenschaften" > "Allgemein" > "DNS-Konfiguration".
3. Tragen Sie in der Tabelle "Serverliste" in der Spalte "DNS-Serveradressen" die IPv4-Adresse von einem DNS-Server ein.
Sie können bis zu 4 IPv4-Adressen von DNS-Servern eintragen.

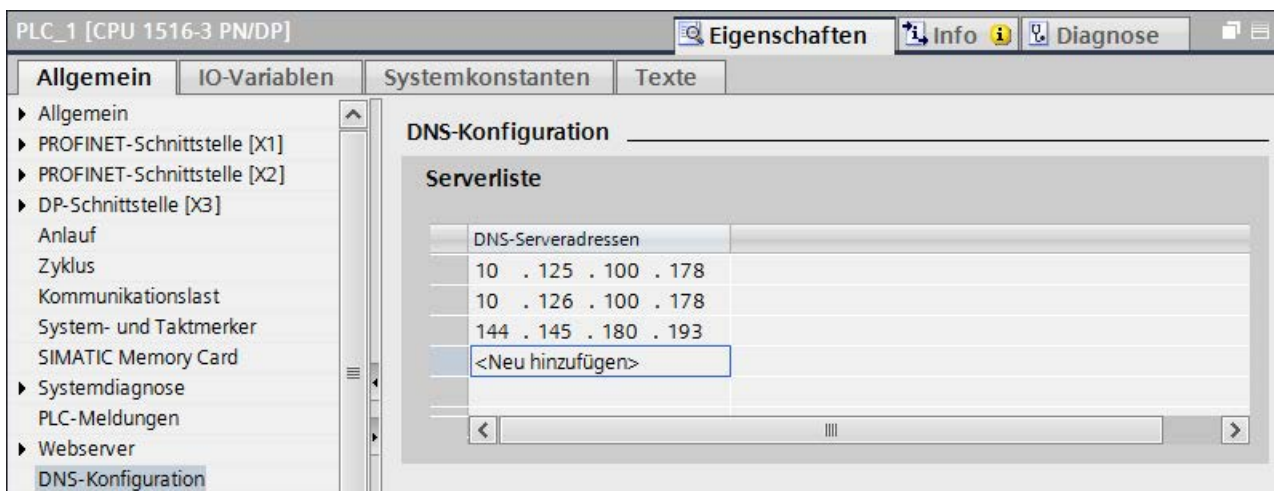


Bild 6-1 DNS-Serveradressen eintragen am Beispiel einer CPU 1516-3 PN/DP

TCP-Verbindung über den Domainnamen des Kommunikationspartners einrichten

Für die TCP-Kommunikation über den Domainnamen müssen Sie selbst einen Datenbaustein mit dem Systemdatentyp TCON_QDN erstellen, parametrieren und direkt an der Anweisung aufrufen. Die Anweisungen TCON, TSEND_C und TRCV_C unterstützen den Systemdatentyp TCON_QDN:

Um eine TCP-Verbindung über den Domainnamen des Kommunikationspartners einzurichten, gehen Sie folgendermaßen vor:

- Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
- Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_QDN.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "DNS Connection1" vom Datentyp TCON_QDN definiert ist.

Data_block_1				
	Name	Datentyp	Startwert	Kommentar
1	Static			
2	DNS Connection1	TCON_QDN		
3	Interfaceld	HW_ANY	0	not relevant
4	ID	CONN_OUC	16#0	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connection: 16#0B=11=TCP/IP, 16#13=19=UDP
6	ActiveEstablished	Bool	false	active/passive connection establishment
7	RemoteQDN	String[254]	"	fully or partially qualified domain name of remote partner
8	RemotePort	UInt	0	remote UDP / TCP port number
9	LocalPort	UInt	0	local UDP / TCP port number

Bild 6-2 Datentyp TCON_QDN

- Programmieren Sie die Parameter der TCP-Verbindung (z. B. den vollqualifizierten Domainnamen (FQDN)) in der Variablen vom Datentyp TCON_QDN.
- Legen Sie im Programmeditor eine Anweisung TCON an.
- Verschalten Sie den Parameter CONNECT der Anweisung TCON mit der Variable vom Datentyp TCON_QDN.

Im folgenden Beispiel ist der Parameter CONNECT der Anweisung TCON mit der Variablen "DNS connection1" (Datentyp TCON_QDN) verschaltet.

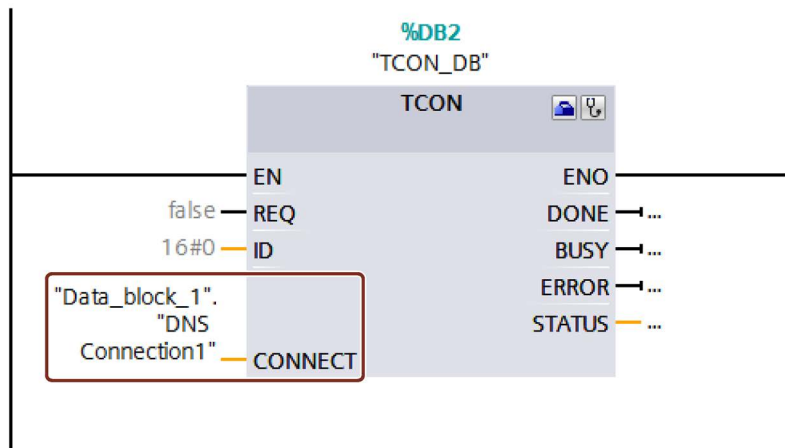


Bild 6-3 Anweisung TCON

UDP-Verbindung über den Domainnamen des Kommunikationspartners adressieren

Beim Senden von Daten über UDP können Sie für S7-1500-CPU's ab Firmware-Version V2.0 den Empfänger mit seinem voll qualifizierten Domainnamen (FQDN) adressieren. Dabei verweisen Sie bei der Anweisung TUSEND am Parameter ADDR auf eine Struktur vom Typ TADDR_SEND_QDN.

Der Empfänger kann eine IPv4- oder eine IPv6-Adresse zurückliefern. Verweisen Sie bei der Anweisung TURCV am Parameter ADDR daher auf eine Struktur vom Typ TADDR_RCV_IP. Nur diese kann beide IP-Adresstypen aufnehmen.

Hinweis

Netzlast

Im Gegensatz zu TCP arbeitet das Protokoll UDP nicht verbindungsorientiert. Bei jeder Flanke am Bausteinparameter REQ führt die Anweisung TUSEND bzw. TURCV Abfrage des DNS-Servers durch. Dies kann zu hoher Netzwerklast bzw. Last auf dem DNS-Server führen.

Weitere Information

Weitere Informationen zu den Systemdatentypen TCON_QDN, TADDR_SEND_QDN und TADDR_RCV_IP finden Sie in der Onlinehilfe zu STEP 7.

Wie Sie eine gesicherte TCP-Verbindung über den Domainnamen des Kommunikationspartners einrichten, finden Sie im Kapitel Secure Open User Communication (Seite 98).

6.5 Open User Communication über TCP, ISO-on-TCP, UDP und ISO einrichten

Verbindung für die Anweisungen TSEND_C, TRCV_C oder TCON parametrieren

Voraussetzung: Im Programmiereditor ist eine Anweisung TSEND_C, TRCV_C oder TCON angelegt.

1. Selektieren Sie im Programmiereditor einen Baustein der Open User Communication TCON, TSEND_C oder TRCV_C.
2. Öffnen Sie im Inspektorfenster das Register "Eigenschaften > Konfiguration".

3. Selektieren Sie die Gruppe "Verbindungsparameter". Solange Sie noch keinen Verbindungspartner selektiert haben, ist nur die leere Klappliste für den Partner-Endpunkt aktiv. Alle anderen Eingabemöglichkeiten sind deaktiviert.

Es werden die bereits bekannten Verbindungsparameter angezeigt:

- Name des lokalen Endpunkts
- Schnittstelle des lokalen Endpunkts
- IPv4-Adresse des lokalen Endpunkts

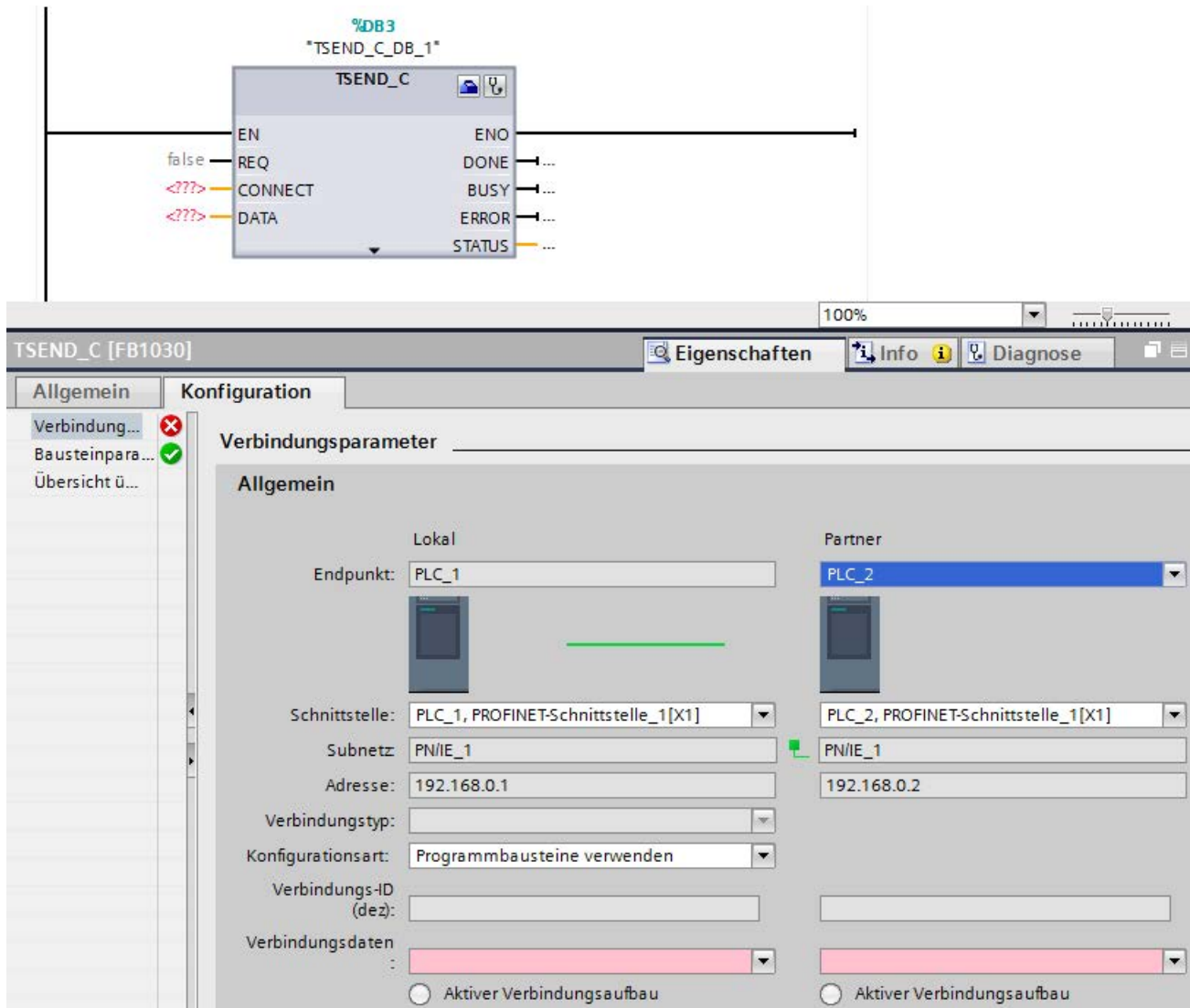


Bild 6-4 Verbindungsparametrierung für TSEND_C

4. Wählen Sie in der Klappliste des Partner-Endpunkts einen Verbindungspartner. Als Kommunikationspartner kommt ein unspezifiziertes Gerät oder eine im Projekt vorhandene CPU in Frage. Bestimmte Verbindungsparameter werden danach als Vorgabe automatisch eingetragen.

Die folgenden Parameter werden eingestellt:

- Name des Partner-Endpunkts
- Schnittstelle des Partner-Endpunkts
- IPv4-Adresse des Partner-Endpunkts

Wenn die Verbindungspartner vernetzt sind, wird der Name des Subnetzes angezeigt.

5. Wählen Sie in der Klappliste "Konfigurationsart" zwischen der Verwendung von Programmbausteinen oder konfigurierten Verbindungen.

6. Wählen Sie in der Klappliste "Verbindungsdaten" einen vorhandene Verbindungsbeschreibungs-DBs oder bei konfigurierten Verbindungen unter "Verbindungsname" eine vorhandene Verbindung. Sie können auch einen neue Verbindungsbeschreibungs-DBs oder eine neue konfigurierte Verbindung anlegen. Sie können später noch andere Verbindungsbeschreibungs-DBs oder konfigurierte Verbindungen wählen oder die Namen der Verbindungsbeschreibungs-DBs ändern, um neue Datenbausteine zu erstellen:
- Den ausgewählten Datenbaustein sehen Sie auch an der Beschaltung des Eingangsparameters CONNECT der ausgewählten Anweisung TCON, TSEND_C oder TRCV_C.
 - Wenn Sie für den Verbindungspartner bereits einen Verbindungsbeschreibungs-DB über den Parameter CONNECT der Anweisung TCON, TSEND_C oder TRCV_C angegeben haben, können Sie entweder diesen DB verwenden oder einen neuen DB anlegen.
 - Wenn Sie den Namen des angezeigten Datenbausteins in der Klappliste bearbeiten, wird automatisch ein neuer Datenbaustein mit dem geänderten Namen, aber derselben Struktur und demselben Inhalt generiert und für die Verbindung verwendet.
 - Geänderte Namen eines Datenbausteins müssen im Kontext des Kommunikationspartners eindeutig sein.
 - Ein Verbindungsbeschreibungs-DB muss je nach CPU-Typ und Verbindung die Struktur TCON_Param, TCON_IP_v4 oder TCON_IP_RFC haben.
 - Ein Datenbaustein kann nicht für einen unspezifizierten Partner ausgewählt werden.

Nach Auswahl oder Anlegen des Verbindungsbeschreibungs-DBs oder der konfigurierten Verbindung werden weitere Werte ermittelt und eingetragen.

Für spezifizierte Verbindungspartner gilt:

- Verbindungstyp ISO-on-TCP
- Verbindungs-ID mit dem Vorgabewert 1
- Aktiver Verbindungsaufbau vom lokalen Partner
- TSAP-ID
für S7-1200/1500: E0.01.49.53.4F.6F.6E.54.43.50.2D.31

Für unspezifizierte Verbindungspartner gilt:

- Verbindungstyp TCP
- Partnerport 2000

Bei konfigurierter Verbindung mit spezifiziertem Verbindungspartner gilt:

- Verbindungstyp TCP
- Verbindungs-ID mit dem Vorgabewert 257
- Aktiver Verbindungsaufbau vom lokalen Partner
- Partnerport 2000

Bei konfigurierter Verbindung mit unspezifiziertem Verbindungspartner gilt:

- Verbindungstyp TCP
- Lokaler Port 2000

7. Geben Sie ggf. eine Verbindungs-ID für den Verbindungspartner an. Für einen unspezifizierten Partner kann keine Verbindungs-ID vergeben werden.

Hinweis

Sie müssen bei einem bekannten Verbindungspartner einen eindeutigen Wert für die Verbindungs-ID eingeben. Die Eindeutigkeit der Verbindungs-ID wird nicht durch die Verbindungsparametrierung geprüft und es wird bei Anlegen einer neuen Verbindung kein Vorgabewert für die Verbindungs-ID eingetragen!

8. Wählen Sie den gewünschten Verbindungstyp aus der entsprechenden Klappliste. Die Adressdetails werden abhängig vom Verbindungstyp mit Werten vorbelegt. Sie haben die Wahl zwischen:

- TCP
- ISO-on-TCP
- UDP
- ISO (nur bei Konfigurationsart "Konfigurierte Verbindung verwenden")

Sie können die Eingabefelder in den Adressdetails bearbeiten. Je nach eingestelltem Protokoll können Sie die Ports (für TCP und UDP) oder die TSAPs (für ISO-on-TCP und ISO) bearbeiten.

9. Stellen Sie bei TCP, ISO und ISO-on-TCP das Verhalten für den Verbindungsaufbau über die Optionsfelder "Aktiver Verbindungsaufbau" ein. Sie können auswählen, welcher Kommunikationspartner die Verbindung aktiv aufbauen soll.

Geänderte Werte werden von der Verbindungsparametrierung sofort auf Eingabefehler geprüft und in den Datenbaustein für die Verbindungsbeschreibung eingetragen.

Hinweis

Die Open User Communication zwischen zwei Kommunikationspartnern ist erst dann lauffähig, wenn auch der Programmteil für den Partner-Endpunkt in die Hardware geladen wurde. Achten Sie darauf, dass Sie für eine funktionierende Kommunikation nicht nur die Verbindungsbeschreibung der lokalen CPU in das Gerät laden, sondern auch die der Partner-CPU.

Verbindungen, z. B. für TSEND/TRCV, projektieren

Wenn Sie z. B. die Anweisungen für TSEND/TRCV für offene Kommunikation nutzen wollen, müssen Sie zunächst eine Verbindung (z. B. TCP-Verbindung) projektieren.

Um eine TCP-Verbindung zu projektieren, gehen Sie folgendermaßen vor:

1. Konfigurieren Sie in der Netzsicht des Hardware- und Netzwerkeditors von STEP 7 die Kommunikationspartner.
2. Wählen Sie die Schaltfläche "Verbindungen" und aus der Klappliste den Verbindungstyp "TCP-Verbindung".
3. Verbinden Sie per Drag & Drop die Kommunikationspartner miteinander (über Schnittstelle oder lokalen Endpunkt). Falls noch kein zugehöriges S7-Subnetz vorhanden ist, wird dieses automatisch angelegt.

Alternativ können Sie auch eine Verbindung zu unspezifizierten Partnern einrichten.

4. Selektieren Sie die angelegte Verbindung in der Netzsicht.
5. Stellen Sie im Bereich "Allgemein", im Register "Eigenschaften" ggf. die Eigenschaften der Verbindung ein, z. B. den Namen der Verbindung und die verwendeten Schnittstellen der Kommunikationspartner.

Für Verbindungen zu einem unspezifizierten Partner stellen Sie die Adresse des Partners ein.

Im Bereich "Lokale ID" finden Sie die lokale ID (Referenz der Verbindung im Anwenderprogramm).

6. Wählen Sie in der Projektnavigation für eine der beiden CPUs den Ordner "Programmbausteine" und öffnen Sie im Ordner den OB 1 durch Doppelklick. Der Programmeditor öffnet sich.
7. Wählen Sie aus der Task Card "Anweisungen", Bereich "Kommunikation", "Open User Communication" die gewünschte Anweisung, z. B. TSEND und ziehen Sie sie per Drag & Drop in ein Netzwerk des OB 1.
8. Vergeben Sie am Parameter ID der Anweisung die lokale ID der projektierten Verbindung, die für die Übertragung der Daten verwendet werden soll.
9. Verschalten Sie den Parameter "DATA" an der Anweisung TSEND mit den Anwenderdaten, z. B. in einem Datenbaustein.
10. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Richten Sie nach der oben beschriebenen Vorgehensweise die Verbindung in der Partner-CPU mit der Anweisung zum Empfangen, TRCV ein und laden Sie sie in die CPU.

Besonderheit bei ISO-Verbindungen mit CP 1543-1

Wenn Sie den Verbindungstyp "ISO-Verbindung" nutzen, müssen Sie das Kontrollkästchen "ISO-Protokoll verwenden" in den Eigenschaften des CPs aktivieren, damit die Adressierung über MAC-Adressen funktioniert.

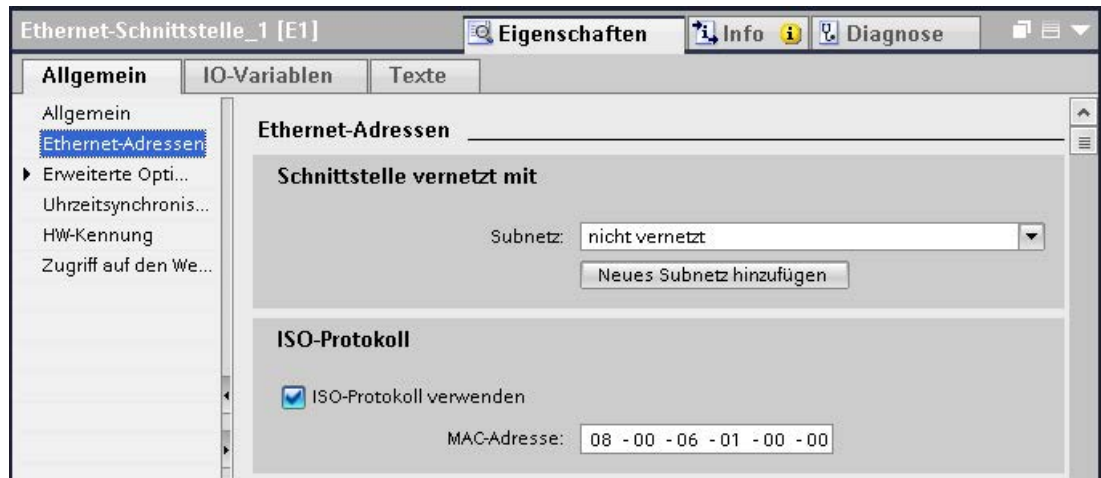


Bild 6-5 CP 1543-1 ISO-Protokoll wählen

Weitere Informationen

In der Online-Hilfe STEP 7 finden Sie beschrieben:

- Die Anweisungen für offene Kommunikation
- Die Verbindungsparameter

Wie sich die Anweisungen TSEND_C und TRCV_C in der S7-1500 verhalten, finden Sie in diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/109479564>).

6.6 Kommunikation über FDL einrichten

Voraussetzung

- Projektierungs-Software: STEP 7 Professional V14
- Endpunkt der Verbindung: CPU S7-1500 ab Firmware-Version V2.0 mit dem Kommunikationsmodul CM 1542-5 mit Firmware-Version V2.0

Einrichten einer konfigurierten FDL-Verbindung

Um in STEP 7 eine konfigurierte FDL-Verbindung einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie im Programmeditor eine Anweisung TSEND_C an.
2. Selektieren Sie die Anweisung TSEND_C und navigieren Sie im Inspektorfenster zu "Eigenschaften" > "Allgemein" > "Verbindungsparameter".
3. Wählen Sie unter Endpunkt den Partner-Endpunkt aus. Nutzen Sie einen der beiden folgenden Partner-Endpunkten:
 - CPU S7-1500 mit CM 1542-5
 - Unspezifiziert
4. Wählen Sie unter Konfigurationsart "Konfigurierte Verbindung verwenden aus".
5. Wählen Sie unter Verbindungstyp "FDL" aus.
6. Wählen Sie unter Schnittstelle die folgenden Schnittstellen aus:
 - Lokal: PROFIBUS-Schnittstelle des CM 1542-5
 - Spezifizierter Partner: PROFIBUS-Schnittstelle des CM 1542-5
7. Wählen Sie bei Verbindungsdaten die Einstellung <neu> aus.

Das folgende Bild zeigt eine vollständig konfigurierte FDL-Verbindung in STEP 7.

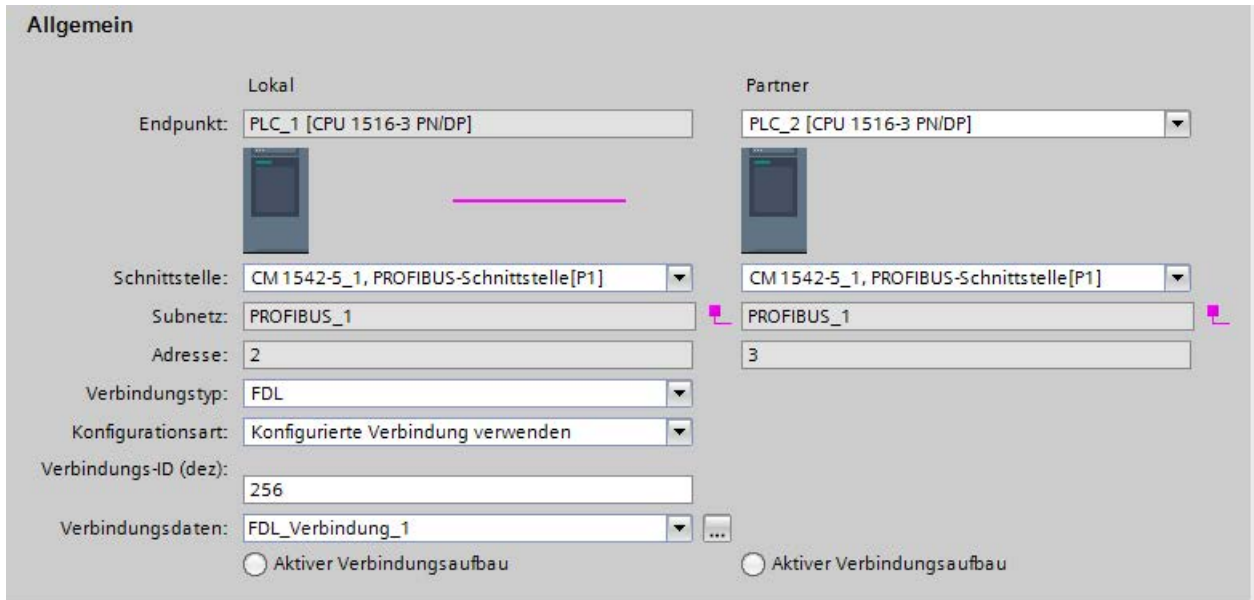


Bild 6-6 FDL-Verbindung konfigurieren

Einrichten einer FDL-Verbindung im Anwenderprogramm

Für die Kommunikation über FDL müssen Sie jeweils den Datenbaustein des Systemdatentyps TCON_FDL selbst erstellen, parametrieren und direkt an der Anweisung aufrufen. Gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_FDL.

Das folgende Beispiel zeigt den globalen Datenbaustein "FDL_connection", in dem die Variable "FDL_connection" vom Datentyp TCON_FDL definiert ist.

FDL_connection										
	Name	Datentyp	Startwert	R...	E...	S...	S...	E...	Überw..	Kommentar
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	FDL_connection	TCON_FDL		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	InterfaceId	HW_ANY	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		HW identifier of PB interface submodule
4	ID	CONN_OUC	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		connection reference / identifier
5	ConnectionType	Byte	16#15	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		type of connection: 21= FDL connection
6	ActiveEstablished	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		active/passive connection establishment
7	ServiceId	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		service id: 0 – default, 1 – SDA, 2 – SDN
8	RemotePBAddress	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		remote Profibus partner address
9	LocalPBAddress	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		local Profibus partner address
10	RemoteLSAP	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		remote PB link-layer service access point
11	LocalLSAP	Byte	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		local PB link-layer service access point

Bild 6-7 FDL-Verbindung programmieren

3. Programmieren Sie die Parameter der FDL-Verbindung (z. B. die PROFIBUS-Adressen) in der Variablen vom Datentyp TCON_FDL.

4. Legen Sie im Programmeditor eine Anweisung TCON an.
5. Verschalten Sie den Parameter CONNECT der Anweisung TCON mit der Variable vom Datentyp TCON_FDL.

Im folgenden Beispiel ist der Parameter CONNECT der Anweisung TCON mit der Variablen "FDL_Connection" (Datentyp TCON_FDL) verschaltet.

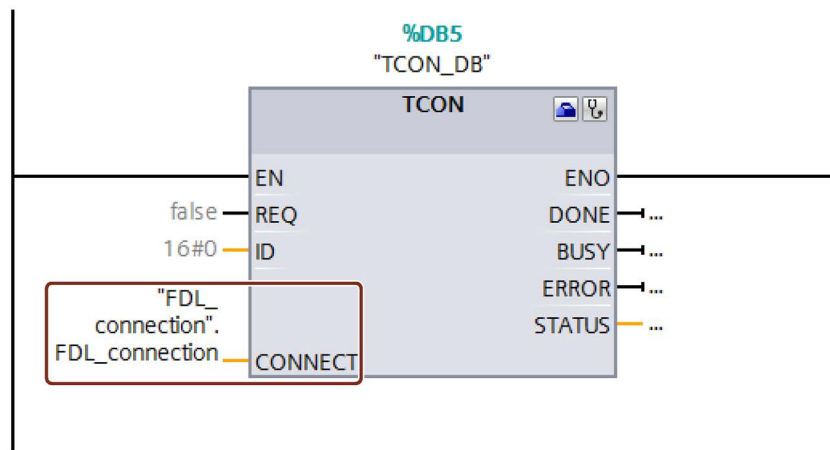


Bild 6-8 Beispiel: Anweisung TCON für FDL-Verbindung

6.7 Kommunikation über Modbus TCP einrichten

Einrichten einer Verbindung über das Anwenderprogramm für Modbus TCP

Die Parametrierung erfolgt im Programmeditor an der Anweisung MB_CLIENT bzw. MB_SERVER.

Vorgehen zum Einrichten der Kommunikation über Modbus TCP

Die Anweisung MB_CLIENT kommuniziert als Modbus TCP-Client über die TCP-Verbindung. Mit der Anweisung bauen Sie eine Verbindung zwischen dem Client und dem Server auf, senden Modbus-Anfragen zum Server und empfangen die entsprechenden Modbus-Antworten. Weiterhin steuern Sie mit dieser Anweisung den Abbau der TCP-Verbindung.

Die Anweisung MB_SERVER kommuniziert als Modbus TCP-Server über eine TCP-Verbindung. Die Anweisung verarbeitet Verbindungsanfragen eines Modbus-Clients, empfängt und bearbeitet Modbus-Anfragen und sendet Antwort-Meldungen. Weiterhin können Sie den Abbau der TCP-Verbindung steuern.

Voraussetzung: Der Client kann den Server über IP-Kommunikation im Netzwerk erreichen.

1. Konfigurieren Sie in der Netzsicht des Hardware- und Netzwerkeditors von STEP 7 ein Automatisierungssystem S7-1500 mit CPU.
2. Wählen Sie in der Projektnavigation für die CPU den Ordner "Programmbausteine" und öffnen Sie im Ordner den OB 1 durch Doppelklick. Der Programmeditor öffnet sich.
3. Wählen Sie aus der Task Card "Anweisungen", Bereich "Kommunikation", "Weitere", "MODBUS TCP" die gewünschte Anweisung, z. B. MB_CLIENT und ziehen Sie sie per Drag & Drop in ein Netzwerk des OB 1.

4. Parametrieren Sie die Anweisung MB_CLIENT bzw. MB_SERVER. Halten Sie dabei folgende Regeln ein:

Für jede MB_CLIENT-Verbindung muss eine IPv4-Adresse des Servers spezifiziert sein.

Jede Verbindung MB_CLIENT oder MB_SERVER muss einen eindeutigen Instanz-DB mit einer der Datenstrukturen TCON_IP_v4 oder TCON_QDN verwenden.

Jede Verbindung benötigt eine eindeutige Verbindungs-ID. Verbindungs-ID und Instanz-DB gehören jeweils paarweise zusammen und müssen für jede Verbindung eindeutig sein.

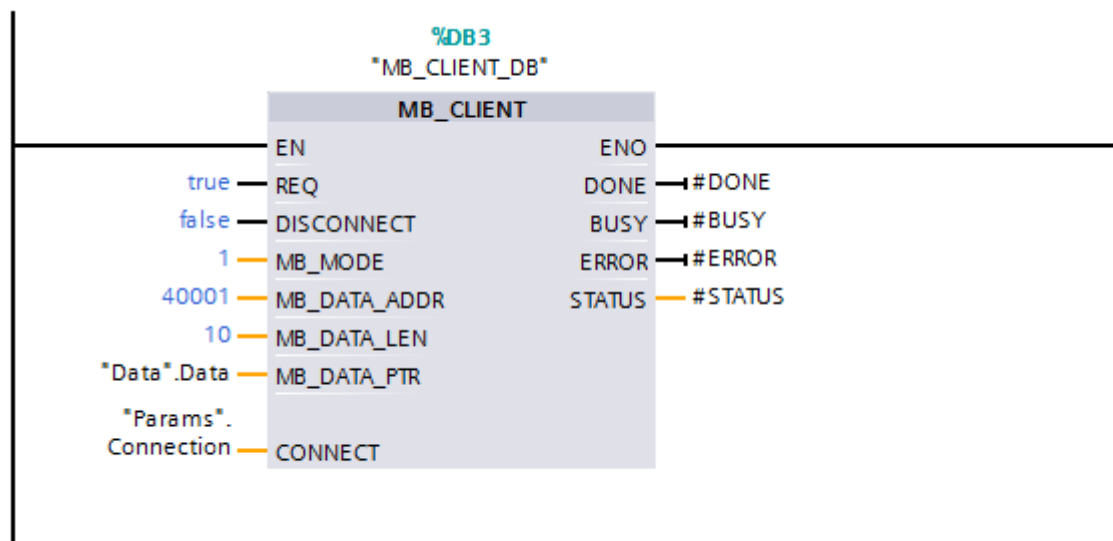


Bild 6-9 MB_CLIENT

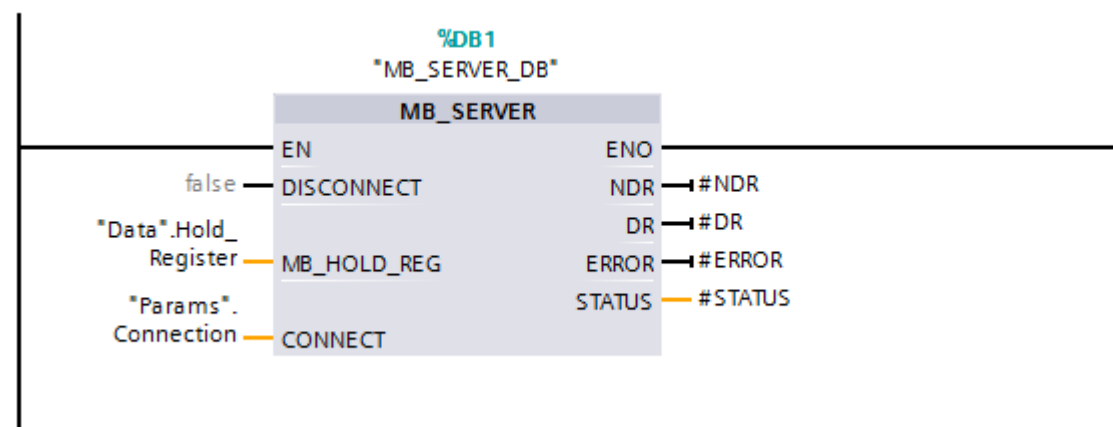


Bild 6-10 MB_SERVER

5. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Modbus TCP-Server als Gateway zu Modbus RTU

Wenn Sie einen Modbus TCP-Server als Gateway zu einem Modbus RTU-Protokoll verwenden, dann adressieren Sie das Slavegerät im seriellen Netzwerk über den statischen Parameter MB_UNIT_ID. Der Parameter MB_UNIT_ID entspricht dem Feld der Slaveadresse beim Modbus RTU-Protokoll. Der Parameter MB_UNIT_ID würde in diesem Fall die Anforderung an die richtige Modbus RTU-Slaveadresse weiterleiten.

Die Gateway Funktion müssen Sie selbst programmieren.

Den Parameter MB_UNIT_ID finden Sie in dem der Anweisung MB_CLIENT zugehörigen Instanzdatenbaustein.

Weitere Informationen zum Parameter MB_UNIT_ID finden Sie in der Online-Hilfe zu STEP 7.

Modbus TCP-Kommunikation zwischen zwei S7-1500 CPUs

Wie Sie die Modbus-TCP-Kommunikation zwischen zwei S7-1500 CPUs programmieren und parametrieren, finden Sie in diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/94766380>).

6.8 Kommunikation über E-Mail einrichten

Einrichten einer Verbindung über das Anwenderprogramm für E-Mail

Für die Kommunikation über E-Mail müssen Sie jeweils den Datenbaustein des entsprechenden Systemdatentyps selbst erstellen, parametrieren und direkt an der Anweisung aufrufen. Das Vorgehen ist nachfolgend dargestellt.

Vorgehen zum Einrichten der Kommunikation über E-Mail

E-Mails können von einer CPU gesendet werden. Für das Senden von E-Mails aus dem Anwenderprogramm der CPU setzen Sie die Anweisung TMAIL_C ein.

Voraussetzung: Der SMTP-Server ist über das IPv4-Netzwerk erreichbar.

1. Konfigurieren Sie in der Netzsicht des Hardware- und Netzwerkeditors von STEP 7 ein Automatisierungssystem S7-1500 mit CPU.
2. Parametrieren Sie die Anweisung TMAIL_C, geben Sie z. B. bei Subject den Betreff der Mail ein.
3. Erzeugen Sie in einem globalen Datenbaustein eine Variable vom Typ TMAIL_v4, TMAIL_v6 (nur CP 1543-1) oder TMAIL_FQDN (nur CP 1543-1).
4. Stellen Sie in der Variable die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "MailServerAdress" die IPv4-Adresse des Mailservers ein (für TMAIL_v4)

Hinweis

Verbindungsparameter Interfaceld und ID

Beachten Sie, daß Sie ab Anweisungsversion V5.0 der Anweisung TMAIL_C im Datentyp TMAIL_V4_SEC den Wert "0" für die Interfaceld und die ID eintragen können. In diesem Fall sucht die CPU selbst nach einer passenden lokalen Schnittstelle der CPU bzw. einer freien Verbindungs-ID.

Verschalten Sie die Variable mit dem Parameter MAIL_ADDR_PARAM der Anweisung TMAIL_C.

5. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Weitere Informationen

In der Online-Hilfe STEP 7 finden Sie beschrieben:

- die Systemdatentypen
- die Anweisungen für offene Kommunikation
- die Verbindungsparameter

6.9 Kommunikation über FTP einrichten

Einrichten einer Verbindung über das Anwenderprogramm für FTP

Für die Kommunikation über FTP müssen Sie jeweils den Datenbaustein des entsprechenden Systemdatentyps selbst erstellen, parametrieren und direkt an der Anweisung aufrufen. Das Vorgehen ist nachfolgend dargestellt.

FTP-Client und -Server-Funktionalität

Dateien können von einer CPU an einen FTP-Server gesendet und von diesem empfangen werden. Die Kommunikation über FTP ist für S7-1500 nur über CP 1543-1 möglich. Der CP kann FTP-Server, FTP-Client oder beides sein. FTP-Clients können auch Fremdsysteme/PCs sein.

Für die FTP-Server-Funktionalität projektieren Sie den CP entsprechend in STEP 7.

Mit der FTP-Client-Funktionalität realisieren Sie z. B. den Aufbau und Abbau einer FTP-Verbindung, das Übertragen und Löschen von Dateien auf dem Server. Für die FTP-Client-Funktionalität setzen Sie die Anweisung FTP_CMD ein.

Vorgehen zum Einrichten der FTP-Server-Funktionalität

Voraussetzung: Der FTP-Server ist über das IPv4-Netzwerk erreichbar.

1. Konfigurieren Sie in der Gerätesicht des Hardware- und Netzwerkeditors von STEP 7 ein Automatisierungssystem S7-1500 mit CPU und CP 1543-1.

Zugleich müssen Sie in der HW-Konfiguration der CPU S7-1500 unter der Bereichsnavigation "Schutz" im Abschnitt "Verbindungsmechanismen" die CheckBox "Zugriff über PUT/GET Kommunikation durch entfernten Partner (PLC, HMI, OPC, ...) erlauben" aktivieren.

2. Nehmen Sie in den Eigenschaften des CP unter "FTP-Konfiguration" folgende Einstellungen vor:
 - Wählen Sie das Auswahlkästchen "FTP-Server für S7-CPU-Daten verwenden" an.
 - Ordnen Sie die CPU, einen Datenbaustein und einen Dateinamen, unter dem der DB für FTP abgelegt wird zu.

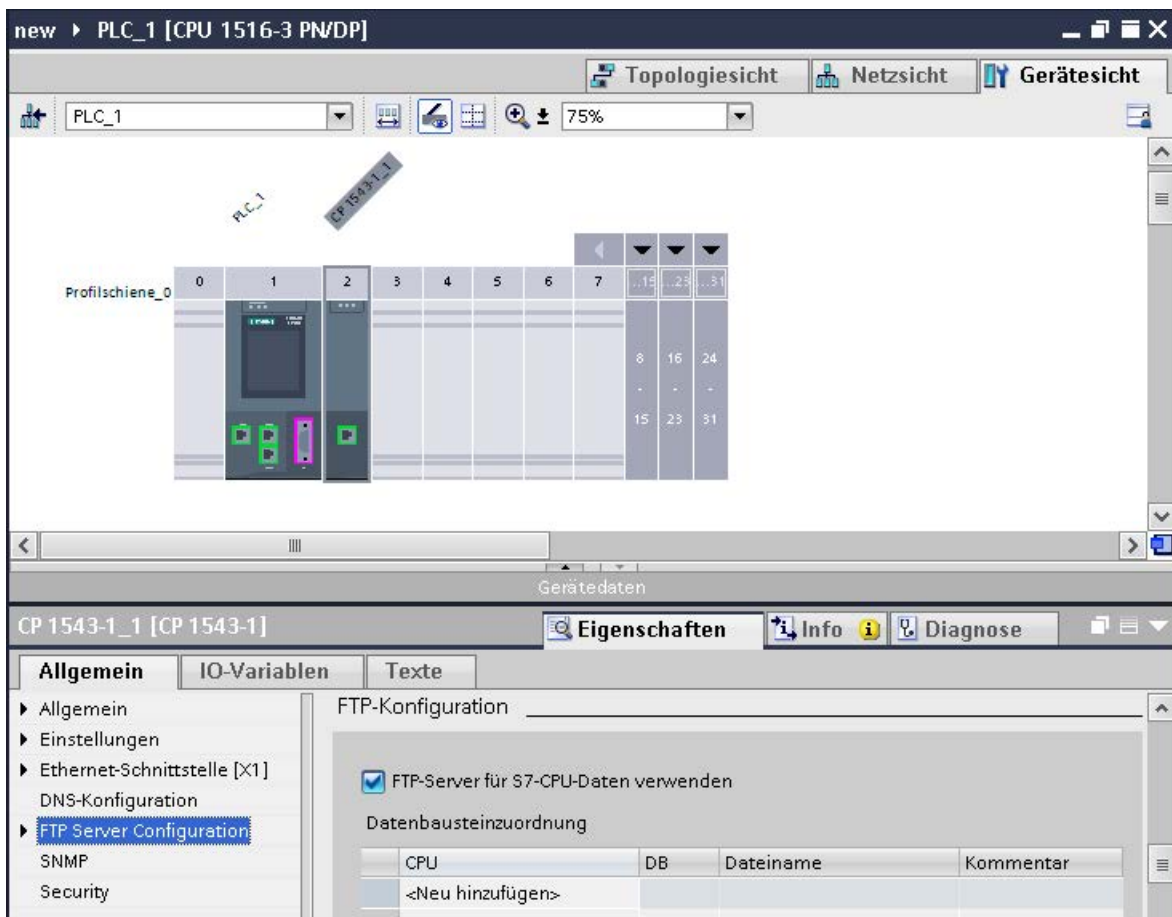


Bild 6-11 FTP-Konfiguration einrichten

3. Laden Sie die Hardware-Konfiguration in die CPU.

Vorgehen zum Einrichten der FTP-Client-Funktionalität

Voraussetzung: Der FTP-Server ist über das IPv4-Netzwerk erreichbar.

1. Konfigurieren Sie in der Gerätesicht des Hardware- und Netzwerkeditors von STEP 7 ein Automatisierungssystem S7-1500 mit CPU und CP 1543-1.

Zugleich müssen Sie in der HW-Konfiguration der CPU S7-1500 unter der Bereichsnavigation "Schutz" im Abschnitt "Verbindungsmechanismen" die CheckBox "Zugriff über PUT/GET Kommunikation durch entfernten Partner (PLC, HMI, OPC, ...) erlauben" aktivieren.
2. Rufen Sie die Anweisung FTP_CMD im Anwenderprogramm der CPU auf.
3. Parametrieren Sie an der Anweisung FTP_CMD die Verbindungsparameter für den FTP-Server.
4. Erzeugen Sie einen Global-DB und innerhalb dieses Global-DBs eine Variable vom Type FTP_CONNECT_IPV4, FTP_CONNECT_IPV6 oder FTP_CONNECT_NAME.
5. Verschalten Sie die Variable innerhalb des Datebausteins mit der Anweisung FTP_CMD.
6. Für die Verbindung zum FTP-Server geben Sie im DB an:
 - den Benutzernamen, das Kennwort und die IP-Adresse für den FTP-Zugang im entsprechenden Datentyp (FTP_CONNECT_IPV4, FTP_CONNECT_IPV6 oder FTP_CONNECT_NAME)
7. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Anwendungsbeispiele

- Anwendungsbeispiel: FTP-Kommunikation mit S7-1500 und CP 1543-1
Das Anwendungsbeispiel finden Sie im Internet
(<https://support.industry.siemens.com/cs/ww/de/view/103550797>).
- Anwendungsbeispiel: Bewerten FTP-Client Kommunikation mit S7-1200/1500
Das Anwendungsbeispiel finden Sie im Internet
(<https://support.industry.siemens.com/cs/ww/de/view/81367009>).

Weitere Informationen

In der Online-Hilfe STEP 7 finden Sie beschrieben:

- die Systemdatentypen
- die Anweisungen für offene Kommunikation
- die Verbindungsparameter

6.10 Auf- und Abbau von Kommunikationsbeziehungen

Auf- und Abbau von Kommunikationsbeziehungen

Die folgende Tabelle zeigt den Auf- und Abbau von Kommunikationsbeziehungen im Rahmen der offenen Kommunikation.

Tabelle 6- 5 Auf- und Abbau von Kommunikationsbeziehungen

Einrichten der Verbindung	Aufbau der Kommunikationsbeziehung	Abbau der Kommunikationsbeziehung
Über Anwenderprogramm	<p>Nach dem Laden des Anwenderprogramms in die CPUs:</p> <p>Der passive Kommunikationspartner richtet mit dem Aufruf von TSEND_C/TRCV_C bzw. TCON den lokalen Verbindungszugang ein. Der Aufruf von TSEND_C/TRCV_C bzw. TCON im aktiven Partner startet den Verbindungsaufbau. Konnte die Verbindung aufgebaut werden, erfolgt eine positive Rückmeldung an den Anweisungen im Anwenderprogramm.</p> <p>Nachdem Sie eine Verbindung mit der Anweisung T_RESET abgebaut haben, wird die Verbindung neu aufgebaut.</p> <p>Bei einem Verbindungsabbruch versucht der aktive Partner, die eingerichtete Verbindung wieder aufzubauen. Dies gilt nur, wenn zuvor der Verbindungsaufbau mit TCON erfolgreich war.</p>	<ul style="list-style-type: none"> • Über die Anweisungen TSEND_C/TRCV_C, TDISCON und T_RESET • Wenn die CPU vom Betriebszustand RUN in STOP übergeht • Bei NETZ-AUS/NETZ-EIN Ein an einer CPU
Über Verbindungsprojektion	Nach dem Laden der Verbindungsprojektion und des Anwenderprogramms in die CPUs.	Durch Löschen der Verbindungsprojektion in STEP 7 und Laden der geänderten Projektion in die CPU.

6.11 Secure Open User Communication

6.11.1 Secure OUC von einer S7-1500 CPU als TLS-Client zu einem Fremd-PLC (TLS-Server)

Im Folgenden ist beschrieben, wie Sie eine Secure Open User Communication über TCP von einer S7-1500 CPU als TLS-Client zu einem TLS-Server einrichten.

Gesicherte TCP-Verbindung von einer S7-1500 CPU als TLS-Client zu einem TLS-Server einrichten

S7-1500 CPUs ab Firmwarestand V2.0 unterstützen Secure Communication mit Adressierung über ein Domain Name System (DNS).

Für die gesicherte TCP-Kommunikation über den Domainnamen müssen Sie selbst einen Datenbaustein mit dem Systemdatentyp TCON_QDN_SEC erstellen, parametrieren und direkt an einer der Anweisungen TSEND_C, TRCV_C oder TCON aufrufen.

Voraussetzungen:

- Aktuelles Datum und Uhrzeit sind in der CPU eingestellt.
- In Ihrem Netz befindet sich mindestens ein DNS-Server.
- Sie haben für die S7-1500 CPU mindestens einen DNS-Server konfiguriert.
- TLS-Client und TLS-Server besitzen alle notwendigen Zertifikate.

Um eine gesicherte TCP-Verbindung zu einem TLS-Server einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_QDN_SEC.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "DNS ConnectionSEC" vom Datentyp TCON_QDN_SEC definiert ist.

Data_block_1				
	Name	Data type	Start value	Comment
1	Static			
2	DNS Connection SEC	TCON_QDN_SEC		
3	ConnPara	TCON_QDN		parameter of the TCP connection
4	Interfaceld	HW_ANY	0	not relevant
5	ID	CONN_OUC	5	connection reference / identifier
6	ConnectionType	Byte	11	type of connection: 16#0B=11=TCP/IP, 16#13=13=SSL/TLS
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteQDN	String[254]	'plc_1.factory127.'	fully or partially qualified domain name of remote host
9	RemotePort	UInt	4000	remote UDP / TCP port number
10	LocalPort	UInt	0	local UDP / TCP port number
11	ActivateSecureConn	Bool	true	activate the security functionality of that connection
12	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
13	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address
14	TLSServerCertRef	UDInt	7	for Server side: Reference to own X.509 V3 server certificate
15	TLSClientCertRef	UDInt	0	for Client side: add id of own X.509 V3 client certificate

Bild 6-12 Datentyp TCON_QDN_SEC

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "RemoteQDN" den vollqualifizierten Domainnamen (FQDN) des TLS-Servers ein.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein.
 - "ActivateSecureConn": Aktivierung von secure communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine non-secure TCP- oder UDP-Verbindung einrichten.
 - "ExtTlSCapabilities": Wenn Sie den Wert 1 eintragen, dann validiert der Client den subjectAlternateName im X.509-V3-Zertifikat des Servers, um die Identität des Servers zu überprüfen. Diese Validierung erfolgt im Kontext der Anweisung.
 - "TLSServerCertRef": ID des X.509-V3-Zertifikats (gewöhnlich ein CA-Zertifikat), das vom TLS-Client benutzt wird, um die Authentifizierung des TLS-Servers zu validieren. Wenn dieser Parameter 0 ist, benutzt der TLS-Client zur Validierung der Server-Authentifizierung alle (CA-) Zertifikate, die aktuell im Certificate Store des Clients geladen sind.

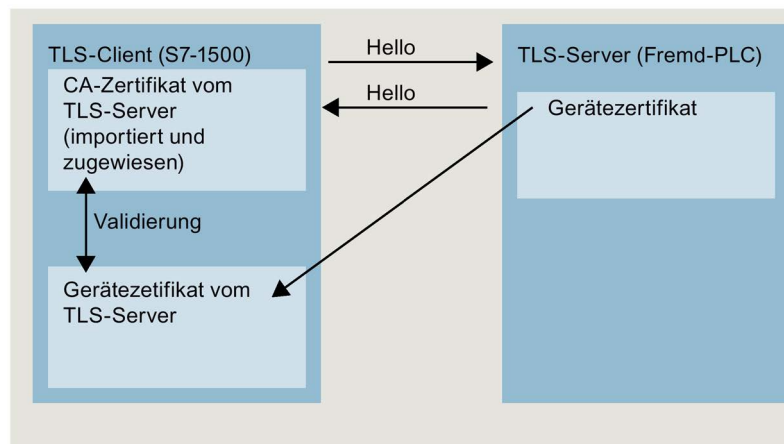


Bild 6-13 Zertifikate-Handling aus Perspektive der S7-1500 als TLS-Client

- "TLSClientCertRef": ID des eigenen X.509-V3-Zertifikats.

5. Legen Sie im Programmeditor eine der Anweisungen TSEND_C, TRCV_C oder TCON an.
6. Verschalten Sie den Parameter CONNECT einer der Anweisungen TSEND_C, TRCV_C oder TCON mit der Variablen vom Datentyp TCON_QDN_SEC.

Im folgenden Beispiel ist der Parameter CONNECT der Anweisung TCON mit der Variablen "DNS connectionSEC" (Datentyp TCON_QDN_SEC) verschaltet.

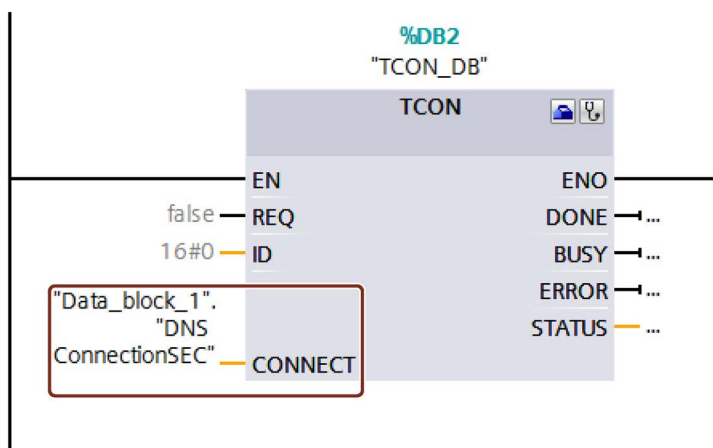


Bild 6-14 Anweisung TCON

Weitere Informationen

Weitere Informationen zum Systemdatentyp TCON_QDN_SEC finden Sie in der Onlinehilfe zu STEP 7.

Weitere Informationen zur sicheren Kommunikation finden Sie im Kapitel Secure Communication (Seite 38).

6.11.2 Secure OUC von einer S7-1500 CPU als TLS-Server zu einem Fremd-PLC (TLS-Client)

Im Folgenden ist beschrieben, wie Sie eine Secure Open User Communication über TCP von einer S7-1500 CPU als TLS-Server zu einem TLS-Client einrichten.

Gesicherte TCP-Verbindung über den Domainnamen des Kommunikationspartners einrichten

S7-1500 CPUs ab Firmwarestand V2.0 unterstützen Secure Communication mit Adressierung über ein Domain Name System (DNS).

Für die gesicherte TCP-Kommunikation über den Domainnamen müssen Sie selbst einen Datenbaustein mit dem Systemdatentyp TCON_QDN_SEC erstellen, parametrieren und direkt an einer der Anweisungen TSEND_C, TRCV_C oder TCON aufrufen.

Voraussetzungen:

- Aktuelles Datum und Uhrzeit sind in der CPU eingestellt.
- In Ihrem Netz befindet sich mindestens ein DNS-Server.
- Sie haben für die S7-1500 CPU mindestens einen DNS-Server konfiguriert.
- TLS-Client und TLS-Server besitzen alle notwendigen Zertifikate.

Um eine gesicherte TCP-Verbindung zu einem TLS-Client einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_QDN_SEC.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "DNS ConnectionSEC" vom Datentyp TCON_FDL_SEC definiert ist.

Data_block_1				
	Name	Data type	Start value	Comment
1	▼ Static			
2	▼ DNS Connection SEC2	TCON_QDN_SEC		
3	▼ ConnPara	TCON_QDN		parameter of the TCP connection
4	Interfaceld	HW_ANY	0	not relevant
5	ID	CONN_OUC	8	connection reference / identifier
6	ConnectionType	Byte	11	type of connection: 16#0B=11=TCP/IP, 16#13=...
7	ActiveEstablished	Bool	false	active/passive connection establishment
8	RemoteQDN	String[254]	"	fully or partially qualified domain name of rem
9	RemotePort	UInt	0	remote UDP / TCP port number
10	LocalPort	UInt	2010	local UDP / TCP port number
11	ActivateSecureConn	Bool	true	activate the security functionality of that conn
12	TLSReqClientCert	Bool	false	Just for server side: The TLS server requests a
13	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 ad
14	TLSReqServerCertRef	UDInt	5	for Server side: Reference to own X.509 V3 se
15	TLSClientCertRef	UDInt	0	for Client side: add id of own X.509 V3 client c

Bild 6-15 TCON_QDN_SEC_Server

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "ID" die lokale ID der TCP-Verbindung ein.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein.
- "ActivateSecureConn": Aktivierung von secure communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine non-secure TCP- oder UDP-Verbindung einrichten.
 - "TLSServerReqClientCert": Anforderung eines X.509-V3-Zertifikats vom TLS-Client.
 - "TLSServerCertRef": ID des eigenen X.509-V3-Zertifikats.

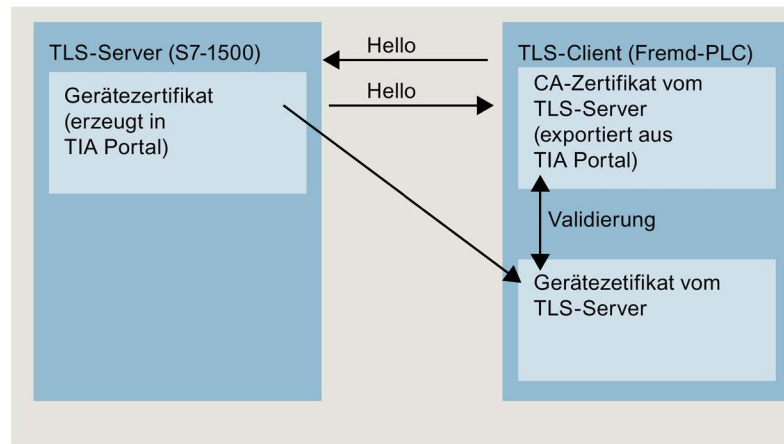


Bild 6-16 Zertifikate-Handling aus Perspektive der S7-1500 als TLS-Server

- "TLSCClientCertRef": ID des X.509-V3-Zertifikats (oder einer Gruppe von X.509-V3-Zertifikaten), das vom TLS-Server benutzt wird, um die Authentifizierung des TLS-Clients zu validieren. Wenn dieser Parameter 0 ist, benutzt der TLS-Server zur Validierung der Client-Authentifizierung alle (CA-) Zertifikate, die aktuell im Certificate Store des Servers geladen sind.

5. Legen Sie im Programmierer eine der Anweisungen TSEND_C, TRCV_C oder TCON an.
6. Verschalten Sie den Parameter CONNECT einer der Anweisungen TSEND_C, TRCV_C oder TCON mit der Variablen vom Datentyp TCON_QDN_SEC.

Im folgenden Beispiel ist der Parameter CONNECT der Anweisung TCON mit der Variablen "DNS connectionSEC" (Datentyp TCON_QDN_SEC) verschaltet.

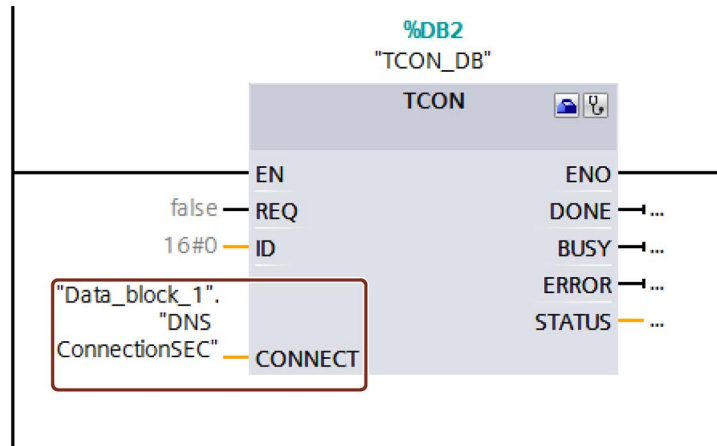


Bild 6-17 Anweisung TCON

Weitere Informationen

Weitere Informationen zu den Systemdatentypen TCON_QDN_SEC finden Sie in der Onlinehilfe zu STEP 7.

Weitere Informationen zur sicheren Kommunikation finden Sie im Kapitel Secure Communication (Seite 38).

6.11.3 Secure OUC zwischen zwei S7-1500 CPUs

Im Folgenden ist beschrieben, wie Sie eine Secure Open User Communication über TCP zwischen zwei S7-1500 CPUs einrichten. Dabei agiert eine S7-1500 CPU als TLS-Client (aktiver Verbindungsaufbau) und die andere S7-1500 CPU als TLS-Server (passiver Verbindungsaufbau).

Gesicherte TCP-Verbindung zwischen zwei S7-1500 CPUs einrichten

Für die gesicherte TCP-Kommunikation zwischen zwei S7-1500 CPUs müssen Sie in jeder CPU einen Datenbaustein mit dem Systemdatentyp TCON_IPv4_SEC erstellen, parametrieren und direkt an einer der Anweisungen TSEND_C, TRCV_C oder TCON aufrufen.

Voraussetzungen:

- Aktuelles Datum und Uhrzeit sind in der CPU eingestellt.
- Beide S7-1500 CPU haben mindestens Firmwarestand V2.0
- TLS-Client und TLS-Server besitzen alle notwendigen Zertifikate.

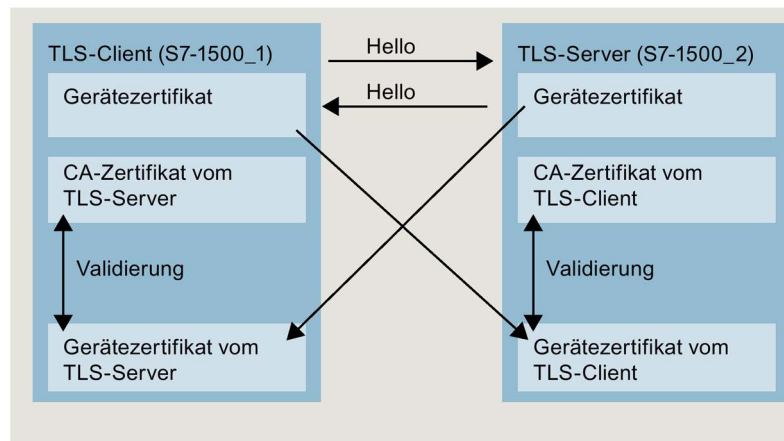


Bild 6-18 Zertifikate-Handling bei Secure OUC zwischen zwei S7-1500 CPUs

Einstellungen am TLS-Client

Um eine gesicherte TCP-Verbindung im TLS-Client einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_IP_V4_SEC.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "SEC connection 1 TLS-Client" vom Datentyp TCON_IP_V4_SEC definiert ist.

Data_block_1				
	Name	Data type	Start value	Comment
1	▼ Static			
2	▼ SEC connection 1 TLS-Client	TCON_IP_V4_SEC		
3	▼ ConnPara	TCON_IP_v4		parameter of the TCP connection
4	Interfaceld	HW_ANY	72	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connction: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	▼ RemoteAddress	IP_V4		remote IP address (IPv4)
9	▼ ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	100	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against the
19	TLSServerCertRef	UDInt	1	for Server side: Reference to own X.509 V3 server certificate; for
20	TLSClientCertRef	UDInt	5	for Client side: add id of own X.509 V3 client certificate; for Sen

Bild 6-19 IP_V4_SEC_Client

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "RemoteAdress" die IPv4-Adresse des TLS-Servers ein.

Hinweis

Verbindungsparameter Interfaceld und ID

Beachten Sie, daß Sie im Datentyp TCON_V4_SEC den Wert "0" für die Interfaceld und die ID eintragen können. In diesem Fall sucht die CPU selbst nach einer passenden lokalen Schnittstelle der CPU bzw. einer freien Verbindungs-ID.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein.
 - "ActivateSecureConn": Aktivierung von secure communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine non-secure TCP- oder UDP-Verbindung einrichten.
 - "TLSServerCertRef": Tragen Sie den Wert 2 (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA256) bzw. den Wert 1 ein (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA1)). Wenn Sie ein anderes CA-Zertifikat nutzen, tragen Sie die entsprechende ID aus dem Zertifikatsmanager der Globalen Security-Einstellungen ein.
 - "TLSClientCertRef": ID des eigenen X.509-V3-Zertifikats.
5. Legen Sie im Programmeditor eine der Anweisungen TSEND_C, TRCV_C oder TCON an.
6. Verschalten Sie den Parameter CONNECT einer der Anweisungen TSEND_C, TRCV_C oder TCON mit der Variablen vom Datentyp TCON_IP_V4_SEC.

Einstellungen am TLS-Server

Um eine gesicherte TCP-Verbindung im TLS-Server einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_IP_V4_SEC.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "SEC connection 1 TLS-Server" vom Datentyp TCON_IP_V4_SEC definiert ist.

Data_block_1				
	Name	Data type	Start value	Comment
1	Static			
2	SEC connection 1 TLS-Server	TCON_IP_V4_SEC		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	Interfaceld	HW_ANY	120	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connetion: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	false	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	10	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	true	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against the
19	TLSServerCertRef	UDInt	6	for Server side: Reference to own X.509 V3 server certificate; for
20	TLSClientCertRef	UDInt	1	for Client side: add id of own X.509 V3 client certificate; for Serv

Bild 6-20 IP_V4_SEC_Server

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "RemoteAdress" die IPv4-Adresse des TLS-Clients ein.
4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein.
 - "ActivateSecureConn": Aktivierung von secure communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine non-secure TCP- oder UDP-Verbindung einrichten.
 - "TLSServerReqClientCert ": Anforderung eines X.509-V3-Zertifikats vom TLS-Client. Tragen Sie den Wert "true" ein.
 - "TLSServerCertRef": ID des eigenen X.509-V3-Zertifikats.
 - "TLSClientCertRef": Tragen Sie den Wert 2 (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA256) bzw. den Wert 1 ein (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA1)). Wenn Sie ein anderes CA-Zertifikat nutzen, tragen Sie die entsprechende ID aus dem Zertifikatsmanager der Globalen Security-Einstellungen ein.
5. Legen Sie im Programmeditor einer der Anweisungen TSEND_C, TRCV_C oder TCON an.
6. Verschalten Sie den Parameter CONNECT einer der Anweisungen TSEND_C, TRCV_C oder TCON mit der Variablen vom Datentyp TCON_IP_V4_SEC.

Im folgenden Beispiel ist der Parameter CONNECT der Anweisung TSEND_C mit der Variablen "SEC connection 1 TLS-Server" (Datentyp TCON_IP_V4_SEC) verschaltet.

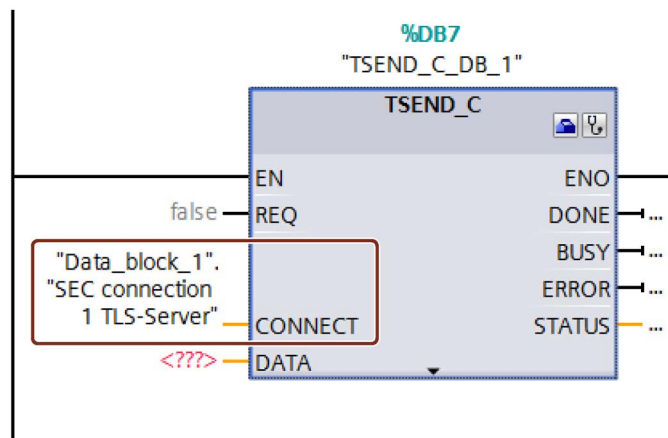


Bild 6-21 TSEND_C

Weitere Informationen

Weitere Informationen zu den Systemdatentypen TCON_IP_V4_SEC finden Sie in der Onlinehilfe zu STEP 7.

Weitere Informationen zur sicheren Kommunikation finden Sie im Kapitel Secure Communication (Seite 38).

6.11.4 Secure OUC über CP-Schnittstelle

Im Folgenden sind die Besonderheiten beschrieben, die bei Secure Open User Communication über eine CP-Schnittstelle zu berücksichtigen sind. Mindestens eine Station ist eine S7-1500 Station mit folgenden Baugruppen:

- S7-1500 CPU ab Firmwarestand V2.0 (ausgenommen S7-1500 Software Controller)
- CP 1543-1 ab Firmwarestand V2.0 bzw. CP 1543SP-1 ab Firmwarestand V1.0

Der CP agiert in einer S7-1500 Station als TLS-Client (aktiver Verbindungsaufbau) oder als TLS-Server (passiver Verbindungsaufbau).

Die grundsätzliche Vorgehensweise und das Konzept für die Nutzung von Secure Communication über eine CP-Schnittstelle ist ähnlich wie Secure Communication über die Schnittstellen der S7-1500 CPUs. Im Wesentlichen müssen Sie dem CP in der Rolle als TLS-Server oder TLS-Client die Zertifikate zuordnen und nicht der CPU. Daher gelten andere Regeln und Vorgehensweisen, die im Folgenden beschrieben sind.

Hantierung von Zertifikaten für CPs

Generell gilt: Sie müssen beim Zertifikatsmanager in den Globalen Security-Einstellungen angemeldet sein. Auch das Erstellen von selbst signierten Zertifikaten ist nicht ohne Anmeldung für die Globalen Security-Einstellungen möglich! Sie müssen als Benutzer mit ausreichenden Rechten ausgestattet sein (Administrator oder Benutzer mit der Rolle "Standard" mit dem Recht "Security konfigurieren").

Ausgangspunkt für die Erstellung oder Zuweisung von Zertifikaten beim CP ist der Bereich "Security > Security-Eigenschaften". In diesem Bereich melden Sie sich für die Globalen Security-Einstellungen an.

Vorgehensweise:

1. Markieren Sie in der Netzsicht von STEP 7 den CP und wählen im Inspektorfenster den Bereich "Security > Security-Eigenschaften".
2. Klicken Sie auf die Schaltfläche "Benutzeranmeldung".
3. Melden Sie sich mit Benutzernamen und Passwort an.
4. Aktivieren Sie die Option "Aktiviere Security-Funktionen".

Die Security-Eigenschaften werden initialisiert.

5. Klicken Sie in die erste Zeile der Tabelle "Gerätezertifikate", um ein neues Gerätezertifikat zu erzeugen oder ein bestehendes Gerätezertifikat auszuwählen.
6. Falls der Kommunikationspartner ebenfalls eine S7-1500 Station ist, müssen Sie dem Kommunikationspartner ebenfalls mit STEP 7 ein Gerätezertifikat zuweisen wie hier bzw. wie bei der S7-1500 CPU beschrieben.

Beispiel: Gesicherte TCP-Verbindung zwischen zwei S7-1500 CPUs über CP-Schnittstellen einrichten

Für die gesicherte TCP-Kommunikation zwischen zwei S7-1500 CPs müssen Sie in jeder CPU einen Datenbaustein mit dem Systemdatentyp TCON_IPv4_SEC erstellen, parametrieren und direkt an einer der Anweisungen TSEND_C, TRCV_C oder TCON aufrufen.

Voraussetzungen:

- Beide S7-1500 CPUs haben mindestens Firmwarestand V2.0; wenn Sie den CP 1543SP-1 verwenden: Firmwarestand ab V1.0.
- Beide CPs (z. B. CP 1543-1) haben mindestens Firmwarestand V2.0
- TLS-Client und TLS-Server besitzen alle notwendigen Zertifikate.
 - Ein Gerätezertifikat (End-Entity-Zertifikat) für den CP muss erzeugt sein und sich im Zertifikatsspeicher des CP befinden. Wenn ein Kommunikationspartner ein Fremdgerät ist (z. B. ein MES oder ERP-System), muss für dieses Gerät ebenfalls ein Gerätezertifikat vorhanden sein.
 - Das Stammzertifikat (CA-Zertifikat), mit dem das Gerätezertifikat des Kommunikationspartners signiert ist, muss sich im Zertifikatsspeicher des CP bzw. im Zertifikatsspeicher des Fremdgeräts befinden. Falls Sie Zwischenzertifikate nutzen, müssen Sie sicherstellen, dass der gesamte Zertifikatepfad im validierenden Gerät vorhanden ist. Diese Zertifikate nutzt ein Gerät zur Validierung des Gerätezertifikats des Kommunikationspartners.
- Den Kommunikationspartner müssen Sie grundsätzlich über seine IPv4-Adresse adressieren, nicht über seinen Domännennamen.

Das folgende Bild zeigt die verschiedenen Zertifikate in den Geräten für den Fall, dass beide Kommunikationspartner über einen CP 1543-1 kommunizieren. Außerdem zeigt das Bild die Übertragung der Gerätezertifikate beim Verbindungsaufbau ("Hello").

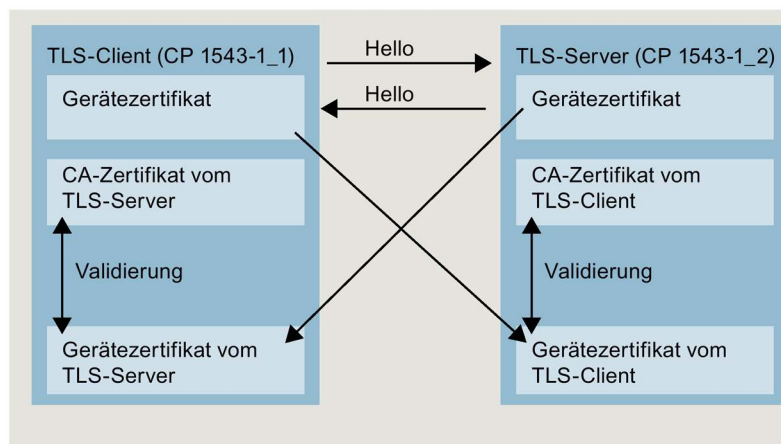


Bild 6-22 Zertifikate-Handling bei Secure OUC zwischen zwei S7-1500 CPUs über CP-Schnittstellen

Einstellungen am TLS-Client

Um eine gesicherte TCP-Verbindung im TLS-Client einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_IP_V4_SEC. Geben Sie dazu im Feld "Datentyp" die Zeichenfolge "TCON_IP_V4_SEC" ein.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "SEC connection 1 TLS-Client" vom Datentyp TCON_IP_V4_SEC definiert ist.

Die Interfaceld hat den Wert der HW-Kennung der IE-Schnittstelle des lokalen CP (TLS-Client).

Data_block_1				
	Name	Data type	Start value	Comment
1	Static			
2	SEC connection 1 TLS-Client	TCON_IP_V4_SEC		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	Interfaceld	HW_ANY	258	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connction: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	100	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against th
19	TLSServerCertRef	UDInt	1	for Server side: Reference to own X.509 V3 server certificate; fo
20	TLSClientCertRef	UDInt	5	for Client side: add id of own X.509 V3 client certificate; for Sen

Bild 6-23 IP_V4_SEC_Client

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "RemoteAddress" die IPv4-Adresse des TLS-Servers ein.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein.
 - "ActivateSecureConn": Aktivierung von Secure Communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine ungesicherte TCP- oder UDP-Verbindung einrichten.
 - "TLSServerCertRef": Tragen Sie den Wert 2 (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA256) bzw. den Wert 1 ein (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA1)). Wenn Sie ein anderes CA-Zertifikat nutzen, tragen Sie die entsprechende ID aus dem Zertifikatsmanager der Globalen Security-Einstellungen ein.
 - "TLSClientCertRef": ID des eigenen X.509-V3-Zertifikats.
5. Legen Sie im Programmeditor eine der Anweisungen TSEND_C, TRCV_C oder TCON an.
6. Verschalten Sie den Parameter CONNECT einer der Anweisungen TSEND_C, TRCV_C oder TCON mit der Variablen vom Datentyp TCON_IP_V4_SEC.

Einstellungen am TLS-Server

Um eine gesicherte TCP-Verbindung im TLS-Server einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_IP_V4_SEC.

Das folgende Beispiel zeigt den globalen Datenbaustein "Data_block_1", in dem die Variable "SEC connection 1 TLS-Server" vom Datentyp TCON_IP_V4_SEC definiert ist.

Die Interfaceld hat den Wert der HW-Kennung der IE-Schnittstelle des lokalen CP (TLS-Server).

Data_block_1				
	Name	Data type	Start value	Comment
1	▼ Static			
2	▼ SEC connection 1 TLS-Server	TCON_IP_V4_SEC		
3	▼ ConnPara	TCON_IP_v4		parameter of the TCP connection
4	Interfaceld	HW_ANY	260	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	10	connection reference / identifier
6	ConnectionType	Byte	11	type of connction: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	false	active/passive connection establishment
8	▼ RemoteAddress	IP_V4		remote IP address (IPv4)
9	▼ ADDR	Array[1..4] of Byte		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	1	IPv4 address
13	ADDR[4]	Byte	10	IPv4 address
14	RemotePort	UInt	4711	remote UDP/TCP port number
15	LocalPort	UInt	4711	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSServerReqClientCert	Bool	true	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	16#0	Bit 0: Just for client side: validate given IPv4 address against the
19	TLSServerCertRef	UDInt	6	for Server side: Reference to own X.509 V3 server certificate; for
20	TLSClientCertRef	UDInt	1	for Client side: add id of own X.509 V3 client certificate; for Serv

Bild 6-24 IP_V4_SEC_Server

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "RemoteAddress" die IPv4-Adresse des TLS-Clients ein.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein.
 - "ActivateSecureConn": Aktivierung von Secure Communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine unsichere TCP- oder UDP-Verbindung einrichten.
 - "TLSServerReqClientCert ": Anforderung eines X.509-V3-Zertifikats vom TLS-Client. Tragen Sie den Wert "true" ein.
 - "TLSServerCertRef": ID des eigenen X.509-V3-Zertifikats.
 - "TLSClientCertRef": Tragen Sie den Wert 2 (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA256) bzw. den Wert 1 ein (Referenz auf das CA-Zertifikat des TIA Portal-Projekts (SHA1)). Wenn Sie ein anderes CA-Zertifikat nutzen, tragen Sie die entsprechende ID aus dem Zertifikatsmanager der Globalen Security-Einstellungen ein.
5. Legen Sie im Programmeditor eine Anweisung TSEND_C, TRCV_C oder TCON an.
6. Verschalten Sie den Parameter CONNECT der Anweisung TSEND_C, TRCV_C oder TCON mit der Variablen vom Datentyp TCON_IP_V4_SEC.

Laden des Geräts als neue Station

Wenn Sie eine Konfiguration mit Zertifikaten und projektierter Secure Open User Communication als neue Station in Ihr STEP 7-Projekt hochladen, dann werden die Zertifikate des CP im Gegensatz zu den Zertifikaten der CPU nicht mit hochgeladen. Nach dem Laden des Geräts als neue Station sind keine Zertifikate in den entsprechenden Tabellen der CPs für die Gerätezertifikate mehr enthalten!

Sie müssen die Projektierung der Zertifikate nach dem Hochladen erneut durchführen. Andernfalls führt ein erneutes Laden der Konfiguration dazu, dass die ursprünglich im CP vorhandenen Zertifikate gelöscht werden und die Secure Communication nicht funktioniert.

Secure OUC-Verbindungen über CPU- und CP-Schnittstellen - Gemeinsamkeiten

- Verbindungsressourcen:
Keine Unterschiede zwischen OUC und Secure OUC. Eine programmierte Secure OUC-Verbindung verbraucht ebenso eine Verbindungsressource wie eine OUC-Verbindung, unabhängig davon, über welche IE-/PROFINET-Schnittstelle die Station kommuniziert.
- Verbindungsdiagnose:
Keine Unterschiede zwischen OUC und Secure OUC-Verbindungsdiagnose.
- Laden von Projekten mit Secure OUC-Verbindungen in die CPU:
Nur im STOP der CPU möglich, falls Zertifikate mitgeladen werden.
Empfehlung: Laden in Gerät > Hardware und Software. Grund: Sicherstellen der Konsistenz zwischen Programm mit Secure OUC, Hardware-Konfiguration und Zertifikaten.
Zertifikate werden mit der Hardware-Konfiguration geladen - daher erfordert das Laden ein Stoppen der CPU. Das Nachladen von Bausteinen, die weitere Secure OUC-Verbindungen nutzen, ist nur dann im RUN möglich, wenn sich die dafür erforderlichen Zertifikate bereits auf der Baugruppe befinden.

6.11.5 Secure OUC mit Modbus TCP

Für die gesicherte Modbus TCP-Verbindung müssen Sie selbst einen Datenbaustein mit einem der Systemdatentypen TCON_IP_V4_SEC oder TCON_QDN_SEC erstellen, parametrieren und direkt an der Anweisung MB_Server bzw. MB_CLIENT aufrufen.

Voraussetzungen

- S7-1500 CPU ab Firmware Version V2.5
- Der Modbus-Client (TLS-Client) kann den Modbus-Server (TLS-Server) über IP-Kommunikation im Netzwerk erreichen.
- TLS-Client und TLS-Server besitzen alle notwendigen Zertifikate

Beispiel Gesicherte Modbus TCP-Verbindung zu einem Modbus TCP-Server einrichten

Im Folgenden ist beschrieben, wie Sie eine Secure Open User Communication über Modbus TCP von einem Modbus TCP-Client zu einem Modbus TCP-Server einrichten.

Um eine gesicherte Verbindung von einem Modbus TCP-Client (TLS-Client) zu einem Modbus TCP-Server (TLS-Server) einzurichten IPv4-Adresse des Mailservers einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TCON_IP_V4_SEC.

Data_block_1				
	Name	Data type	Start value	Comment
1	Static			
2	SEC_ModbusTCP_1	TCON_IP_V4...		
3	ConnPara	TCON_IP_v4		parameter of the TCP connection
4	Interfaceld	HW_ANY	64	HW-identifier of IE-interface submodule
5	ID	CONN_OUC	15	connection reference / identifier
6	ConnectionType	Byte	11	type of connection: 11=TCP/IP, 19=UDP (17=TCP/IP)
7	ActiveEstablished	Bool	true	active/passive connection establishment
8	RemoteAddress	IP_V4		remote IP address (IPv4)
9	ADDR	Array[1..4] of B...		IPv4 address
10	ADDR[1]	Byte	192	IPv4 address
11	ADDR[2]	Byte	168	IPv4 address
12	ADDR[3]	Byte	10	IPv4 address
13	ADDR[4]	Byte	100	IPv4 address
14	RemotePort	UInt	502	remote UDP/TCP port number
15	LocalPort	UInt	502	local UDP/TCP port number
16	ActivateSecureConn	Bool	true	activate the security functionality of that connection in general
17	TLSReqClientCert	Bool	false	Just for server side: The TLS server requests a client certificate
18	ExtTLSCapabilities	Word	0	Bit 0: Just for client side: validate given IPv4 address against the subjectAlt
19	TLSReqServerCert	UDInt	2	for Server side: Reference to own X.509 V3 server certificate; for Client side:
20	TLSClientCertRef	UDInt	7	for Client side: add id of own X.509 V3 client certificate; for Server side: add

Bild 6-25 TCON_IP_V4_SEC

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "MailServerAddress" die IPv4-Adresse des Mailservers ein.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein. Tragen Sie z. B. bei "TLSServerCertRef" die Zertifikat-ID vom CA-Zertifikat des Kommunikationspartners ein.
 - "ActivateSecureConn": Aktivierung von secure communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine non-secure Modbus TCP-Verbindung einrichten.
 - "TLSServerCertRef": Referenz auf das X.509 V3 (CA)-Zertifikat des Modbus TCP-Servers, welches vom TLS Client benutzt wird, um die Authentifizierung des Modbus TCP-Servers zu validieren.
5. Legen Sie im Programmeditor eine Anweisung MB_Client an.
6. Verschalten Sie den Parameter CONNECT der Anweisung MB_Client mit der Variablen vom Datentyp TCON_IP_V4_SEC.

6.11.6 Secure OUC über E-Mail

Gesicherte Verbindung zu einem Mailserver über die Schnittstelle der CPU

Für die gesicherte Verbindung zu einem Mailserver müssen Sie selbst einen Datenbaustein mit einem der Systemdatentypen TMAIL_V4_SEC, TMAIL_QDN_SEC erstellen, parametrieren und direkt an der Anweisung TMAIL_C aufrufen.

Voraussetzungen

- Anweisung TMAIL_C ab Anweisungsversion V5.0
- STEP 7 ab V15
- CPU S7-1500 ab V2.5
- Sie haben der CPU (TLS-Client) alle CA-Zertifikate des Mailservers (TLS-Servers) zugewiesen und die Konfiguration in die CPU geladen.
- Aktuelles Datum und Uhrzeit sind in der CPU eingestellt.

Verfahren zum Aufbau der gesicherten Verbindung zum Mailserver

Sie haben zwei Verfahren zur Auswahl, wie die gesicherte Verbindung zum Mailserver aufgebaut wird:

- SMTPS: Der Client versucht sofort eine TLS-Verbindung zum Mailserver herzustellen ("Handshake"-Verfahren). Wenn der Mailserver TLS nicht unterstützt, dann kommt keine Verbindung zustande.
- STARTTLS: Client baut eine TCP-Verbindung zum Mailserver auf. Über die TCP-Verbindung sendet der Client eine Anfrage zum "Upgrade" der bestehenden Verbindung zu einer gesicherten TLS-Verbindung. Unterstützt der Mailserver TLS, dann sendet er dem Client den Befehl zum Aufbau einer gesicherten Verbindung. Der Mailserver nutzt dazu den SMTP-Befehl "STARTTLS". Der Client baut daraufhin eine gesicherte Verbindung zum Mailserver auf. Vorteil: Wenn der Mailserver kein TLS unterstützt, dann können Client und Mailserver ungesichert miteinander kommunizieren.

Welches Verfahren Sie für die Kommunikation verwenden, legen Sie über die Einstellung "Remote Port" im Datentypen am Bausteinparameter "MAIL_ADDR_PARAM" fest.

Tabelle 6- 6 Portnummern für die Verfahren SMTPS und STARTTLS

Verfahren	Port
SMTPS	465 ¹
STARTTLS	beliebig (#465) ²

¹ Die Anweisung TMAIL_C verwendet nur für Port 465 SMTPS. Für alle anderen Ports wird STARTTLS verwendet.

² gemäß RFC nutzen Mailserver die Ports 25 und 587 für gesicherte Verbindungen mit STARTTLS. Die Verwendung anderer Portnummern für SMTP ist nicht RFC-konform, erfolgreiche Kommunikation mit einem solchen Mailserver ist nicht garantiert.

Beispiel: Gesicherte Verbindung zu einem Mailserver einrichten über IPv4

Im Folgenden ist beschrieben, wie Sie mit der Kommunikationsanweisung TMAIL_C eine gesicherte Verbindung zu einem IPv4-Mailserver einrichten.

Um eine gesicherte Verbindung über die IPv4-Adresse des Mailservers einzurichten, gehen Sie folgendermaßen vor:

1. Legen Sie in der Projektnavigation einen globalen Datenbaustein an.
2. Definieren Sie im globalen Datenbaustein eine Variable vom Datentyp TMAIL_V4_SEC.

Das folgende Beispiel zeigt den globalen Datenbaustein "MailConnDB", in dem die Variable "MailConnectionSEC" vom Datentyp TMAIL_V4_SEC definiert ist.

MailConnDB				
Name	Datentyp	Startwert	Kommentar	
Static				
MailConnectionSEC	TMail_V4_SEC			
Interfaceld	HW_ANY	0	Use HW-identifier of the IE-interface to specify the module which	
ID	CONN_OUC	0	connection reference / identifier	
ConnectionType	Byte	16#20	type of connection 16#20=32=TMail_V4 or TMail_V4_SEC	
ActiveEstablished	Bool	true	active / passive connection establishment	
WatchDogTime	Time	T#5000ms	watchdog time to monitor SMTP server association (time duration	
MailServerAddress	IP_V4		IPv4 address of mail server	
ADDR	Array[1..4] of Byte		IPv4 address	
ADDR[1]	Byte	144	IPv4 address	
ADDR[2]	Byte	145	IPv4 address	
ADDR[3]	Byte	2	IPv4 address	
ADDR[4]	Byte	20	IPv4 address	
UserName	String[254]	'MyName'	user name which is necessary to login into the user's mail account	
PassWord	String[254]	'MyPW'	user password which is necessary to login into the user's mail acc	
From	EMAIL_ADDR		source mail address	
LocalPartPlusAtSign	String[64]	'Mustermann@'	local part of e-mail address plus "@" sign	
FullQualifiedDomainName	String[254]	'siemens.com'	full qualified domain name part of e-mail address	
RemotePort	UInt	587	remote TCP port number	
ActivateSecureConn	Bool	true	activate the security functionality of that connection in general	
ExtTLSCapabilities	Byte	16#0	for further capability extensions of the TLS handshake protocol	
TLSServerCertRef	UDInt	7	Reference to the X.509 V3 (CA-) certificate of the mail server	

Bild 6-26 Datentyp TMAIL_V4_SEC

3. Stellen Sie die Verbindungsparameter der TCP-Verbindung in der Spalte "Startwert" ein. Tragen Sie z. B. bei "MailServerAdress" die IPv4-Adresse des Mailservers ein.

Hinweis

Verbindungsparameter Interfaceld und ID

Beachten Sie, daß Sie ab Anweisungsversion V5.0 der Anweisung TMAIL_C im Datentyp TMAIL_V4_SEC den Wert "0" für die Interfaceld und die ID eintragen können. In diesem Fall sucht die CPU selbst nach einer passenden lokalen Schnittstelle der CPU bzw. einer freien Verbindungs-ID.

4. Stellen Sie die Parameter für Secure Communication in der Spalte "Startwert" ein. Tragen Sie z. B. bei "TLSServerCertRef" die Zertifikat-ID vom CA-Zertifikat des Kommunikationspartners ein.
 - "ActivateSecureConn": Aktivierung von secure communication für diese Verbindung. Falls dieser Parameter den Wert FALSE hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine non-secure TCP- oder UDP-Verbindung einrichten.
 - "TLSServerCertRef": Referenz auf das X.509 V3 (CA)-Zertifikat des Mail Servers, welches vom TLS Client benutzt wird, um die Authentifizierung des Mailservers zu validieren.
5. Legen Sie im Programmeditor eine Anweisung TMAIL_C an.
6. Verschalten Sie den Parameter MAIL_ADDR_PARAM der Anweisung TMAIL_C mit der Variablen vom Datentyp TMAIL_V4_SEC.

Im folgenden Beispiel ist der Parameter Mail_ADDR_PARAM der Anweisung TMAIL_C mit der Variablen "MailConnectionSEC" (Datentyp TMAIL_V4_SEC) verschaltet.

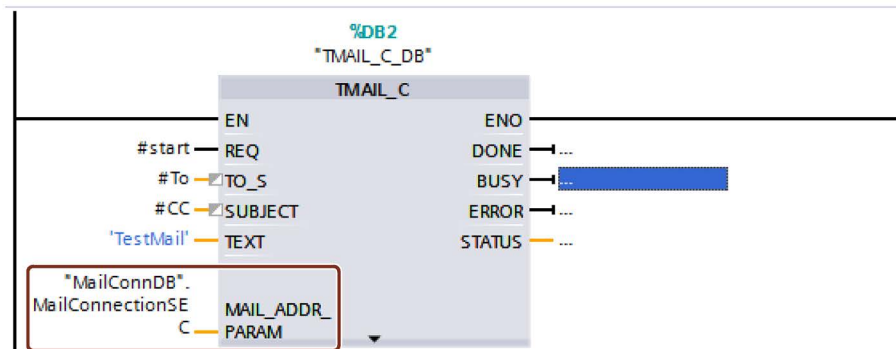


Bild 6-27 Anweisung TMAIL_C

Gesicherte Verbindung zu einem Mailserver über die Schnittstelle eines Kommunikationsmoduls

Für die gesicherte Verbindung zu einem Mailserver über ein Kommunikationsmodul müssen Sie selbst einen Datenbaustein mit einem der Systemdatentypen TMAIL_V4_SEC, TMAIL_QDN_SEC oder TMAIL_V6_SEC (nur CP) erstellen, parametrieren und direkt an der Anweisung TMAIL_C aufrufen.

Voraussetzungen:

- Anweisung TMAIL_C mit Anweisungsversion **V4.0**
- S7-1500 CPU ab Firmwarestand V2.0 mit Kommunikationsmodul CP 1543-1 ab Firmwarestand V2.0
- ET 200SP CPU ab Firmwarestand V2.0 mit Kommunikationsmodul CP 1542SP-1 (IRC) ab Firmwarestand V1.0
- Sie haben dem CP (TLS-Client) alle CA-Zertifikate des Mailservers (TLS-Servers) zugewiesen und die Konfiguration in die CPU geladen.
- Aktuelles Datum und Uhrzeit sind in der CPU eingestellt.

Wie Sie die Verbindung gesicherte Verbindung zu einem Mailserver über die Schnittstelle eines Kommunikationsmoduls festlegen, finden Sie in der Onlinehilfe zu STEP 7 beschrieben.

Anwendungsbeispiel

Wie Sie über den CP einer S7-1500 oder S7-1200 Station eine gesicherte Verbindung zu einem E-Mail-Server einrichten und mit der Standard-Anweisung "TMAIL_C" aus der S7-CPU eine E-Mail verschicken, finden Sie in diesem Anwendungsbeispiel (<https://support.industry.siemens.com/cs/ww/de/view/46817803>).

Weitere Informationen

Weitere Informationen zu den Systemdatentypen TMail_V4_SEC und TMAIL_QDN_SEC finden Sie in der Onlinehilfe zu STEP 7.

Weitere Informationen zur sicheren Kommunikation finden Sie im Kapitel Secure Communication (Seite 38).

S7-Kommunikation

Merkmale S7-Kommunikation

Die S7-Kommunikation als SIMATIC-homogene Kommunikation zeichnet sich aus durch herstellerspezifische Kommunikation zwischen SIMATIC-CPU's (kein offener Standard). Die S7-Kommunikation dient der Migration und Anbindung an bestehende Systeme (S7-300, S7-400).

Für die Datenübertragung zwischen zwei Automatisierungssystemen S7-1500 empfehlen wir Ihnen, die offene Kommunikation zu nutzen (siehe Kapitel Open User Communication (Seite 70)).

Eigenschaften der S7-Kommunikation

Über S7-Kommunikation tauscht die CPU Daten mit einer weiteren CPU aus. Sobald der Anwender die Daten auf der Empfängerseite empfangen hat, wird der Empfang der Daten automatisch an die Sende-CPU quittiert.

Der Datenaustausch erfolgt über projektierte S7-Verbindungen. S7-Verbindungen können einseitig oder zweiseitig projektiert werden.

S7-Kommunikation ist möglich über:

- Integrierte PROFINET- oder PROFIBUS DP-Schnittstelle einer CPU
- Schnittstelle eines CP/CM

Einseitig projektierte S7-Verbindungen

Bei einer einseitig projektierten S7-Verbindung erfolgt die Projektierung für diese Verbindung nur in einem Kommunikationspartner und wird auch nur in diesen geladen.

Eine einseitige S7-Verbindung kann zu einer CPU projektiert werden, die nur Server einer S7-Verbindung ist (z. B. CPU 315-2 DP). Die CPU ist projektiert und damit sind die Adressparameter und Schnittstellen bekannt.

Außerdem kann eine einseitige S7-Verbindung zu einem Partner projektiert werden, der nicht im Projekt vorhanden ist und dessen Adressparameter und Schnittstelle daher nicht bekannt sind. Die Adresse müssen Sie eingeben; sie wird von STEP 7 nicht überprüft. Der Partner ist initial unspezifiziert (beim Anlegen der S7-Verbindung ist noch keine Partneradresse eingetragen). Sobald Sie die Adresse eingeben, ist er "unbekannt" (das heißt: er ist spezifiziert, aber dem Projekt unbekannt).

Damit besteht die Möglichkeit, über Projektgrenzen hinweg S7-Verbindungen einzusetzen. Der Kommunikationspartner ist für das lokale Projekt unbekannt (unspezifiziert) und wird in einem anderen STEP 7- oder Fremd-Projekt projektiert.

Zweiseitig projektierte S7-Verbindungen

Bei einer zweiseitig projektierten S7-Verbindung erfolgt die Projektierung und das Laden der projektierten S7-Verbindungsparameter in beide Kommunikationspartner.

Anweisungen für S7-Kommunikation

Für S7-Kommunikation in S7-1500 sind folgende Anweisungen einsetzbar:

- **PUT/GET**

Mit der Anweisung PUT schreiben Sie Daten in eine remote CPU. Mit der Anweisung GET lesen Sie Daten aus einer remoten CPU aus. Die Anweisungen PUT und GET sind einseitige Anweisungen, d. h. Sie benötigen nur Anweisung in einem Kommunikationspartner. Die Anweisungen PUT und GET können Sie bequem über die Verbindungsparametrierung einrichten.

Hinweis

Datenbausteine für Anweisungen PUT/GET

Bei Verwendung der Anweisungen PUT/GET dürfen Sie nur Datenbausteine mit absoluter Adressierung einsetzen. Symbolische Adressierung von Datenbausteinen ist nicht möglich.

Desweiteren müssen Sie diesen Service im Bereich "Schutz" in der CPU-Projektierung freischalten.

Wie Sie für den Datenaustausch zwischen zwei S7-1500 CPUs eine S7-Verbindung und die Kommunikationsanweisungen PUT und GET projektieren und programmieren, finden Sie in diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/82212115>).

- **BSEND/BRCV**

Die Anweisung BSEND sendet Daten an eine remote Partneranweisung vom Typ BRCV. Die Anweisung BRCV empfängt Daten von einer remoten Partneranweisung vom Typ BSEND. Die S7-Kommunikation über das Anweisungspaar BSEND/BRCV verwenden Sie für sicheres Übertragen von Daten.

- **USEND/URCV**

Die Anweisung USEND sendet Daten an eine remote Partneranweisung vom Typ URCV. Die Anweisung URCV empfängt Daten von einer remoten Partneranweisung vom Typ USEND. Die S7-Kommunikation über das Anweisungspaar USEND/URCV verwenden Sie für schnelles, ungesichertes Übertragen von Daten unabhängig von der zeitlichen Bearbeitung des Kommunikationspartners; z. B. für Betriebs- und Wartungsmeldungen.

S7-Kommunikation über PROFIBUS DP-Schnittstelle im Slave-Betrieb

Sie finden in STEP 7, in den Eigenschaften der PROFIBUS DP-Schnittstelle von Kommunikationsmodulen (z. B. CM 1542-5) das Kontrollkästchen "Test, Inbetriebnahme und Routing". Über dieses Kontrollkästchen stellen Sie ein, ob die PROFIBUS DP-Schnittstelle des DP-Slaves aktiver oder passiver Teilnehmer am PROFIBUS ist.

- Kontrollkästchen aktiviert: DP-Slave ist aktiver Teilnehmer am PROFIBUS.
- Kontrollkästchen deaktiviert: DP-Slave ist passiver Teilnehmer am PROFIBUS. Zu diesem DP-Slave können Sie nur einseitig projektierte S7-Verbindungen einrichten.

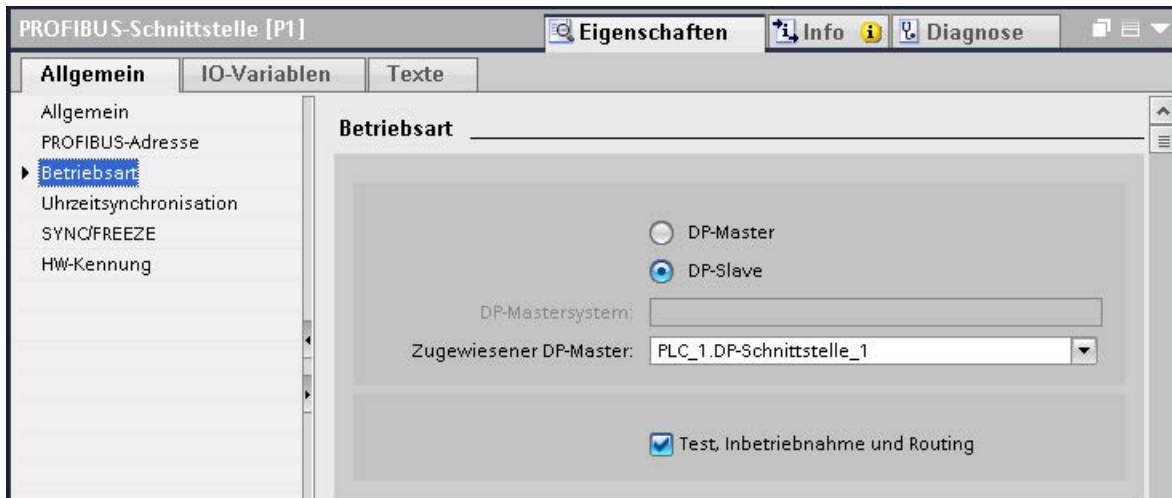


Bild 7-1 Kontrollkästchen "Test, Inbetriebnahme und Routing"

S7-Verbindungen für PUT/GET-Anweisungen parametrieren

In der Verbindungsparametrierung der PUT/GET-Anweisungen können Sie S7-Verbindungen anlegen und parametrieren. Geänderte Werte werden von der Verbindungsparametrierung sofort auf Eingabefehler geprüft.

Voraussetzung: Im Programmiereditor ist eine Anweisung PUT bzw. GET angelegt.

Um eine S7-Verbindung über PUT/GET-Anweisungen zu projektieren, gehen Sie folgendermaßen vor:

1. Selektieren Sie im Programmiereditor den Aufruf der Anweisung PUT oder GET.
2. Öffnen Sie im Inspektorfenster das Register "Eigenschaften > Konfiguration".

3. Selektieren Sie die Gruppe "Verbindungsparameter". Solange Sie noch keinen Verbindungspartner selektiert haben, ist nur die leere Klappliste für den Partner-Endpoint aktiv. Alle anderen Eingabemöglichkeiten sind deaktiviert.

Es werden die bereits bekannten Verbindungsparameter angezeigt:

- Name des lokalen Endpunkts
- Schnittstelle des lokalen Endpunkts
- IPv4-Adresse des lokalen Endpunkts

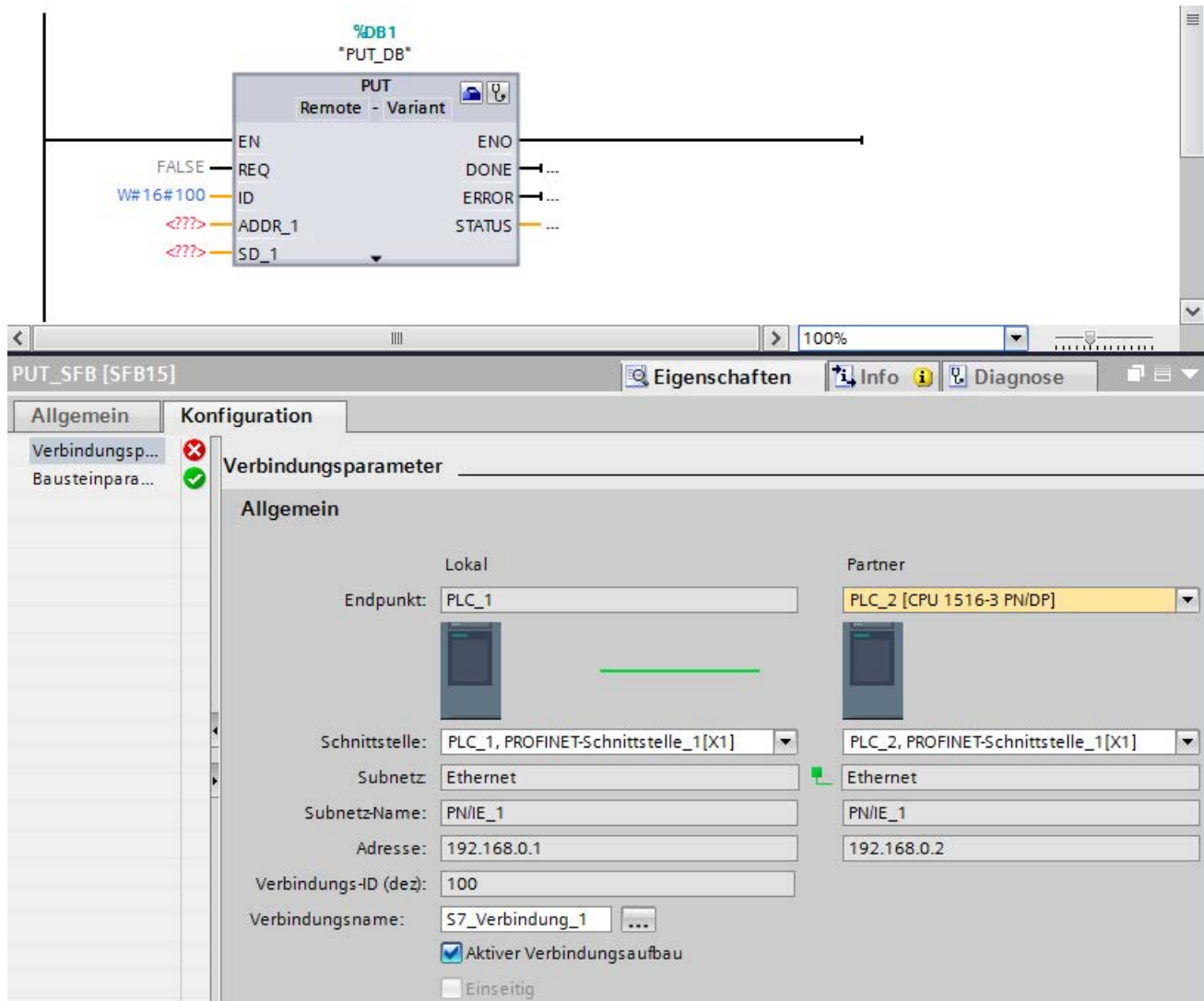


Bild 7-2 Verbindungsparametrierung für PUT-Anweisung

4. Wählen Sie in der Klappliste des Partner-Endpunkts einen Verbindungspartner. Als Kommunikationspartner kommt ein unspezifiziertes Gerät oder eine im Projekt vorhandene CPU in Frage.

Die folgenden Parameter werden automatisch eingetragen, sobald Sie den Verbindungspartner gewählt haben:

- Name des Partner-Endpunkts
 - Schnittstelle des Partner-Endpunkts. Wenn mehrere Schnittstellen zur Verfügung stehen, dann können Sie die Schnittstelle bei Bedarf ändern.
 - Schnittstellentyp des Partner-Endpunkts
 - Subnetz-Name beider Endpunkte
 - IPv4-Adresse des Partner-Endpunkts
 - Name der Verbindung, die für die Kommunikation genutzt wird.
5. Ändern Sie bei Bedarf den Verbindungsnamen im Eingabefeld "Verbindungsname" ab. Wenn Sie eine neue Verbindung erstellen, oder eine vorhandene Verbindung bearbeiten möchten, klicken Sie auf die Schaltfläche "Verbindung auswählen" rechts neben dem Eingabefeld für den Verbindungsnamen.

Hinweis

Die Anweisungen PUT und GET zwischen zwei Kommunikationspartnern sind erst dann lauffähig, wenn sowohl die Hardware-Konfiguration wie auch der Programmteil für den Partner-Endpunkt in die Hardware geladen wurden. Achten Sie darauf, dass Sie für eine funktionierende Kommunikation nicht nur die Verbindungsbeschreibung der lokalen CPU in das Gerät laden, sondern auch die der Partner-CPU.

S7-Verbindungen für z. B. BSEND/BRCV projektieren

Wenn Sie z. B. die Anweisungen für BSEND/BRCV für S7-Kommunikation nutzen wollen, müssen Sie zunächst eine S7-Verbindung projektieren.

Um eine S7-Verbindung zu projektieren, gehen Sie folgendermaßen vor:

1. Konfigurieren Sie in der Netzsicht des Hardware- und Netzwerkeditors von STEP 7 die Kommunikationspartner.
2. Wählen Sie die Schaltfläche "Verbindungen" und aus der Klappliste den Eintrag "S7-Verbindung".
3. Verbinden Sie per Drag & Drop die Kommunikationspartner miteinander (über Schnittstelle oder lokalen Endpunkt). Falls noch kein zugehöriges S7-Subnetz vorhanden ist, wird dieses automatisch angelegt.

Alternativ können Sie auch eine Verbindung zu unspezifizierten Partnern einrichten.

4. Wählen Sie im Register "Verbindungen" die Zeile der S7-Verbindung.

5. Stellen Sie im Bereich "Allgemein", im Register "Eigenschaften" ggf. die Eigenschaften der S7-Verbindung ein, z. B. den Namen der Verbindung und die verwendeten Schnittstellen der Kommunikationspartner.

Für S7-Verbindungen zu einem unspezifizierten Partner stellen Sie die Adresse des Partners ein.

Im Bereich "Lokale ID" finden Sie die lokale ID (Referenz der S7-Verbindung im Anwenderprogramm).

6. Wählen Sie in der Projektnavigation für eine der beiden CPUs den Ordner "Programmbausteine" und öffnen Sie im Ordner den OB 1 durch Doppelklick. Der Programmeditor öffnet sich.
7. Rufen Sie im Programmeditor die entsprechenden Anweisungen für S7-Kommunikation im Anwenderprogramm des Kommunikationspartners (einseitig) bzw. in den Anwenderprogrammen der Kommunikationspartner (zweiseitig) auf. Wählen Sie z. B. aus der Task Card "Anweisungen", Bereich "Kommunikation" die Anweisungen BSEND und BRCV und ziehen Sie sie per Drag & Drop in ein Netzwerk des OB 1.
8. Vergeben Sie am Parameter ID der Anweisung die lokale ID der projektierten Verbindung, die für die Übertragung der Daten verwendet werden soll.
9. Parametrieren Sie die Anweisungen, welche Daten wohin geschrieben bzw. woher gelesen werden.
10. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU(s).

S7-Kommunikation über CP 1543-1

Wenn Sie die S7-Kommunikation über die Industrial Ethernet-Schnittstelle des CP 1543-1 einrichten, können Sie in den Eigenschaften der S7-Verbindung unter "Allgemein" das Transportprotokoll für die Datenübertragung wählen:

- Kontrollkästchen "TCP/IP" aktiviert (voreingestellt): ISO-on-TCP (RFC1006): für die S7-Kommunikation zwischen CPUs S7-1500
- Kontrollkästchen "TCP/IP" deaktiviert: ISO-Protokoll (IEC8073): Adressierung über MAC-Adressen

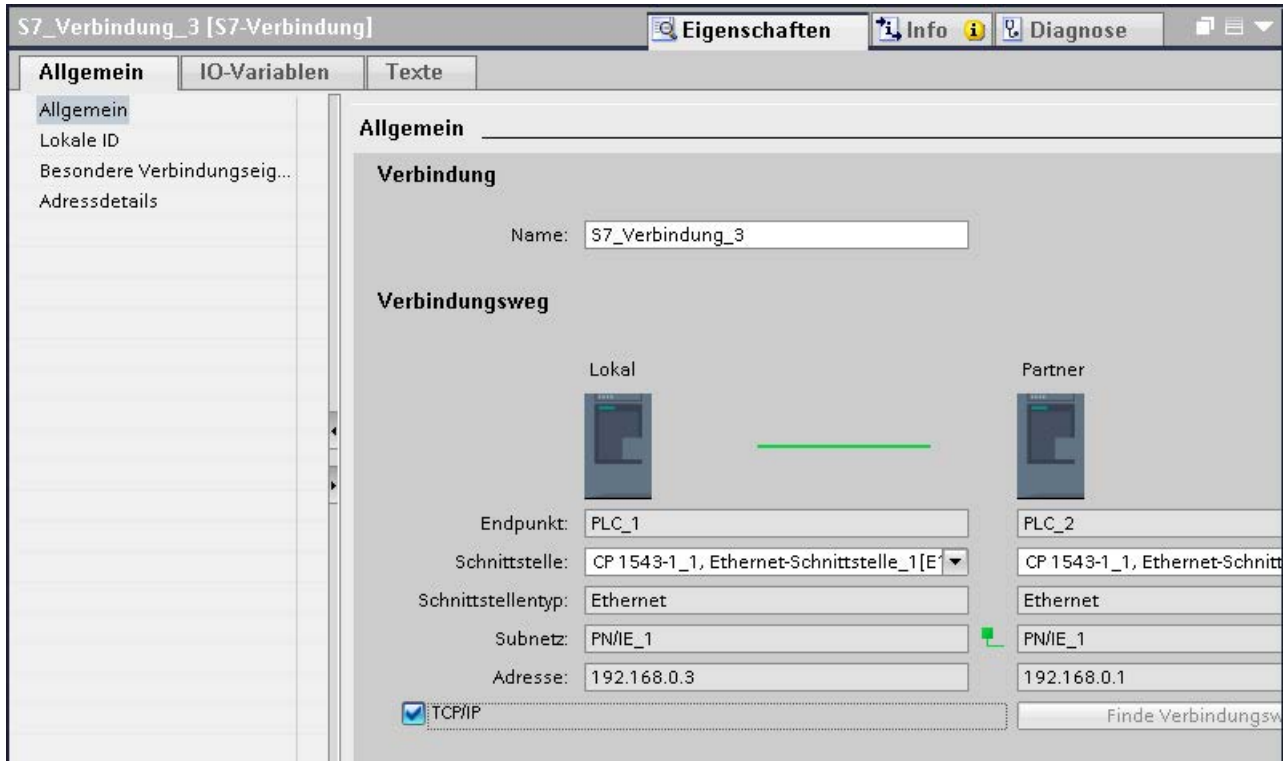


Bild 7-3 CP 1543-1 Transportprotokoll wählen

Vorgehen zum Einrichten einer S7-Verbindung über unterschiedliche S7-Subnetze

Sie haben die Möglichkeit eine S7-Verbindung über mehrere S7-Subnetze (PROFIBUS, PROFINET/Industrial Ethernet) hinweg zu nutzen (S7-Routing (Seite 235)).

1. Konfigurieren Sie in der Netzsicht des Hardware- und Netzwerkeditors von STEP 7 die Kommunikationspartner.
2. Wählen Sie die Schaltfläche "Vernetzen".
3. Verbinden Sie per Drag & Drop die entsprechenden Schnittstellen mit den jeweiligen S7-Subnetzen (PROFIBUS oder PROFINET / Industrial Ethernet).

4. Wählen Sie die Schaltfläche "Verbindungen" und aus der Klappliste den Eintrag "S7-Verbindung".
5. Verbinden Sie per Drag & Drop in unserem Beispiel die PLC_1 im linken S7-Subnetz (PROFIBUS) mit der PLC_3 rechten S7-Subnetz (PROFINET).

Die S7-Verbindung von CPU 1 zu CPU 3 ist konfiguriert.

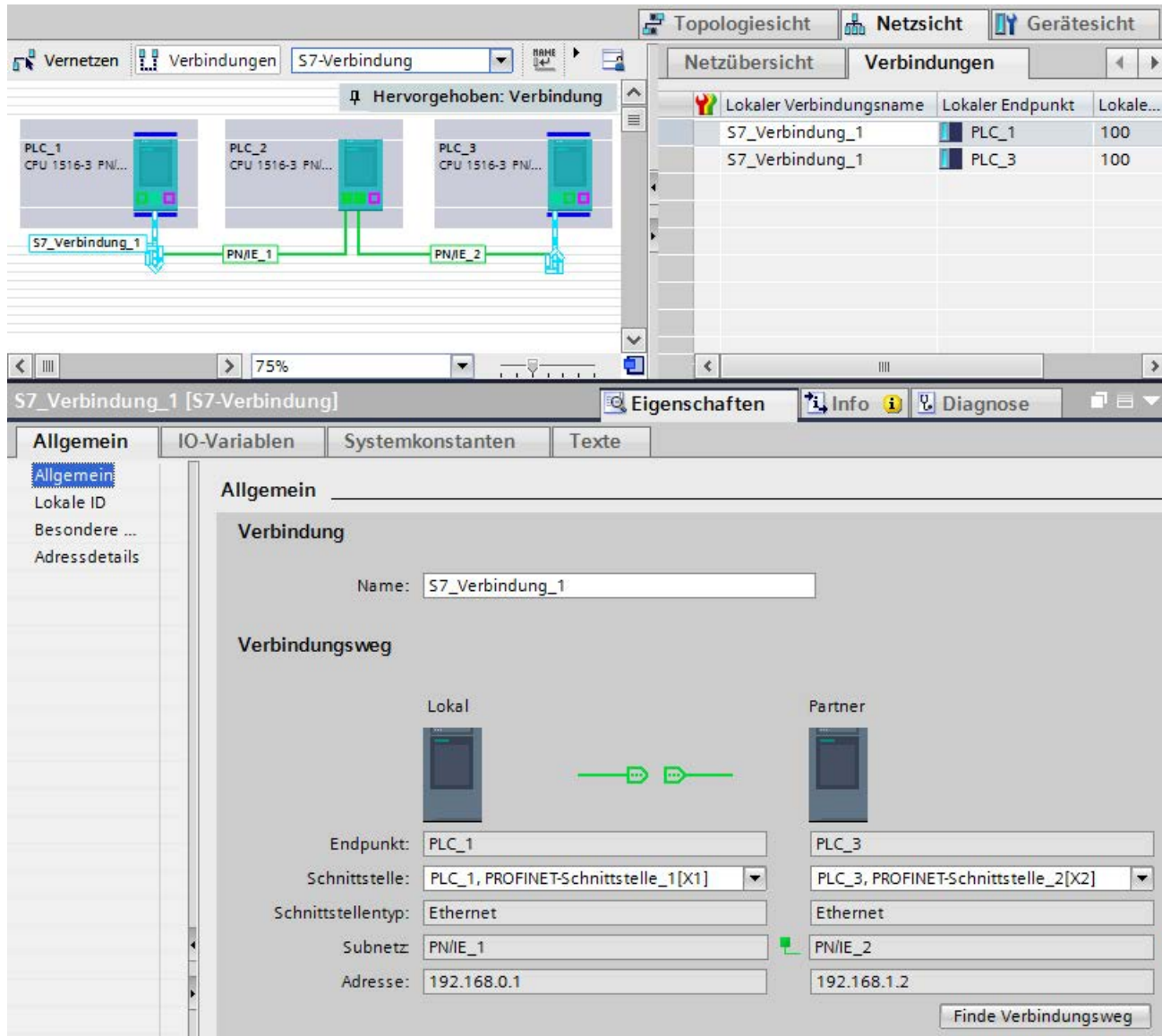


Bild 7-4 S7-Verbindungen über unterschiedliche Subnetze

ET 200SP Open Controller als Router für S7-Verbindungen

Wenn Sie die Schnittstelle "PROFINET onboard [X2]" der CPU 1515SP PC (F) der SIMATIC PC-Station zuweisen, kann die CPU 1515SP PC (F) als Router für S7-Verbindungen verwendet werden. Wenn Sie die CP-Schnittstelle für "Keine oder eine andere Windows-Einstellung" verwenden, dann können Sie den Open Controller nicht als Router für geroutete S7-Verbindungen verwenden.

Eine bestehende durch die CPU 1515SP PC (F) geroutete S7-Verbindung wird ungültig, wenn die Zuweisung der Schnittstelle der CPU 1515SP PC (F) von "SIMATIC PC-Station" zu "Keine oder eine andere Windows-Einstellung" geändert wird. Da die PLC nun keine Routingfunktion mehr für diese Verbindung übernimmt, wird beim Übersetzen der CPU 1515SP PC (F) kein Hinweis auf die ungültige Verbindung angezeigt. Die ungültige geroutete S7-Verbindung wird Ihnen erst beim Übersetzen der Endpunkte der Verbindung angezeigt.

Die für geroutete S7-Verbindungen benötigten Schnittstellen müssen in der CPU 1515SP PC (F) explizit zugewiesen bleiben. Sie können die Zuweisung der Schnittstelle der CPU 1515SP PC (F) in den Eigenschaften unter "PROFINET onboard [X2] > Schnittstellenzuweisung" bearbeiten.

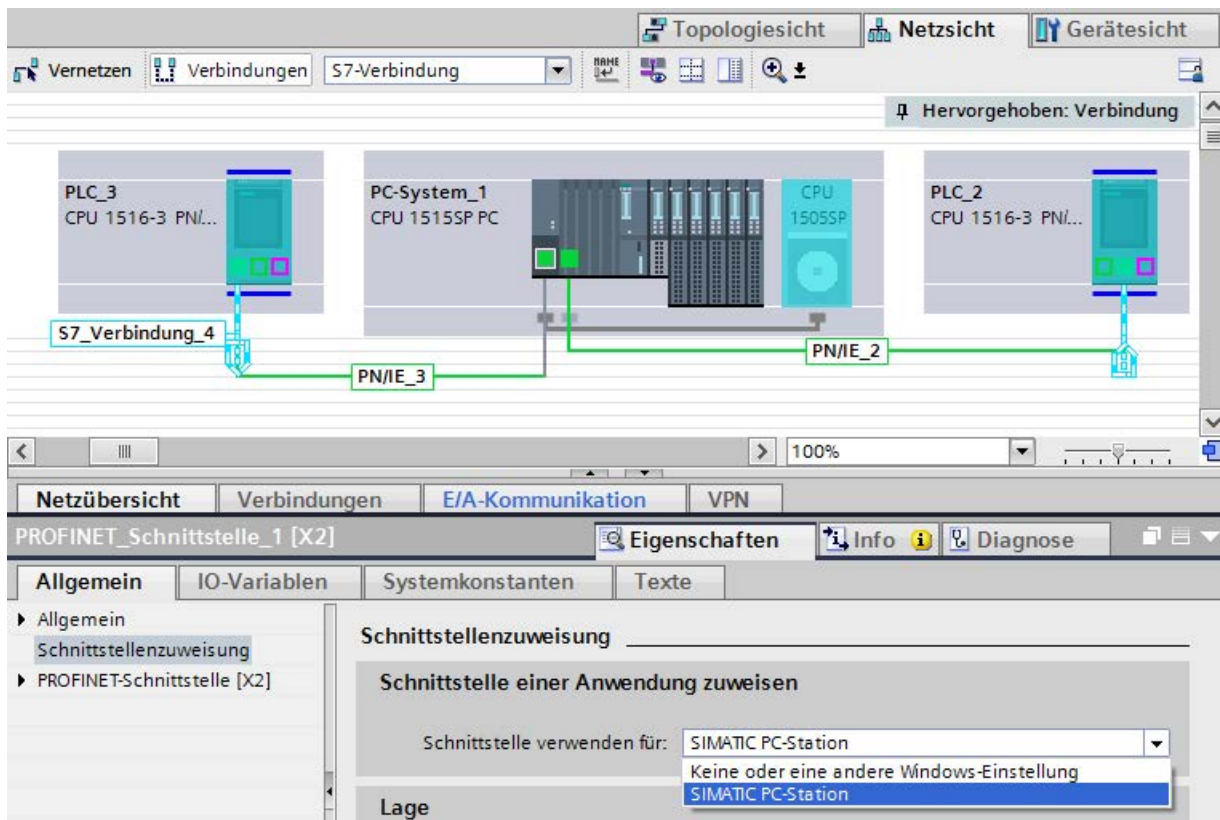


Bild 7-5 S7-Routing PC-Station

Weitere Informationen

Detaillierte Informationen zum Projektieren von S7-Verbindungen und wie Sie Anweisungen für die S7-Kommunikation im Anwenderprogramm nutzen, finden Sie in der Online-Hilfe STEP 7 beschrieben.

Punkt-zu-Punkt-Kopplung

Funktionalität

Die Kommunikation über Punkt-zu-Punkt-Kopplung bei S7-1500, ET 200MP und ET 200SP erfolgt über Kommunikationsmodule (CM) mit seriellen Schnittstellen (RS232, RS422 oder RS485):

- S7-1500/ET 200MP:
 - CM PtP RS232 BA
 - CM PtP RS422/485 BA
 - CM PtP RS232 HF
 - CM PtP RS422/485 HF
- ET 200SP:
 - CM PtP

Der bidirektionale Datenaustausch über Punkt-zu-Punkt-Kopplung funktioniert zwischen Kommunikationsmodulen oder kommunikationsfähigen Fremdsystemen oder -geräten. Zur Kommunikation sind mindestens 2 Kommunikationspartner notwendig ("Punkt zu Punkt"). Bei RS422 und RS485 sind mehr als zwei Kommunikationspartner möglich.

Protokolle für die Kommunikation über Punkt-zu-Punkt-Kopplung

- Freeport-Protokoll (auch ASCII-Protokoll genannt)
- Prozedur 3964(R)
- Modbus-Protokoll im RTU-Format (RTU: Remote Terminal Unit)
- USS-Protokoll (Universelles-serielles-Schnittstellen-Protokoll)

Die Protokolle nutzen unterschiedliche Schichten nach dem ISO/OSI-Referenzmodell:

- Freeport: nutzt Schicht 1 (Bitübertragungsschicht)
- 3964(R), USS und Modbus: nutzen Schicht 1 und 2 (Bitübertragungsschicht und Sicherungsschicht; damit höhere Übertragungssicherheit als bei Freeport). USS und Modbus nutzen zusätzlich Schicht 4.

Eigenschaften des Freeport-Protokolls

- Der Empfänger erkennt das Ende der Datenübertragung über ein parametrierbares Endekriterium (z. B. Ablauf Zeichenverzugszeit, Empfang Endezeichen, Empfang feste Anzahl Daten).
- Der Sender kann nicht erkennen, ob die gesendeten Daten beim Empfänger fehlerfrei angekommen sind.

Eigenschaften Prozedur 3964(R)

- Beim Senden werden den Daten Steuerzeichen hinzugefügt (Start-, Ende- und Blockprüfzeichen). Dabei müssen Sie beachten, dass diese Steuerzeichen nicht als Daten in dem Telegramm vorhanden sind.
- Der Verbindungsauf- und -abbau erfolgt über Steuerzeichen.
- Bei Übertragungsfehlern wird die Datenübertragung automatisch wiederholt.

Datenaustausch über Freeport- bzw. 3964(R)-Kommunikation

Die Sendedaten werden im Anwenderprogramm der zugehörigen CPU in Datenbausteinen (Sendepuffer) abgelegt. Für die Empfangsdaten steht im Kommunikationsmodul ein Empfangspuffer zur Verfügung. Kontrollieren Sie die Eigenschaften des Empfangspuffers und passen Sie diese gegebenenfalls an. In der CPU müssen Sie einen Datenbaustein für den Empfang anlegen.

Im Anwenderprogramm der CPU übernehmen die Anweisungen "Send_P2P" und "Receive_P2P" den Datentransfer zwischen CPU und CM.

Vorgehen zum Einrichten von Freeport- bzw. 3964(R)-Kommunikation

1. Konfigurieren Sie in der Gerätesicht des Hardware- und Netzwerkeditors von STEP 7 einen S7-1500-Aufbau mit CPU und CM.
2. Selektieren Sie die Schnittstelle des CM in der Gerätesicht von STEP 7.
3. Parametrieren Sie die Schnittstelle (z. B. Anschlusskommunikation, Konfiguration des Nachrichtensendens) im Inspektorfenster von STEP 7 unter "Eigenschaften" > "Allgemein".
4. Wählen Sie in der Task Card "Anweisungen" unter "Kommunikation" > "Kommunikationsprozessor" die Anweisungen "Send_P2P" bzw. "Receive_P2P" und ziehen Sie sie die Anweisung per Drag & Drop in das Anwenderprogramm (z. B. in einen FB).
5. Parametrieren Sie die Anweisungen entsprechend Ihren Vorgaben.
6. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Alternative: Dynamische Parametrierung des Kommunikationsmoduls

In bestimmten Anwendungsbereichen ist es vorteilhaft, die Kommunikation dynamisch, d. h. über eine spezifische Applikation programmgesteuert, einzurichten.

Typische Anwendungsfälle finden sich z. B. bei Herstellern von Serienmaschinen. Um ihren Kunden möglichst komfortable Bedienoberflächen anzubieten, passen diese Hersteller die Kommunikationsdienste an die jeweiligen Bedieneingaben an.

Anweisungen für Freeport-Kommunikation

Es stehen Ihnen 3 Anweisungen für die dynamische Projektierung im Anwenderprogramm für Freeport-Kommunikation zur Verfügung. Für alle 3 Anweisungen gilt: die bisher gültigen Konfigurationsdaten werden überschrieben, aber nicht dauerhaft im Zielsystem gespeichert.

- Die Anweisung "Port_Config" dient der programmgesteuerten Konfiguration des entsprechenden Ports des Kommunikationsmoduls.
- Die Anweisung "Send_Config" dient der dynamischen Projektierung von z. B. Zeitabständen und Pausen bei der Übertragung (serielle Übertragungsparameter) für den entsprechenden Port.
- Die Anweisung "Receive_Config" dient der dynamischen Projektierung von z. B. Bedingungen für Anfang und Ende einer zu übertragenden Nachricht (serielle Empfangsparameter) für den entsprechenden Port.

Anweisungen für 3964(R)-Kommunikation

Es stehen Ihnen 2 Anweisungen für die dynamische Projektierung im Anwenderprogramm für 3964(R)-Kommunikation zur Verfügung. Für die Anweisungen gilt: die bisher gültigen Konfigurationsdaten werden überschrieben, aber nicht dauerhaft im Zielsystem gespeichert.

- Die Anweisung "Port_Config" dient der programmgesteuerten Konfiguration des entsprechenden Ports des Kommunikationsmoduls.
- Die Anweisung "P3964_Config" dient der dynamischen Projektierung von Protokollparametern.

Eigenschaften USS-Protokoll

- Einfaches serielles Datenübertragungsprotokoll mit zyklischem Telegrammverkehr im Halbduplexbetrieb, das auf die Anforderungen in der Antriebstechnologie zugeschnitten ist.
- Die Datenübertragung funktioniert nach dem Master-Slave-Prinzip.
 - Der Master hat Zugriff auf die Funktionen des Antriebs und kann u. a. den Antrieb steuern, Statuswerte lesen sowie die Antriebsparameter lesen und schreiben.

Datenaustausch über USS-Kommunikation

Das Kommunikationsmodul ist der Master. Der Master sendet kontinuierlich Telegramme (Auftragstelegramme) an die bis zu 16 Antriebe und erwartet jeweils ein Antworttelegramm vom angesprochenen Antrieb.

Ein Antrieb sendet bei folgenden Bedingungen ein Antworttelegramm:

- Wenn ein Telegramm fehlerfrei empfangen wurde
- Wenn der Antrieb in diesen Telegramm adressiert wurde

Ein Antrieb darf nicht senden, wenn diese Bedingungen nicht erfüllt sind oder der Antrieb im Broadcast angesprochen wurde.

Für den Master besteht die Verbindung zu dem betreffenden Antrieben dann, wenn er nach einer definierten Bearbeitungszeit (Antwortverzugszeit) vom Antrieb ein Antworttelegramm erhält.

Vorgehen zum Einrichten von USS-Kommunikation

1. Konfigurieren Sie in der Gerätesicht des Hardware- und Netzwerkeditors von STEP 7 einen S7-1500-Aufbau mit CPU und CM.
2. Wählen Sie in der Projektnavigation für die CPU den Ordner "Programmbausteine" und öffnen Sie im Ordner den OB 1 durch Doppelklick. Der Programmeditor öffnet sich.
3. Wählen Sie aus der Task Card "Anweisungen", Bereich "Kommunikation", Ordner "Kommunikationsprozessor" die Anweisungen für USS-Kommunikation entsprechend Ihrer Aufgabenstellung und ziehen Sie sie per Drag & Drop in ein Netzwerk des OB 1:
 - Die Anweisung "USS_Port_Scan" ermöglicht die Kommunikation über das USS-Netzwerk.
 - Die Anweisung "USS_Drive_Control" bereitet Sendedaten für den Antrieb vor und wertet die Antwortdaten des Antriebs aus.
 - Die Anweisung "USS_Read_Param" dient dem Auslesen von Parametern aus dem Antrieb.
 - Die Anweisung "USS_Write_Param" dient dem Ändern von Parametern im Antrieb.
4. Parametrieren Sie die Anweisungen entsprechend Ihren Vorgaben.
5. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Eigenschaften Modbus-Protokoll (RTU)

- Die Kommunikation läuft über serielle asynchrone Übertragungen mit einer Übertragungsgeschwindigkeit bis 115,2 kbit/s, halbduplex ab.
- Die Datenübertragung funktioniert nach dem Master-Slave-Prinzip.
- Der Modbus-Master kann Aufträge zum Lesen und Schreiben von Operanden an den Modbus-Slave senden:
 - Lesen von Eingängen, Zeiten, Zählern, Ausgängen, Merkern, Datenbausteinen
 - Schreiben von Ausgängen, Merkern, Datenbausteinen
- Broadcast an alle Slaves ist möglich.

Datenaustausch über Modbus-Kommunikation (RTU)

Das Kommunikationsmodul kann Modbus-Master oder Modbus-Slave sein. Ein Modbus-Master kann mit einem oder mehreren Modbus-Slaves kommunizieren (Anzahl hängt von der Schnittstellen-Physik ab). Nur der vom Modbus-Master explizit angesprochene Modbus-Slave darf Daten an den Modbus-Master zurücksenden. Der Slave erkennt das Ende der Datenübertragung und quittiert diese. Im Fehlerfall stellt er dem Master einen Fehlercode zur Verfügung.

Vorgehen zum Einrichten von Modbus-Kommunikation (RTU)

1. Konfigurieren Sie in der Gerätesicht des Hardware- und Netzwerkeditors von STEP 7 einen S7-1500-Aufbau mit CPU und CM.
2. Wählen Sie in der Projektnavigation für die CPU den Ordner "Programmbausteine" und öffnen Sie im Ordner den OB 1 durch Doppelklick. Der Programmeditor öffnet sich.
3. Wählen Sie aus der Task Card "Anweisungen", Bereich "Kommunikation", Ordner "Kommunikationsprozessor" die Anweisungen für Modbus-Kommunikation entsprechend Ihrer Aufgabenstellung und ziehen Sie sie per Drag & Drop in ein Netzwerk des OB 1:
 - Die Anweisung "Modbus_Comm_Load" konfiguriert den Port des CM für die Modbus-Kommunikation.
 - Die Anweisung "Modbus_Master" wird eingesetzt für die Modbus-Master-Funktionalität.
 - Die Anweisung "Modbus_Slave" wird eingesetzt für die Modbus-Slave-Funktionalität.
4. Parametrieren Sie die Anweisungen entsprechend Ihren Vorgaben.
5. Laden Sie die Hardware-Konfiguration und das Anwenderprogramm in die CPU.

Weitere Informationen

- Weitere Informationen zur Kommunikation über Punkt-zu-Punkt-Kopplung und Grundlagen der seriellen Datenübertragung finden Sie im Funktionshandbuch CM PtP - Konfigurationen für Punkt-zu-Punkt-Kopplungen (<http://support.automation.siemens.com/WW/view/de/59057093>).
- Wie Sie die genannten Anweisungen für die Punkt-zu-Punkt-Kopplung im Anwenderprogramm nutzen, finden Sie in der Online-Hilfe STEP 7 beschrieben.
- Informationen zu den Kommunikationsmodulen mit serieller Schnittstelle finden Sie im jeweiligen Gerätehandbuch des Kommunikationsmoduls.

OPC UA-Kommunikation

9.1 Wissenswertes zu OPC UA

9.1.1 OPC UA und Industrie 4.0

Einheitlicher Standard für den Datenaustausch

Industrie 4.0 steht für die intensive Nutzung, Auswertung und Analyse der zahlreichen Daten aus der Produktion in IT-Systemen der Unternehmensebene. Mit Industrie 4.0 nimmt der Datenaustausch zwischen Produktions- und Unternehmensebene sehr stark zu. Eine Voraussetzung für das Gelingen ist ein einheitlicher Standard zum Datenaustausch.

Der Standard OPC UA (OPC Unified Architecture) ist wegen seiner Unabhängigkeit von bestimmten Betriebssystemen, seinem sicheren Übertragungsverfahren und der semantischen Beschreibung der Daten für den Ebenen übergreifenden Datenaustausch besonders geeignet. OPC UA stellt nicht nur Daten bereit, sondern auch Informationen zu den Daten (Datentypen), und damit wird ein maschineninterpretierbarer Zugriff auf die Daten möglich.

9.1.2 Aufbau der Beschreibung

Bei OPC UA arbeitet ein System als Server und stellt anderen Systemen (Clients) Daten und Methoden zur Verfügung.

OPC UA-Clients greifen lesend und schreibend auf Daten eines OPC UA-Servers zu. OPC UA-Clients rufen Methoden im OPC UA-Server auf.

Ein System kann sowohl Client als auch Server sein.

OPC UA-Server der S7-1500 CPU

Ab Firmware V2.0 ist eine S7-1500 CPU mit einem OPC UA-Server ausgestattet.

Die folgenden Kapitel beschreiben, wie Sie den OPC UA-Server der S7-1500 CPU konfigurieren und damit Daten und Methoden für OPC UA-Clients zur Verfügung stellen, sodass Clients lesend oder schreibend auf PLC-Variablen der CPU zugreifen und Server-Methoden aufrufen können.

Ausserdem zeigen die folgenden Kapitel, wie Sie Companion Spezifikationen in den Adressraum des OPC UA-Servers einbinden.

OPC UA-Clients allgemein

Um den Umgang mit OPC UA-Clients zu verdeutlichen, verwendet die folgende Beschreibung unterschiedliche OPC UA-Clients:

- "UaExpert" von Unified Automation. Ein umfangreicher Client, der kostenlos verwendet werden kann:
Link zum Download von UaExpert (<https://www.unified-automation.com/downloads/opc-ua-clients.html>)
- "UA Sample Client" der OPC Foundation. Dieser Client ist kostenlos verfügbar für Anwender, die bei OPC Foundation registriert sind:
Link zum Download des Beispiel-Clients der OPC Foundation (<https://opcfoundation.org>)

Anwendungsbeispiel im Industry Online Support

Der Siemens Industry Online Support stellt ein kostenloses Anwendungsbeispiel mit einer Client-API zur Verfügung. Mit den Funktionen dieser Schnittstelle können .NET-Entwickler auf den OPC UA-Server einer S7-1500 zugreifen. Die Client-API setzt auf dem .NET OPC UA-Stack der OPC Foundation auf.

Das Anwendungsbeispiel zeigt z. B. den Aufbau von Verbindungen zwischen Server und Client. Das Beispiel zeigt ebenso das Lesen und Schreiben von PLC-Variablen.

Link zum Download OPC UA .NET Client für den SIMATIC S7-1500 OPC UA Server (<http://support.automation.siemens.com/WW/view/de/109737901>)

9.1.3 Allgemeine Eigenschaften von OPC UA

Die wichtigsten Merkmale von OPC UA

- OPC UA ist unabhängig von einer bestimmten Betriebssystemplattform
OPC UA kann zum Beispiel unter Windows, Linux, Mac OS X, einem Echtzeitbetriebssystem oder einem mobilen Betriebssystem (etwa Android) eingesetzt werden.
- OPC UA ist in verschiedenen Programmiersprachen umgesetzt.
Die OPC Foundation hat den Standard OPC UA in mehreren Programmiersprachen implementiert: Stacks für ANSI C, .NET und Java sind verfügbar.
- Die OPC Foundation bietet den Java- und .Net-Stack sowie Beispielpprogramme als Open Source Software an. Siehe Github (<https://github.com/opcfoundation> (<https://github.com/opcfoundation>)).
- Mehrere Unternehmen bieten Software Development Kits (SDK) an, die die Stacks der OPC Foundation und weitere Funktionen enthalten, welche die Entwicklung von Lösungen erleichtern.
Vorteil der Nutzung von SDKs:
 - Support durch den Zulieferer
 - Getestete Software
 - Ausführliche Dokumentation
 - Klare Lizenzbedingungen (wichtig für Weiterverkauf von Lösungen)
- Skalierbarkeit
OPC UA kann in Sensoren genauso verwendet werden wie in eingebetteten Systemen, in Steuerungen, in PC-Systemen und Smartphones wie auch in Servern, auf denen MES- oder ERP-Anwendungen laufen.
Auch kann OPC UA und PROFINET gemeinsam genutzt werden. Beide Protokolle nutzen dieselbe Netzwerk-Infrastruktur.
- Einfaches Client-Server-Prinzip
Ein OPC UA-Server stellt innerhalb eines Netzwerks Informationen bereit und ein OPC UA-Client ruft diese Informationen ab.

- Integrierte Sicherheitsmechanismen

OPC UA verwendet Sicherheitsmechanismen auf unterschiedlichen Ebenen:

- Der Aufbau sicherer Verbindungen zwischen einem OPC UA-Server und einem OPC UA-Client ist nur möglich, wenn sich Client und Server mit Hilfe von X.509-v3 Zertifikaten anmelden können und gegenseitig die Zertifikate anerkennen (Sicherheit auf Anwendungsebene). Verschiedene Security Policies sind möglich, auch eine ungesicherte Verbindung zwischen Server und Client (Security Policy: "Keine Security").
- Ein Server kann für den Zugriff auf Informationen folgendes vom Anwender fordern:
 - ein Zertifikat (nicht in STEP 7 projektierbar)
 - Benutzernamen und Passwort
 - keine Legitimation des Anwenders

Die Sicherheitsmechanismen sind optional und konfigurierbar.

- Unabhängigkeit von einer bestimmten Transportschicht

Die folgenden Transportmechanismen unterstützt OPC UA aktuell:

- Übertragen von Nachrichten als Binärstrom direkt über TCP/IP
- Übertragen von Nachrichten mit XML über TCP/IP und HTTP. Damit ist nur eine langsame Übertragung möglich und wird deshalb wenig verwendet. Es ist nicht in STEP 7 projektierbar und wird von S7-1500 CPUs nicht unterstützt.

Den binären Datenaustausch unterstützt jede OPC UA-Anwendung (durch die OPC UA-Spezifikation vorgeschrieben).

- Abbildung der PLC-Variablen

Die Informationen des OPC UA-Servers (z. B. PLC-Variablen) sind als Knoten (Nodes) modelliert, die miteinander über Referenzen verbunden sind. Dadurch ist es möglich, mit einem OPC UA-Client von Knoten zu Knoten zu navigieren und zu erfahren, welche Inhalte gelesen, beobachtet oder geschrieben werden können.

- Informationen

OPC UA-Server stellen sehr viele Informationen bereit, z. B. zur CPU, zum OPC UA-Server selbst, zu den Daten und zu den Datentypen.

- Instanz-Konzept

OPC UA basiert auf einem Typ-Instanz-Konzept. Sowohl Instanzen als auch deren Typdefinitionen sind zur Laufzeit verfügbar.

- Differenzierung der Funktionalität über Profile: Die S7-1500 unterstützt z. B. das Embedded UA-Server Profil

Siehe auch

Unified Automation (<https://www.unified-automation.com>)

OPC Foundation (<https://opcfoundation.org>)

9.1.4 Von der klassischen OPC-Schnittstelle zu OPC UA

Einheitliche Schnittstelle

Das klassische OPC läuft nur unter Windows-Betriebssystemen.

Um diese Einschränkung zu umgehen, entwickelte die OPC Foundation den Standard OPC UA.

Der Standard ist plattformunabhängig und verwendet ein optimiertes, TCP-basiertes Binärprotokoll für High-Performance Anwendungen.

Damit können unterschiedliche Systeme miteinander Daten austauschen, z. B.:

- Steuerungen mit MES- und ERP-Systemen
- Steuerungen von Siemens mit Steuerungen anderer Hersteller
- Smartphones mit Steuerungen
- Eingebettete Systeme mit Steuerungen
- Intelligente Sensoren mit Steuerungen

9.1.5 Der OPC UA-Server der S7-1500 CPUs

Die S7-1500 CPUs ab Firmware V2.0 sind mit einem OPC UA-Server ausgestattet. Dies betrifft neben den Standard-S7-1500 CPUs auch die Varianten S7-1500F, S7-1500T, S7-1500C, S7-1500pro CPUs, ET 200SP CPUs, SIMATIC S7-1500 SW Controller und PLCSIM Advanced.

Konvention: Mit "S7-1500 CPUs" sind auch die oben genannten CPU-Varianten gemeint.

Grundlagen zum OPC UA-Server der S7-1500 CPU

Der Zugriff auf den OPC UA-Server der CPU ist über alle integrierten PROFINET-Schnittstellen der S7-1500 CPU möglich.

Über CP oder CM ist kein direkter Zugriff über den Rückwandbus des Automatisierungssystems auf den OPC UA-Server der CPU möglich.

Für den Zugriff durch Clients speichert der Server die freigegebenen PLC-Variablen und andere Informationen in Form von Knoten ab (siehe Zugriff auf PLC-Variablen projektieren (Seite 168)). Diese Knoten sind miteinander verbunden und bilden ein Netzwerk. OPC UA definiert Einstiegspunkte in dieses Netzwerk (Well-known Nodes), die das Navigieren zu unterlagerten Knoten ermöglichen.

Mit einem OPC UA-Client können Sie Werte von Variablen des SPS-Programms lesen, beobachten oder schreiben sowie Methoden aufrufen, die der Server zur Verfügung stellt. Ab Firmware Version 2.5 können Sie Methoden implementieren, siehe Server-Methoden (Seite 194).

Knotenklassen

OPC UA-Server stellen Informationen in Form von Knoten (Nodes) zur Verfügung. Ein Knoten kann zum Beispiel ein Objekt, eine Variable oder eine Methode sein.

Das folgende Beispiel zeigt den Adressraum des OPC UA-Servers einer S7-1500 CPU (Ausschnitt aus dem OPC UA-Client "UaExpert" von Unified Automation).

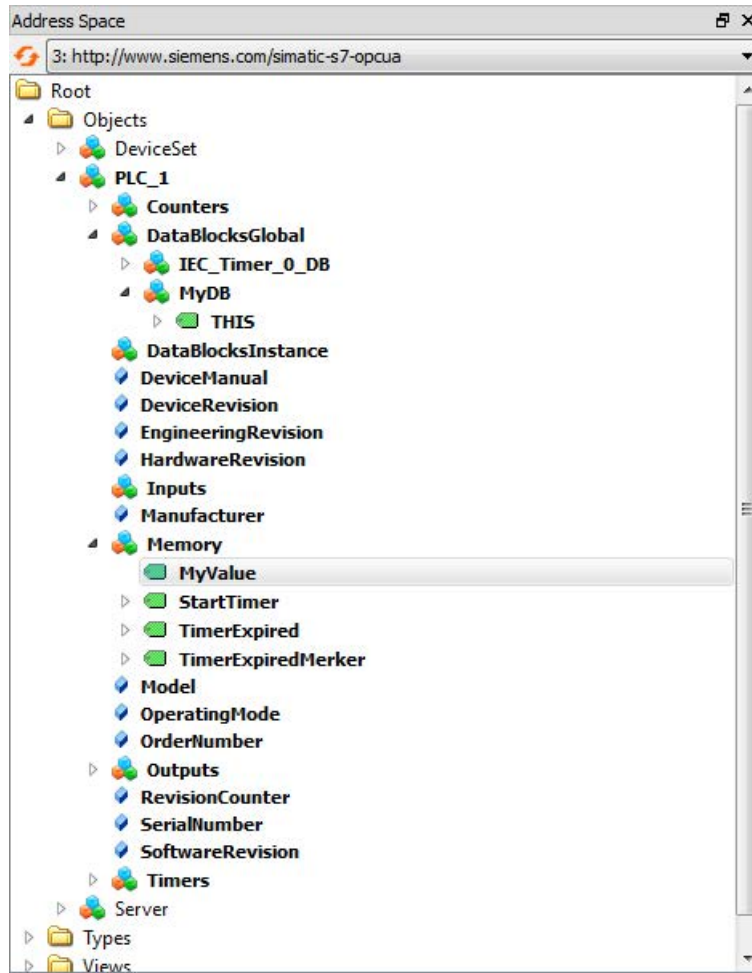


Bild 9-1 Beispiel für den Adressraum des OPC UA-Servers einer S7-1500 CPU

Im Bild oben ist die Variable "MyValue" markiert (grau unterlegt).

Diese Variable befindet sich unterhalb des Knotens "Memory", der die Knotenklasse "Object" besitzt.

"Memory" wiederum befindet sich unterhalb des Knotens "PLC_1" (ebenfalls ein Object).

Adressraum

Die Knoten sind untereinander über Referenzen verbunden, zum Beispiel über die Referenz "HasComponent", die eine hierarchische Beziehung zwischen einem Knoten und seinen unterlagerten Knoten wiedergibt. Über ihre Referenzen bilden die Knoten ein Netzwerk, das zum Beispiel die Form eines Baums besitzen kann.

Ein Netzwerk aus Knoten wird auch als Adressraum bezeichnet. Von der Wurzel ausgehend sind alle Knoten im Adressraum erreichbar.

9.1.6 Adressierung von Knoten

Knoten im OPC UA-Adressraum werden durch eine Nodeld (Node ID oder Node Identifier) eindeutig bestimmt.

Die Nodeld besteht aus dem Identifier, Identifier Type und einem Namensraumindex. Namensräume werden verwendet, um Namenskonflikte zu vermeiden. Die OPC Foundation hat eine Reihe von Knoten definiert, die Auskunft über den jeweiligen OPC UA-Server geben. Diese Knoten sind im Namensraum der OPC Foundation zu finden und besitzen den Index 0.

Weiterhin hat die OPC Foundation Daten- und Variablentypen definiert.

Namespace

Alle Variablen bzw. Methoden einer S7-1500 befinden sich im Namensraum (Namespace) "http://www.siemens.com/simatic-s7-opcua". Standardmäßig besitzt dieser Namensraum den Index 3. Wenn weitere Namensräume in den Server eingefügt oder vorhandene gelöscht werden sollten, kann sich der Index später ändern. Deshalb ist es erforderlich, den aktuellen Index des Namensraums beim Server zu erfragen, bevor Werte gelesen oder geschrieben werden.

Das folgende Bild zeigt das Ergebnis einer solchen Anfrage. Als Beispiel dient das Programm "UaClient" von Siemens, siehe Funktionshandbuch S7-1500 Kommunikation (<https://support.industry.siemens.com/cs/ww/de/view/59192925>).

Find Index of Namespace

http://www.siemens.com/simatic-s7-opcua

Find

NamespaceIndex: 3

Identifier

Der Identifier entspricht dem Namen der PLC-Variablen in Anführungszeichen. Das Anführungszeichen ist das einzige Zeichen, das in STEP 7 nicht als Namensbestandteil erlaubt ist. Durch die Anführungszeichen werden Namenskonflikte vermieden.

Das folgende Beispiel liest den Wert der Variablen "StartTimer":

Index	Boolean Variable		Result
3	"StartTimer"	Read	True

Der Identifier kann aus mehreren Komponenten bestehen. Die einzelnen Komponenten sind dann durch einen Punkt getrennt. Das folgende Beispiel liest den Array-Datenbaustein "MyDB" komplett ein. In diesem Datenbaustein befindet sich ein Array mit zehn Integer-Werten. Alle zehn Werte sollen auf einmal gelesen werden. Deshalb ist bei Array Range "0:9" eingetragen:

Index	Array Datablock of Int16		Results										
3	"MyDB"."THIS"	Read	<table border="1"> <thead> <tr> <th>Index</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>7050</td> </tr> <tr> <td>1</td> <td>7051</td> </tr> <tr> <td>2</td> <td>7052</td> </tr> <tr> <td>3</td> <td>7053</td> </tr> </tbody> </table>	Index	Values	0	7050	1	7051	2	7052	3	7053
Index	Values												
0	7050												
1	7051												
2	7052												
3	7053												
	Array Range (for instance 0:9)												
	0:9												

PLC-Variablen im Adressbereich des OPC UA-Servers

Das folgende Bild zeigt, wo sich die PLC-Variablen des Beispiels im Adressraum des OPC UA-Servers befinden (Ausschnitt aus UA Client):

Der Datenbaustein "MyDB" ist ein globaler Datenbaustein. Deshalb befindet sich der Datenbaustein unterhalb des Knotens "DataBlocksGlobal". "StartTimer" ist eine Merker-Variable und wird deshalb unterhalb des Knotens "Memory" gespeichert.

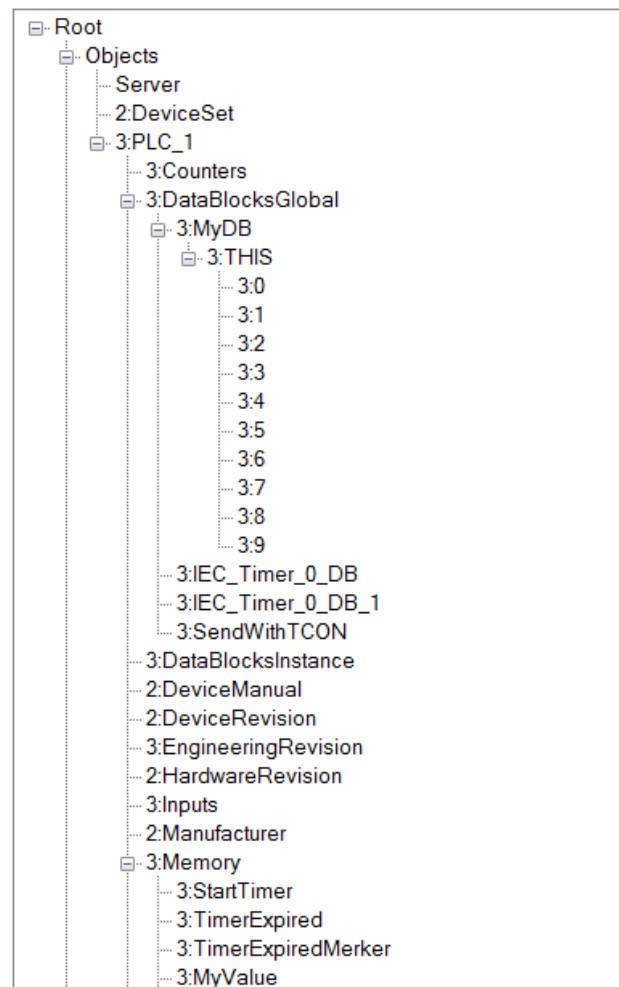


Bild 9-2 PLC-Variablen im Adressbereich des OPC UA-Servers

Methoden im Adressbereiche des OPC UA-Servers

Wenn Sie über Ihr Anwenderprogramm eine Methode implementieren, dann sieht das im Adressraum des OPC UA-Servers folgendermaßen aus (siehe OPC UA-Server-Anweisungen für die Implementierung von Methoden (Seite 199)):

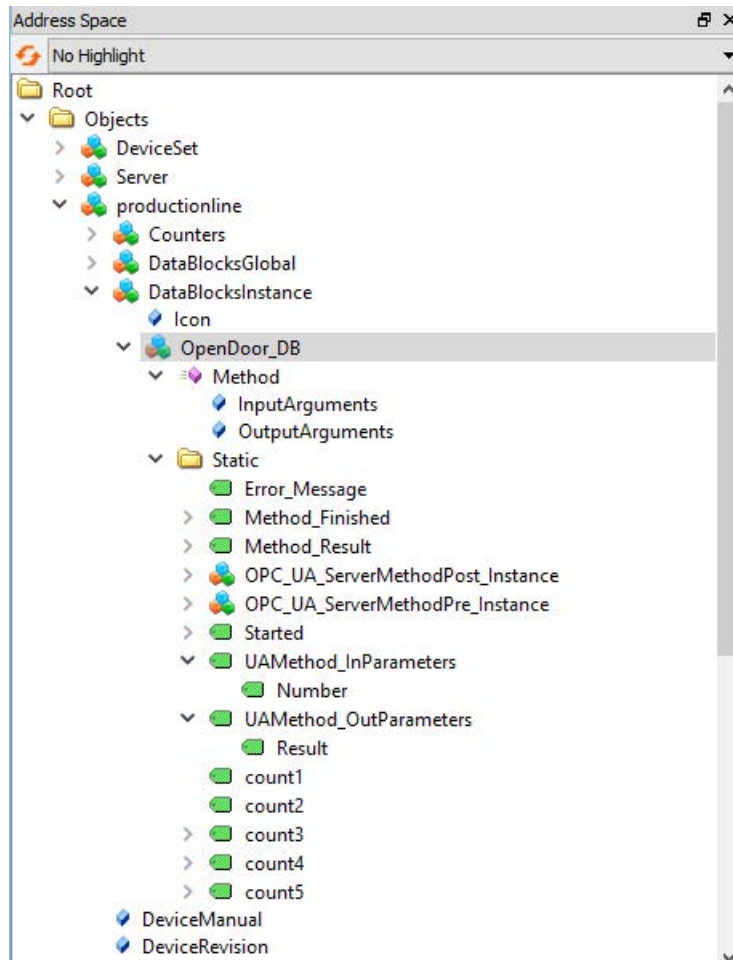


Bild 9-3 Methoden im Adressbereiche des OPC UA-Servers

9.1.7 Mapping von Datentypen

SIMATIC- und OPC UA-Datentypen

SIMATIC-Datentypen stimmen nicht immer mit OPC UA-Datentypen überein.

S7-1500 CPUs stellen SIMATIC-Variablen (mit SIMATIC-Datentypen) dem eigenen OPC UA-Server als OPC UA-Datentypen bereit, sodass OPC UA-Clients über die Server-Schnittstelle auf diese Variablen mit OPC UA-Datentypen zugreifen können.

Ein Client kann von einer solchen Variablen das Attribut "Datatype" lesen und darüber den Original-Datentyp in SIMATIC rekonstruieren.

Beispiel

Eine Variable hat den SIMATIC Datentyp "COUNTER". Sie lesen in der Tabelle COUNTER → UInt16. Sie wissen nun, dass Sie nicht umkodieren müssen, der COUNTER-Wert geht als UInt16 Datentyp über die Leitung.

Der Client kann am Attribut "Datatype" erkennen, dass es eigentlich ein COUNTER-Datentyp ist, und kann ihn mit diesem Wissen rekonstruieren.

Tabelle 9- 1 SIMATIC- und OPC UA-Datentypen

SIMATIC Datentyp	OPC UA-Datentyp
BOOL	Boolean
BYTE	BYTE → Byte
WORD	WORD → UInt16
DWORD	DWORD → UInt32
LWORD	LWORD → UInt64
SINT	SByte
INT	Int16
DINT	Int32
LINT	Int64
USINT	Byte
UINT	UInt16
UDINT	UInt32
ULINT	UInt64
REAL	Float
LREAL	Double
S5TIME	S5TIME → UInt16
TIME	TIME → Int32
LTIME	LTIME → Int64

SIMATIC Datentyp	OPC UA-Datentyp
DATE	DATE → UInt16
TIME_OF_DAY (TOD)	TOD → UInt32
LTIME_OF_DAY (LTOD)	LTOD → UInt64
DATE_AND_TIME (DT)	DT → Byte[8]
LDT	DateTime
DTL	als Struktur gemappt
CHAR	CHAR → Byte
WCHAR	WCHAR → UInt16
STRING (Codepage 1252 bzw. Windows-1252)	STRING → String
WSTRING (UCS-2; Universal Coded Character Set)	String
TIMER	TIMER → UInt16
COUNTER	COUNTER → UInt16

Arrays

Ein Lese- bzw. Schreibauftrag bei OPC UA ist immer ein Array-Zugriff, d. h. grundsätzlich mit Index und Länge versehen, sodass eine Einzelvariable nur ein Sonderfall eines Arrays ist (Index 0 und Länge 1). Auf der Leitung wird der Datentyp einfach mehrfach hintereinander gesendet. Bei der Variablen zeigt das Attribut "Datatype" auf den Basisdatentyp. Aus den Attributen "ValueRank" und "ArrayDimensions" ergibt sich, ob es sich um ein Array handelt und wie groß es ist.

Strukturen

Strukturen werden als ExtensionObject übertragen. Der Server der S7-1500 nutzt die binäre Darstellung für die Übertragung des ExtensionObjects über die Leitung, wobei die einzelnen Strukturelemente direkt hintereinanderliegen. Vorne befindet sich die NodeId des Datentyps, mit deren Hilfe ein Client den Aufbau der Struktur herausfindet.

Bei der OPC UA Specification <= V1.03 muss ein Client dazu das komplette DataTypeDictionary lesen, decodieren und interpretieren (sofern er es nicht bereits vorher Offline durch einen XML-Import gelernt hat).

Weitere Information

Nähere Informationen zur Abbildung der Basisdatentypen, aber auch von Arrays und Strukturen, finden Sie in der OPC UA Spezifikation Part 6, "Mappings" siehe OPC UA BINARY.

9.1.8 Endpunkte der OPC UA-Server

Die Endpunkte der OPC UA-Server definieren die Sicherheitsstufe für eine Verbindung. Je nach Einsatzzweck oder gewünschter Sicherheitsstufe müssen Sie am Endpunkt die entsprechenden Einstellungen für die Verbindung vornehmen.

Verschiedene Security-Einstellungen

Vor dem Aufbau einer gesicherten Verbindung erfragen OPC UA-Clients beim Server, mit welchen Security-Einstellungen Verbindungen möglich sind. Der Server sendet eine Liste zurück mit allen Security-Einstellungen (Endpunkten), die der Server anbietet.

Aufbau von Endpunkten

Endpunkte bestehen aus den folgenden Komponenten:

- Kennung für OPC: "opc.tcp"
- IP-Adresse: 192.168.178.151 (in dem Beispiel)
- Port-Nummer für OPC UA: 4840 (Standard-Port)

Die Port-Nummer ist konfigurierbar, siehe Einstellungen des OPC UA-Servers (Seite 177).

- Security-Einstellung für Nachrichten (Message Security-Modus): None, Sign, SignAndEncrypt.
- Verschlüsselungs- und Hash-Verfahren (Security Policy): None, Basic128Rsa15, Basic256, Basic256Sha256 (in dem Beispiel).

Das folgende Bild zeigt das Programm "UA Sample Client" der OPC Foundation.

Der Client hat eine gesicherte Verbindung zum OPC UA-Server einer S7-1500 CPU aufgebaut, zum Endpunkt "opc.tcp://192.168.178.151:4840 - [SignAndEndCrypt: Basic128Rsa15:Binary]". Die Security-Einstellungen "SignAndEndCrypt:Basic128Rsa15" sind im Endpunkt enthalten.

Hinweis

Endpunkt mit möglichst hoher Security Policy auswählen

Wählen Sie für die Endpunkte eine für die Anwendung angemessene hohe Security Policy aus und deaktivieren Sie am OPC UA-Server die Security Policy mit geringerer Security Policy.

Für die beiden sichersten Endpunkte (Basic256Sha256) des OPC UA-Servers der S7-1500 CPU ist ein Sha256-Zertifikat notwendig.

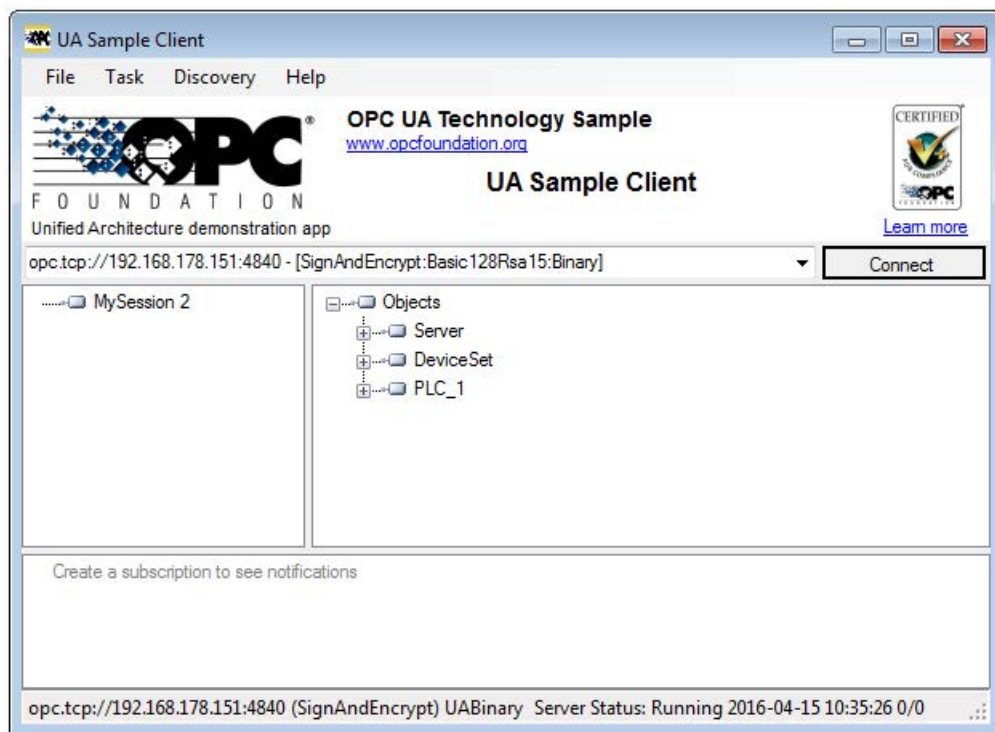


Bild 9-4 Programm "UA Sample Client" der OPC Foundation

Der Aufbau einer Verbindung zu einem Endpunkt des Servers kommt nur zu Stande, wenn der OPC UA-Client die geforderten Sicherheitseinstellungen dieses Endpunkts erfüllt.

Durch den OPC UA-Server bereitgestellte Informationen

OPC UA-Server stellen zahlreiche Daten bereit:

- Die Werte von PLC-Variablen und DB-Komponenten, auf die Clients zugreifen dürfen.
- Die Datentypen dieser PLC-Variablen und DB-Komponenten.
- Angaben zum OPC UA-Server selbst und zur CPU.

Dadurch können sich Clients einen Überblick verschaffen und gezielt Informationen auslesen. Ein vorhergehendes Wissen über das SPS-Programm und den Datenhaushalt der CPU ist nicht erforderlich. Es ist nicht nötig, beim Entwickler des SPS-Programms nachzufragen, wenn PLC-Variablen gelesen werden sollen. Im Server selbst sind alle erforderlichen Angaben gespeichert (zum Beispiel die Datentypen der PLC-Variablen).

Anzeige der Informationen des OPC UA-Servers

Sie haben folgende Möglichkeiten:

- Online: Sie lassen sich zur Laufzeit des OPC UA-Servers alle verfügbaren Informationen anzeigen. Navigieren (Browsen) Sie dazu im Adressraum des Servers.
- Offline: Sie exportieren eine XML-Datei, die auf den XML-Schemata der OPC Foundation basiert.

Selbst erstellte Server-Methoden (FB-Instanz, die von einem OPC UA-Client aufgerufen werden kann) werden nicht mit exportiert (STEP 7 (TIA Portal) V15), siehe Methoden auf dem OPC UA-Server bereitstellen (Seite 194).

- Offline mit dem Openness-API: Sie verwenden in Ihrem Programm das API (Application Programming Interface) des TIA Portals, um die Funktion zum Export aller von OPC UA lesbaren PLC-Variablen aufzurufen. Dafür ist .NET Framework 4.0 erforderlich, siehe TIA Portal Openness, SIMATIC Projekte über Skripte automatisieren (<https://support.industry.siemens.com/cs/ww/de/view/109477163>).
- Wenn Sie bereits die Syntax und das SPS-Programm kennen, können Sie ohne vorhergehende Recherche auf den OPC UA-Server zugreifen.

9.1.9 Verhalten des OPC UA-Servers im Betrieb

Der OPC UA-Server im Betrieb

Der OPC UA-Server der S7-1500 CPU startet, wenn Sie den Server aktivieren und das Projekt in die CPU laden.

Wie Sie den OPC UA-Server aktivieren, ist hier (Seite 173) beschrieben.

Verhalten bei STOP der CPU

Ein aktivierter OPC UA-Server bleibt im Betrieb, auch wenn die CPU in den Betriebszustand "STOP" wechselt. Der OPC UA-Server antwortet dann nach wie vor auf Anfragen von OPC UA-Clients.

Das Verhalten des Servers im Einzelnen:

- Wenn Sie die Werte von PLC-Variablen abfragen, dann erhalten Sie die Werte, die aktuell waren, bevor die CPU in den Betriebszustand "STOP" wechselte oder gesetzt wurde.
- Wenn Sie Werte zum OPC UA-Server schreiben, dann übernimmt der OPC UA-Server diese Werte.

Aber die CPU verarbeitet die Werte nicht, weil das Anwenderprogramm im Betriebszustand "STOP" nicht ausgeführt wird.

Jedoch kann ein OPC UA-Client die neuen Werte aus dem OPC UA-Server der CPU lesen.

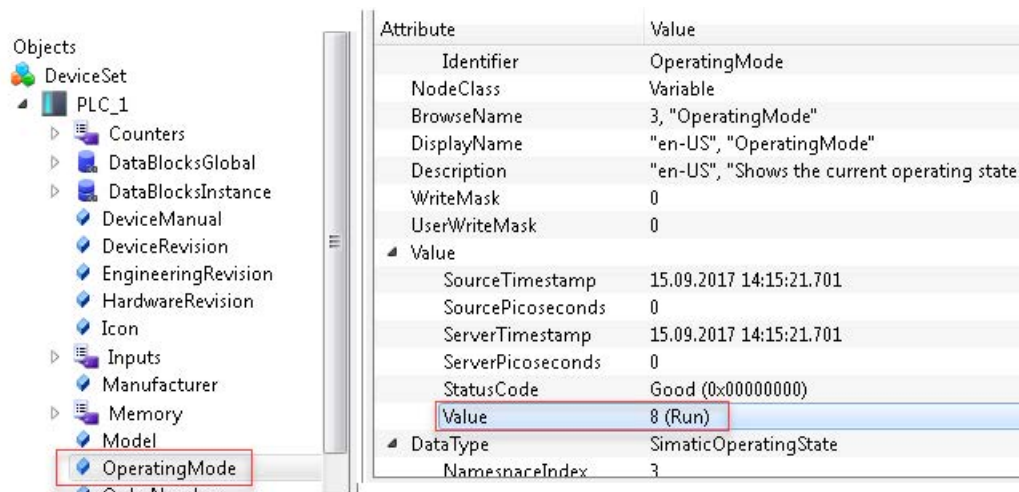
- Wenn Sie eine Server-Methode aufrufen, dann erhalten Sie die Fehlermeldung 16#00AF_0000 (BadInvalidState), da die Server-Methode (Anwenderprogramm) nicht ausgeführt wird.

Neustart des Servers

Der OPC UA-Server wird bei jedem Laden in die CPU gestoppt (z. B. nach dem Laden einer Konfiguration oder eines Bausteins) und startet anschließend neu. Der Neustart des OPC UA-Servers kann abhängig vom Umfang der Datenstruktur etwas dauern.

Betriebszustand der CPU über OPC UA-Server auslesen

Der OPC UA-Server erlaubt Ihnen, den Betriebszustand der CPU auszulesen, siehe folgendes Bild:



Attribute	Value
Identifier	OperatingMode
NodeClass	Variable
BrowseName	3, "OperatingMode"
DisplayName	"en-US", "OperatingMode"
Description	"en-US", "Shows the current operating state"
WriteMask	0
UserWriteMask	0
Value	
SourceTimestamp	15.09.2017 14:15:21.701
SourcePicoSeconds	0
ServerTimestamp	15.09.2017 14:15:21.701
ServerPicoSeconds	0
StatusCode	Good (0x00000000)
Value	8 (Run)
DataType	SimaticOperatingState
NamespaceIndex	3

Bild 9-5 Betriebszustand der CPU über OPC UA-Server auslesen

Neben dem Betriebszustand der CPU können Sie auch den Status (State) des Servers auslesen.

9.1.10 Wissenswertes zu OPC UA-Clients

Grundlagen zu OPC UA-Clients

OPC UA-Clients sind Programme, die Folgendes leisten:

- Informationen von einem OPC UA-Server, z. B. einer S7-1500 CPU, lesen und schreiben
- Methoden durch den OPC UA-Server ausführen lassen

OPC UA-Clients können jedoch nur auf Daten zugreifen, die dafür freigegeben sind (siehe "Schreib- und Leserechte verwalten (Seite 168)").

Um eine Verbindung zu einem OPC UA-Server aufzubauen, benötigen Sie den Endpunkt des Servers (siehe "Endpunkte der OPC UA-Server (Seite 147)").

Informationen aus dem OPC UA-Server auslesen

Wenn eine Verbindung zu einem Endpunkt des Servers besteht, dann können Sie die Navigationsfunktion des Clients nutzen: Sie navigieren von einem definierten Ausgangspunkt ausgehend (vom Wurzelknoten "Root" aus) durch den Adressraum des Servers.

Dadurch erhalten Sie unter anderem die folgenden Informationen:

- Freigegebene PLC-Variablen, Datenbausteine und Datenbaustein-Komponenten
- Namensraumindex und Identifier dieser PLC-Variablen, Datenbausteine und DB-Komponenten
- Datentypen der PLC-Variablen und DB-Komponenten
- Anzahl von Komponenten in Arrays (fürs Lesen und Schreiben von Arrays erforderlich)

Darüber hinaus erhalten Sie Information über den OPC UA-Server selbst, sowie Informationen über die S7-1500 basierend auf dem Standard "OPC UA for Devices" der OPC Foundation, zum Beispiel Seriennummer, Firmware-Version.

Daten vom Server lesen und zum Server schreiben

Sie kennen nun den Namensraumindex, Identifier und Datentyp von PLC-Variablen. Damit können Sie gezielt einzelne PLC-Variablen und DB-Komponenten wie auch ganze Arrays und Strukturen lesen. Beispiele für das Lesen von Bool'schen Variablen und Array-Datenbausteinen finden Sie unter Knoten adressieren (Seite 141).

Mit den Informationen, die Sie beim Navigieren durch den Adressraum des Servers erhalten (Index, Identifier und Datentyp), können Sie mit dem OPC UA-Client auch Werte in die S7-1500 übertragen. Das folgende Beispiel überschreibt im Array-Datenbaustein "MyDB" die ersten drei Werte.

Index	Array Datablock of Int16	Values	Status Code
3	"MyDB"."THIS"	1 2 3	Write Good
	Array Range (for instance 0:9)		
	0:2		

Bei "Array Range" geben Sie an, welche Komponenten des Arrays Sie überschreiben wollen. Am Status Code "Good" ist zu sehen, dass die Werte erfolgreich übertragen werden konnten. Sie können jedoch nur die Werte zur S7-1500 schreiben, nicht die Zeitstempel dieser Werte. Die Zeitstempel können nur gelesen werden.

Schnellerer Zugriff durch Registrierung

Die bisherigen Beispiele verwenden als Identifier-Zeichenketten, zum Beispiel "MyBD2"."THIS". Wenn aber als Identifier eine numerische NodeID anstelle einer String NodeID dient, dann sind Zugriffe wesentlich schneller. Deshalb sollten Sie bei regelmäßigen Zugriffen auf bestimmte Variablen die Funktionen "RegisteredRead" und "RegisteredWrite" nutzen.

Hierbei meldet Ihr Client zunächst die PLC-Variable beim Server an. Und der Server sendet einen Identifier zurück, den der Client für die eigentlichen Zugriffe nutzt. Dieser Identifier gilt ausschließlich für die aktuelle Session und muss im Falle des Abbaus/Verlustes der Session neu erfragt werden.

Beim folgenden Beispiel wurde zunächst die Variable "StartTimer" beim Server registriert. Anschließend wird für das Setzen des Wertes die schnelle Funktion "RegisteredWrite" genutzt.

Index	Boolean Variable		Value		Status Code
3	"StartTimer"	<input type="button" value="Register"/>	<input checked="" type="checkbox"/>	<input type="text" value="True"/>	<input type="button" value="Write"/>
		<input type="button" value="Unregister"/>			
<input type="text" value="Good"/>					

Nach dem gleichen Schema lässt sich auch die Funktion "RegisteredRead" nutzen, was insbesondere beim wiederkehrendem Auslesen von Daten sinnvoll ist. Beachten Sie jedoch, dass es je nach Anwendung sinnvoller sein kann, eine Subscription zu verwenden.

Empfehlung: Registrierungen platzieren Sie am besten im Hochlauf-Programm des OPC UA-Clients, da die Registrierung Zeit in Anspruch nimmt.

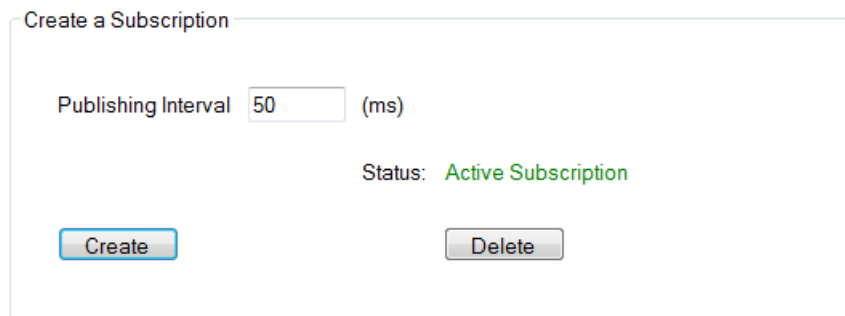
Beachten Sie, dass Sie in den Eigenschaften der S7-1500 CPU die maximale Anzahl registrierter Knoten einstellen können und die Clients diese Anzahl berücksichtigen müssen, siehe Allgemeine Einstellungen des OPC UA-Servers (Seite 177).

Subscription

Mit "Subscription" wird eine Funktion bezeichnet, bei der nur Variablen übertragen werden, für die sich ein OPC UA-Client beim OPC UA-Server angemeldet hat. Der OPC UA-Server sendet für diese angemeldeten Variablen (Subscriptions) nur dann eine Nachricht an den OPC UA-Client, wenn sich ein Wert geändert hat. Durch die Überwachung dieser Variablen entfällt ein ständiges Abfragen durch den OPC UA-Client; die Netzlast wird reduziert.

Um diese Funktion zu nutzen, müssen Sie eine Subscription anlegen. Dazu geben Sie beim UaClient das Sendeintervall ("Publishing Interval") vor und klicken auf die Schaltfläche "Create". Das Sendeintervall ist das Zeitintervall, in dem neue Werte an den Client gesendet werden.

Beim folgenden Beispiel wurde eine Subscription angelegt: Alle 50 Millisekunden erhält hier der Client eine Nachricht mit den neuen Werten (Sendeintervall 50 ms).



The screenshot shows a window titled "Create a Subscription". Inside, there is a label "Publishing Interval" followed by a text input field containing the number "50" and the unit "(ms)". Below this, the status is displayed as "Status: Active Subscription" in green text. At the bottom, there are two buttons: "Create" on the left and "Delete" on the right.

Server vor Überlast schützen

Den OPC UA-Server der S7-1500 CPU können Sie mithilfe des Parameters "Kleinstes Sendeintervall" so einstellen, dass er nicht extrem kurze, vom Client gewünschte Sendeintervalle bedient, siehe Einstellungen des OPC UA-Servers (Seite 177).

Beispiel: Ein Client möchte wie oben beschrieben in einem Sendeintervall von 50 ms bedient werden. Ein so kurzes Sendeintervall würde aber eine hohe Netzlast und eine hohe Belastung des Servers hervorrufen. Daher stellen Sie als "Kleinstes Sendeintervall" beim Server 1000 ms ein. Clients, die kürzere Sendeintervall in ihrer Subscription fordern, werden so auf 1000 ms "heruntergebremst", der Server vor Überlastung geschützt.

Überwachung von PLC-Variablen

Wenn die Subscription angelegt ist, teilen Sie dem Server mit, welche Variablen er damit überwachen soll. Im folgenden Beispiel wurde die Variable "Voltage" der Subscription hinzugefügt.

Index	LREAL Variable	Sampling Interval		Value
<input type="text" value="3"/>	<input type="text" value="Voltage"/>	<input type="text" value="-1"/>	<input type="button" value="Add and Monitor"/>	<input type="text" value="2,214265046"/>
Queue Size	Deadband			
<input type="text" value="1"/>	<input type="text" value="0,1"/>			

Die Variable "Voltage" enthält den Wert einer Spannungsgröße, die von einer S7-1500 CPU erfasst wird (über ein Analogeingabemodul AI 4xU/I 2-wire).

Das Abtastintervall (Sampling Intervall) enthält einen negativen Wert (-1). Dadurch wird festgelegt, dass der OPC UA-Server die Werte mit dem ihm kleinstmöglichen Abstand auf Änderungen überprüft.

Die Länge der Warteschlange ist hier im Beispiel auf "1" festgelegt: Es wird immer nur ein Wert im Intervall von 50 Millisekunden aus der CPU gelesen und anschließend an den OPC UA-Client gesendet, wenn sich der Wert verändert hat.

Der Parameter "Deadband" ist im Beispiel "0,1": Wertänderungen müssen mindestens 0,1 Volt betragen, nur dann sendet der Server den neuen Wert an den Client. Kleinere Wertänderungen sendet der Server nicht. Mit diesem Parameter können Sie zum Beispiel Rauschen ausblenden: geringfügige Änderungen einer Prozessgröße, denen keine reale Bedeutung zukommt.

9.2 Security bei OPC UA

9.2.1 Security-Einstellungen

Gefahren begegnen

OPC UA erlaubt den Datenaustausch zwischen unterschiedlichen Systemen, sowohl innerhalb der Prozess- und Produktionsebene, als auch zu Systemen der Leit- und Unternehmensebene.

Diese Möglichkeit birgt auch Security-Risiken. Deshalb verwendet OPC UA eine Reihe von Sicherheitsmechanismen:

- Prüfung der Identität von OPC UA-Server und -Clients.
- Prüfung der Identität der Anwender.
- Signierter/verschlüsselter Datenaustausch zwischen OPC UA-Server und -Clients.

Die Sicherheitseinstellungen sollten nur in begründeten Fällen umgangen werden:

- Während der Inbetriebnahme
- Bei Inselprojekten ohne Ethernet-Verbindung nach außen

Wenn Sie z. B. beim "UA Sample Client" der OPC Foundation den Endpunkt "None" auswählen, dann gibt das Programm eine deutliche Warnung aus:

Warning: Selected Endpoint has no security.

Hinweis

Nicht erwünschte Security Policys deaktivieren

Wenn Sie bei den Secure-Channel-Einstellungen des S7-1500 OPC UA-Servers alle Security Policys aktiviert haben (Voreinstellung) - also auch den Endpunkt "None" (Keine Security) - dann ist der Datenverkehr zwischen Server und Client auch ungesichert möglich (weder signiert noch verschlüsselt). Der OPC UA-Server der S7-1500 CPU sendet auch bei "None" (Keine Security) sein öffentliches Zertifikat an den Client. Und manche Clients prüfen dieses Zertifikat. Doch der Client sendet kein Zertifikat an den Server. Die Identität des Clients bleibt bei "None" unbekannt. Jeder OPC UA-Client kann sich dann mit dem Server verbinden, unabhängig von sämtlichen noch folgenden Security-Einstellungen.

Achten Sie bei der Projektierung des OPC UA-Servers darauf, dass nur Security Policys aktiviert sind, die mit dem Schutzkonzept für Ihre Maschine oder Anlage vereinbar sind. Alle anderen Security Policys sind zu deaktivieren.

Empfehlung: Verwenden Sie die Einstellung "Basic256Sha256" bei der der Server nur Sha256-Zertifikate akzeptiert.

Weitere Security-Regeln

- Nutzen Sie nur im Ausnahmefall den Endpunkt "None".
- Verwenden Sie nur im Ausnahmefall die "Gast-Authentifizierung" des Benutzers.
- Erlauben Sie nur dann den Zugriff auf PLC-Variablen und DB-Komponenten über OPC UA, wenn es tatsächlich erforderlich ist.
- Nutzen Sie die Listen vertrauenswürdiger Clients in den Einstellungen des S7-1500 OPC UA-Clients, um nur bestimmten Clients Zugriff zu erlauben.

9.2.2 Zertifikate gemäß X.509 der ITU

Bei OPC UA sind Sicherheitsmechanismen in mehreren Schichten integriert. Dabei spielen digitale Zertifikate eine wichtige Rolle. Ein OPC UA-Client kann nur eine gesicherte Verbindung zu einem OPC UA-Server aufbauen, wenn der Server das digitale Zertifikat des Clients akzeptiert und als vertrauenswürdig einstuft.

Siehe Kapitel "OPC UA-Server der S7-1500 konfigurieren (Seite 173)".

Außerdem muss auch der Client das Zertifikat des Servers prüfen und ihm vertrauen. Server und Client müssen sich ausweisen und beweisen, dass sie tatsächlich der sind, der sie zu sein behaupten: Sie müssen ihre Identität nachweisen. Die gegenseitige Authentifizierung von Client und Server verhindert zum Beispiel Angriffe durch einen "Man in the Middle".

Angriffe durch "Man in the Middle"

Zwischen Server und Client könnte sich ein "Man in the Middle" befinden, ein Programm, das die Kommunikation zwischen Server und Client abfängt und behauptet, selbst Client oder Server zu sein, und so wichtige Informationen über das S7-Programm erhält oder Werte in der CPU setzt und damit eine Maschine oder Anlage angreifen kann.

Bei OPC UA werden digitale Zertifikate verwendet, die dem Standard X.509 der International Telecommunication Union (ITU) entsprechen.

Damit lässt sich die Identität eines Programms, eines Rechners oder einer Organisation nachweisen (authentifizieren).

X.509-Zertifikate

Ein X.509-Zertifikat enthält unter anderem die folgenden Informationen:

- Versionsnummer des Zertifikats
- Seriennummer des Zertifikats
- Informationen über den Algorithmus, den die Zertifizierungsstelle zum Signieren des Zertifikats verwendete.
- Name der Zertifizierungsstelle
- Beginn und Ende der Gültigkeit des Zertifikats
- Name des Programms, der Person oder Organisation, für die das Zertifikat von der Zertifizierungsstelle signiert wurde.
- Der öffentliche Schlüssel des Programms, der Person oder Organisation.

Somit verknüpft ein X509-Zertifikat eine Identität (Name eines Programms, einer Person oder einer Organisation) mit dem öffentlichen Schlüssel des Programms, der Person oder Organisation.

Prüfung beim Verbindungsaufbau

Beim Verbindungsaufbau zwischen Client und Server prüfen die Teilnehmer alle Informationen aus dem Zertifikat, die zur Feststellung der Integrität notwendig sind, z. B. Signatur, Gültigkeitsdauer, Applikationsname (URN).

Signieren und Verschlüsseln

Damit überprüft werden kann, ob ein Zertifikat manipuliert wurde, werden Zertifikate signiert.

Hier gibt es verschiedene Vorgehensweisen:

- Sie wenden sich an eine Zertifizierungsstelle (CA) und lassen Ihr Zertifikat signieren.
In diesem Fall überprüft die Zertifizierungsstelle Ihre Identität und signiert Ihr Zertifikat mit dem privaten Schlüssel der Zertifizierungsstelle. Senden Sie dazu an die Zertifizierungsstelle einen CSR (Certificate Signing Request). Wie Sie einen CSR mit dem Tool OpenSSL selbst erzeugen, ist hier beschrieben. (Seite 160)
- Sie erstellen selbst ein Zertifikat und signieren es.
Dazu nutzen Sie zum Beispiel das Programm "Opc.Ua.CertificateGenerator" der OPC Foundation. Wie Sie vorgehen, ist hier (Seite 38) beschrieben. Oder Sie verwenden OpenSSL: Eine Anleitung finden Sie unter PKI-Schlüsselpaare und Zertifikate selbst erzeugen (Seite 162).
- Die einfachste Möglichkeit: Innerhalb des TIA Portals haben Sie beide Möglichkeiten. Das TIA Portal kann Zertifikate erzeugen und signieren. Wenn Sie Ihr Projekt geschützt haben und als Benutzer angemeldet sind mit dem Funktionsrecht, Security-Einstellung vornehmen zu dürfen, dann sind auch die globalen Security-Einstellungen nutzbar. Die globalen Security-Einstellungen erlauben Zugriff auf den Zertifikatsmanager und damit auch auf die Zertifizierungsstelle (CA) des TIA Portals.

Exkurs: Zertifikatstypen

- Selbst signiertes Zertifikat:

Jeder Teilnehmer erzeugt sein eigenes Zertifikat und signiert es. Beispielanwendungen: Statische Konfiguration mit begrenzter Anzahl von Kommunikationsteilnehmern.

Aus einem selbst signierten Zertifikat können keine neuen Zertifikate abgeleitet werden. Allerdings müssen Sie alle selbst signierten Zertifikate der Partnergeräte in die CPU laden (STOP erforderlich).

- CA-Zertifikat:

Alle Zertifikate werden von einer Zertifizierungsstelle erstellt und signiert.

Beispielanwendungen: Dynamisch wachsende Anlagen.

Sie müssen nur das Zertifikat der Zertifizierungsstelle in die CPU laden. Die Zertifizierungsstelle kann neue Zertifikate erzeugen (Hinzufügen von Partnergeräten ohne STOP der CPU möglich).

Signieren

Durch die Signatur lässt sich die Integrität und Herkunft einer Nachricht nachweisen, wie im Folgenden beschrieben ist.

Beim Signieren bildet der Sender zunächst aus dem Klartext (Klarnachricht) einen Hashwert. Dann verschlüsselt der Sender den Hashwert mit seinem privaten Schlüssel und überträgt schließlich den Klartext zusammen mit dem verschlüsselten Hashwert zum Empfänger. Der Empfänger benötigt zur Überprüfung der Signatur den öffentlichen Schlüssel des Senders (ist im X509-Zertifikat des Senders enthalten). Mit dem öffentlichen Schlüssel des Senders entschlüsselt der Empfänger den erhaltenen Hashwert. Dann bildet der Empfänger selbst aus dem empfangenen Klartext den Hashwert (das Hashverfahren ist im Zertifikat des Senders enthalten). Anschließend vergleicht der Empfänger die beiden Hashwerte:

- Wenn die beiden Hashwerte gleich sind, dann ist die Klarnachricht unverändert beim Empfänger angekommen und wurde nicht manipuliert.
- Wenn die beiden Hashwerte nicht gleich sind, dann ist die Klarnachricht nicht identisch beim Empfänger angekommen. Die Klarnachricht wurde manipuliert oder bei der Übertragung verfälscht.

Verschlüsseln

Durch Verschlüsseln von Daten verhindern Sie, dass Unbefugte Kenntnis vom Inhalt erhalten. X509-Zertifikate werden nicht verschlüsselt; sie sind öffentlich und jedermann kann sie einsehen.

Beim Verschlüsseln verschlüsselt der Sender die Klarnachricht mit dem öffentlichen Schlüssel des Empfängers. Dazu benötigt der Sender das X509-Zertifikat des Empfängers, weil darin der öffentliche Schlüssel des Empfängers enthalten ist. Der Empfänger entschlüsselt die Nachricht mit seinem privaten Schlüssel. Nur der Empfänger kann die Nachricht entschlüsseln. Er allein besitzt den privaten Schlüssel. Deshalb darf der private Schlüssel nie weitergegeben werden.

Secure Channel

OPC UA verwendet die privaten und öffentlichen Schlüssel von Client und Server beim Aufbau einer gesicherten Verbindung, des Secure Channels. Wenn die gesicherte Verbindung aufgebaut ist, dann erzeugen Client und Server einen internen, nur ihnen bekannten Schlüssel, den sie zum Signieren und Verschlüsseln von Nachrichten verwenden. Dieses symmetrische Verfahren (ein gemeinsamer Schlüssel) ist sehr viel schneller als unsymmetrische Verfahren (private und öffentliche Schlüssel).

9.2.3 Zertifikate bei OPC UA

Verwendung von X.509-Zertifikaten bei OPC UA

OPC UA verwendet beim Aufbau einer Verbindung von Client zu Server drei Arten von X.509-Zertifikaten:

- OPC UA Applikations-Zertifikate

Solche X.509-Zertifikate identifizieren die Software-Instanz, die jeweilige Installation einer Client- oder Server-Software. Beim Attribut "Organisation Name" tragen Sie den Namen des Unternehmens ein, das die Software einsetzt.

Hinweis

Der OPC UA-Server der S7-1500 verwendet Applikations-Zertifikate auch bei der Security-Einstellung "None" (Keine Security). Dadurch wird die Kompatibilität zu OPC UA V1.1 und früher gewahrt.

- OPC UA Software-Zertifikate

Dieses X509-Zertifikat identifiziert eine konkrete Version der Client- oder Server-Software. Solche Zertifikate enthalten Attribute, die beschreiben, welche Tests diese Version der Software bei der Zertifizierung durch die OPC Foundation (bzw. anerkannte Testlabors) bestanden hat. Beim Attribut "Organisation Name" tragen Sie den Namen des Unternehmens ein, das die Software entwickelt hat oder vertreibt.

Hinweis

Bei STEP 7 V15 werden keine Software-Zertifikate unterstützt.

- OPC UA Anwender-Zertifikate

Dieses X509-Zertifikat identifiziert den konkreten Anwender, der zum Beispiel vom OPC UA-Server einer S7-1500 CPU Prozessdaten abruft. Dieses Zertifikat ist nicht erforderlich, wenn der Anwender seine Berechtigung mit seinem Passwort nachweisen kann oder ein anonymer Zugang konfiguriert ist.

Hinweis

Bei STEP 7 V15 werden keine Anwender-Zertifikate unterstützt.

Diese Zertifikate sind End-Entity-Zertifikate: Sie identifizieren zum Beispiel eine Person, eine Organisation, ein Unternehmen, eine Instanz (Installation) einer Software.

9.2.4 Selbst-signierte Zertifikate erzeugen

Das folgende Kapitel ist nur dann relevant, wenn Sie einen OPC UA-Client verwenden, der kein Client-Zertifikat erzeugt.

Sie können selbst-signierte Zertifikate mit STEP 7 erstellen.

Dazu gehen Sie folgendermaßen vor:

1. In den Eigenschaften der CPU, unter "Schutz & Security > Zertifikatsmanager > Gerätezertifikate" doppelklicken Sie auf "<Neu hinzufügen>"
2. Klicken Sie auf "Hinzufügen".
3. Im Dialog "Neues Zertifikat erzeugen" wählen Sie bei "Verwendungszweck" die Option "OPC UA-Client".
4. Klicken Sie auf "OK".

Das folgende Kapitel beschreibt, wie Sie mit anderen Tools als STEP 7 selbst-signierte Zertifikate erzeugen.

Im Feld "Alternativer Antragstellername" (Subject Alternative Name) trägt STEP 7 automatisch die URI für das erstellte Zertifikat ein. Bei der programmtechnischen Zertifikatserstellung über das .Net-Stack der OPC Foundation heißt das Feld z. B. "ApplicationUri", bei anderen Tools zur Zertifikatserstellung kann es anders heißen.

Zertifikats-Generator der OPC Foundation verwenden

Ein selbst-signiertes Client-Zertifikat können Sie zum Beispiel mit dem OPC.UA.CertificateGenerator erzeugen.

Dazu gehen Sie folgendermaßen vor:

1. Laden Sie das Tool von der Webseite der OPC Foundation. Sie finden das Programm auf der Webseite der OPC Foundation (<https://opcfoundation.org/developer-tools/developer-kits-unified-architecture>) zum Beispiel unter "Ressources > Samples / Code > Unified Architecture" in "Sample Applications".
2. Installieren Sie die Beispielapplikationen der OPC Foundation auf Ihrem PC.
3. Rufen Sie mit dem Windows-Explorer das Installationsverzeichnis auf: Sie erreichen es unter "C:\Program Files (x86)\OPC Foundation\UA 1.02\Sample Applications".
4. Halten Sie die Umschalttaste gedrückt und klicken Sie mit der rechten Maustaste in das Verzeichnis, sodass das Kontextmenu eingeblendet wird.
5. Wählen Sie "Eingabeaufforderung hier öffnen".
6. Tragen Sie in die Eingabeaufforderung nach dem Prompt-Zeichen den folgenden Befehl ein: "Opc.Ua.CertificateGenerator -cmd issue -sp . -an MyClient"

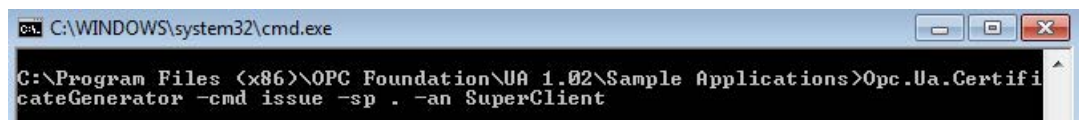
7. Klicken Sie die Eingabetaste.

8. Das Programm erzeugt für "MyClient.":

- Im Unterordner "certs" das Zertifikat "MyClient" mit dem öffentlichen Schlüssel des Clients
- Im Unterordner "private" den privaten Schlüssel des Clients.

"MyClient" ist nur ein Beispiel. Angenommen, Ihr OPC UA-Client heißt "SuperClient", dann geben Sie die folgende Zeile in die Eingabeaufforderung ein:

"Opc.Ua.CertificateGenerator -cmd issue -sp . -an SuperClient". Das folgende Bild zeigt die Eingabe in der Kommandozeile:

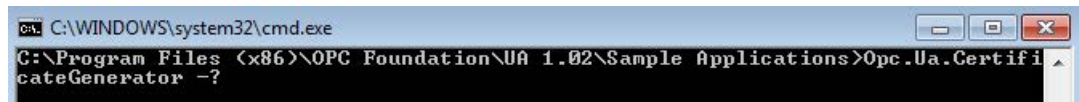


```
C:\WINDOWS\system32\cmd.exe
C:\Program Files (x86)\OPC Foundation\UA 1.02\Sample Applications>Opc.Ua.CertificateGenerator -cmd issue -sp . -an SuperClient
```

Hilfe bei anderen Generatorversionen

Der Beschreibung liegt der "Opc.Ua.CertificateGenerator" der OPC Foundation vom 25. Juni 2015 zu Grunde. Bei anderen Versionen des Generators kann eine andere Eingabe erforderlich sein. Um Informationen über die erforderliche Eingabe zu erhalten, gehen Sie folgendermaßen vor:

1. Geben Sie den folgenden Befehl in die Eingabeaufforderung ein:
"Opc.Ua.CertificateGenerator -?"



```
C:\WINDOWS\system32\cmd.exe
C:\Program Files (x86)\OPC Foundation\UA 1.02\Sample Applications>Opc.Ua.CertificateGenerator -?
```

Die Hilfe wird angezeigt.

2. Verwenden Sie die Optionen, die bei "Create a self-signed Application Certificate" eingetragen sind.

9.2.5 PKI-Schlüsselpaare und Zertifikate selbst erzeugen

Dieses Kapitel ist für Sie nur dann relevant, wenn Sie einen OPC UA-Client verwenden wollen, der nicht selbst ein PKI-Schlüsselpaar und ein Client-Zertifikat erzeugen kann. Sie erzeugen in diesem Fall mit OpenSSL einen privaten und öffentlichen Schlüssel, generieren ein X509-Zertifikat und signieren das Zertifikat selbst.

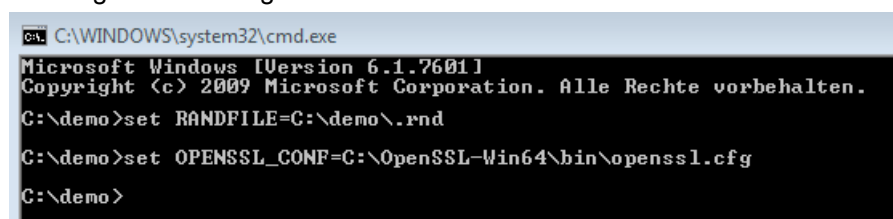
OpenSSL verwenden

OpenSSL ist ein Tool zum Erzeugen von Zertifikaten. Sie können auch andere Tools verwenden, z. B. XCA, eine Schlüsselverwaltungssoftware mit grafischer Oberfläche für eine bessere Übersicht von ausgestellten Zertifikaten.

Um mit OpenSSL unter Windows zu arbeiten, gehen Sie folgendermaßen vor:

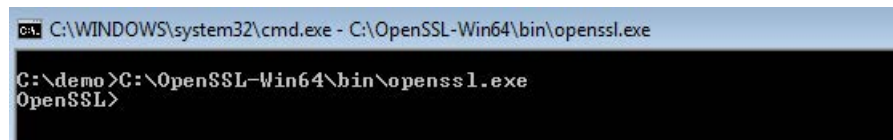
1. Installieren Sie OpenSSL unter Windows. Wenn Sie eine 64-Bit-Version des Betriebssystems verwenden, dann installieren Sie OpenSSL zum Beispiel im Verzeichnis "C:\OpenSSL-Win64". Sie erhalten OpenSSL-Win64 als Download bei verschiedenen Anbietern für Open Source Software.
2. Legen Sie ein Verzeichnis an, zum Beispiel "C:\demo".
3. Öffnen Sie die Kommandozeile (cmd.exe). Dazu klicken Sie auf "Start" und tragen im Suchfeld "cmd" ein. Klicken Sie in der Ergebnisliste mit der rechten Maustaste auf "cmd.exe" und führen Sie das Programm als Administrator aus. Windows öffnet die Kommandozeile (DOS-Prompt).
4. Wechseln Sie zum Verzeichnis "C:\demo". Dazu geben Sie den folgenden Befehl ein: "cd C:\demo".
5. Setzen Sie die folgenden Umgebungsvariablen:
 - set RANDFILE=c:\demo\.rnd
 - set OPENSSL_CONF=C:\OpenSSL-Win64\bin\openssl.cfg

Das folgende Bild zeigt die Kommandozeile mit den Befehlen:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.
C:\demo>set RANDFILE=C:\demo\.rnd
C:\demo>set OPENSSL_CONF=C:\OpenSSL-Win64\bin\openssl.cfg
C:\demo>
```

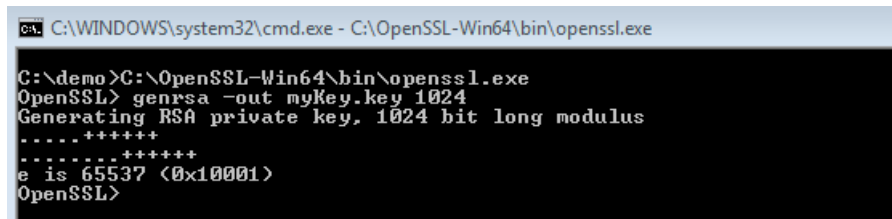
6. Starten Sie nun OpenSSL. Wenn OpenSSL im Verzeichnis C:\OpenSSL-Win64 installiert wurde, dann geben Sie ein: C:\OpenSSL-Win64\bin\openssl.exe. Das folgende Bild zeigt die Kommandozeile mit dem Befehl:



```
C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
C:\demo>C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL>
```

7. Generieren Sie einen privaten Schlüssel. Speichern Sie den Schlüssel in die Datei "myKey.key". Der Schlüssel ist in diesem Beispiel 1024 Bit lang; für eine erhöhte Sicherheit von RSA verwenden Sie in der Praxis 2048 Bit! Geben Sie den folgenden Befehl ein: "genrsa -out myKey.key 2048" ("genrsa -out myKey.key 1024" im Beispiel).

Das folgende Bild zeigt die Kommandozeile mit dem Befehl und der Ausgabe von OpenSSL:



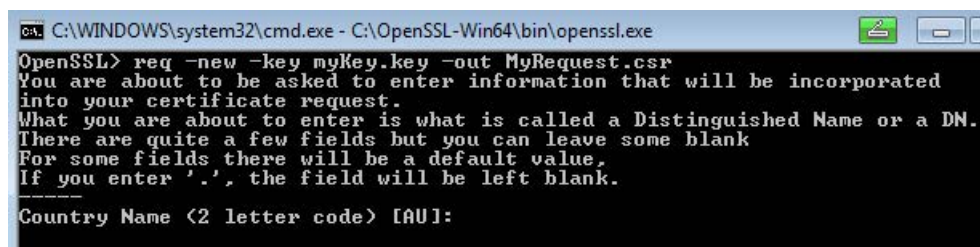
```
C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
C:\demo>C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL> genrsa -out myKey.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
OpenSSL>
```

8. Erzeugen Sie einen CSR (Certificate Signing Request), eine Aufforderung, ein Zertifikat zu signieren. Dazu geben Sie den folgenden Befehl ein: "req -new -key myKey.key -out myRequest.csr". Während der Ausführung dieses Befehls fragt OpenSSL Sie nach Angaben zu Ihrem Zertifikat:

- Country Name: z. B. "DE", für Deutschland, "FR" für Frankreich
- State or Province Name: z. B. "Bayern".
- Location Name: z. B. "Augsburg".
- Organisation Name: Tragen Sie den Namen Ihres Unternehmens ein.
- Organisational Unit Name: z. B. "IT"
- Common Name: z. B. "OPC UA Client der Maschine A"
- Email Address:

Wichtig: Im Feld "Alternativer Antragstellername" (Subject Alternative Name) des erstellten Zertifikats muss die IP-Adresse des Client-Programms hinterlegt sein, sonst akzeptiert die CPU das Zertifikat nicht.

Ihre Angaben werden in das Zertifikat eingefügt. Das folgende Bild zeigt die Kommandozeile mit dem Befehl und der Ausgabe von OpenSSL:



```
C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL> req -new -key myKey.key -out MyRequest.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name <2 letter code> [AU]:
```

Der Befehl legt im Verzeichnis C:\demo eine Datei an, die den Certificate Signing Request (CSR) enthält, im Beispiel "myRequest.csr".

Verwendung des CSR

Sie können einen CSR auf zwei Arten verwenden:

- Sie senden den CSR an eine Zertifizierungsstelle (CA): Beachten dabei Sie die Hinweise der jeweiligen Zertifizierungsstelle. Die Zertifizierungsstelle (CA) überprüft Ihre Angaben und Identität (Authentifizierung) und signiert das Zertifikat mit dem privaten Schlüssel der Zertifizierungsstelle. Sie erhalten das signierte X.509-Zertifikat und verwenden dieses Zertifikat zum Beispiel für OPC UA, HTTPS oder Secure OUC (secure open user communication). Ihre Kommunikationspartner überprüfen durch den öffentlichen Schlüssel der Zertifizierungsstelle, ob Ihr Zertifikat wirklich von dieser Stelle herausgegeben und signiert wurde (d. h. die Zertifizierungsstelle Ihre Angaben im Zertifikat bestätigt hat).
- Sie signieren den CSR selbst: Dazu verwenden Sie Ihren privaten Schlüssel. Diese Möglichkeit zeigt der nächste Schritt.

Zertifikat selber signieren

Damit Sie Ihr Zertifikat (Self-Signed Certificate) selber erzeugen und signieren können, geben Sie den folgenden Befehl ein: "x509 -req -days 365 -in myRequest.csr -signkey myKey.key -out myCertificate.crt". "

Das folgende Bild zeigt die Kommandozeile mit dem Befehl und der Ausgabe von OpenSSL:



```
cmd: C:\WINDOWS\system32\cmd.exe - C:\OpenSSL-Win64\bin\openssl.exe
OpenSSL> x509 -req -days 365 -in myRequest.csr -signkey myKey.key -out myCertificate.crt
Signature ok
subject=C=DE/ST=Bayern/L=Augsburg/O=MyCompany/OU=IT/CN=MyName/emailAddress=MyEmail@me.de
Getting Private key
OpenSSL>
```

Der Befehl erzeugt ein X.509-Zertifikat mit den Attributen, die Sie mit dem CSR (im Beispiel "myRequest.csr") übergeben, z. B. mit einer Gültigkeit von einem Jahr (-days 365). Außerdem signiert der Befehl das Zertifikat mit Ihrem privaten Schlüssel (im Beispiel "myKey.key"). Ihre Kommunikationspartner können durch Ihren öffentlichen Schlüssel (in Ihrem Zertifikat enthalten) überprüfen, ob Ihr Zertifikat wirklich von Ihnen kommt. Damit ist auch ausgeschlossen, dass Ihr Zertifikat von einem Angreifer manipuliert wurde.

Bei Self-Signed Zertifikaten bestätigen Sie selbst, dass Ihre Angaben in Ihrem Zertifikat richtig sind. Es gibt keine unabhängige Stelle, die Ihre Angaben überprüft.

9.2.6 Nachrichten gesichert übertragen

Aufbau sicherer Verbindungen bei OPC UA

OPC UA verwendet sichere Verbindungen zwischen Client und Server. Dabei überprüft OPC UA die Identität der Kommunikationspartner. Für die Authentifizierung von Client und Server nutzt OPC UA Zertifikate gemäß X.509-V3 der ITU (International Telecommunication Union). Ausnahme: Bei der Security Policy "Keine Security" wird keine sichere Verbindung aufgebaut.

Message Security Modus

OPC UA verwendet die folgenden Security Policys zum Schutz von Nachrichten:

- Keine Security

Alle Nachrichten sind ungesichert. Um diese Security Policy zu verwenden, bauen Sie eine Verbindung zu einem None-Endpunkt eines Servers auf.

- Signieren

Alle Nachrichten werden signiert. Dadurch lässt sich die Integrität der empfangenen Nachrichten überprüfen. Manipulationen werden erkannt. Um diese Security Policy zu verwenden, bauen Sie eine Verbindung zu einem Sign-Endpunkt eines Servers auf.

- Signieren & Verschlüsseln

Alle Nachrichten werden signiert und verschlüsselt. Dadurch lässt sich die Integrität der empfangenen Nachrichten überprüfen. Manipulationen werden erkannt. Außerdem kann kein Angreifer den Inhalt der Nachricht lesen. (Schutz der Vertraulichkeit). Um diese Security Policy zu verwenden, bauen Sie eine Verbindung zu einem "Signieren & Verschlüsseln"-Endpunkt eines Servers auf.

Die Security Policys sind nach zusätzlich nach den verwendeten Algorithmen benannt. Beispiel: "Basic256Sha256 - Signieren & Verschlüsseln" bedeutet: Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen für 256-Bit-Hashing und 256-Bit-Verschlüsselung.

Erforderliche Schichten

Das folgende Bild zeigt die drei Schichten Transport-Schicht, Secure Channel und Session, die für den Aufbau einer Verbindung stets erforderlich sind.

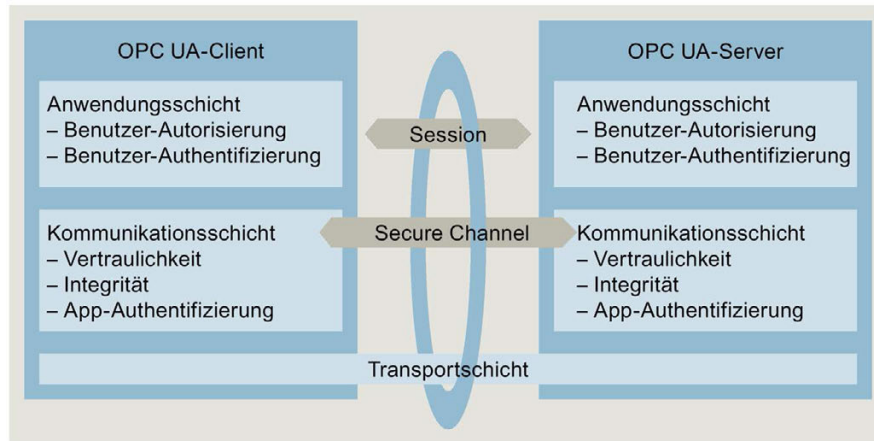


Bild 9-6 Erforderliche Schichten Transport-Schicht, Secure Channel und Session

- **Transportschicht:**

Diese Schicht sendet und empfängt Nachrichten. Dazu verwendet OPC UA ein optimiertes TCP-basiertes Binärprotokoll. Die Transportschicht ist die Grundlage für den nachfolgenden Secure Channel.

- **Secure Channel**

Der Secure Channel erhält von der Transportschicht die empfangenen Daten und leitet sie an die Session weiter. Daten von der Session, die gesendet werden sollen, leitet der Secure Channel an die Transportschicht weiter.

Beim Security-Modus "Signieren" signiert der Secure Channel Daten (Nachrichten), die gesendet werden. Bei ankommenden Nachrichten überprüft der Secure Channel die Signatur, um Manipulationen erkennen zu können.

Bei einer "Signieren & Verschlüsseln" Security Policy signiert und verschlüsselt der Secure Channel die Sendedaten. Empfangene Daten entschlüsselt der Secure Channel. Anschließend überprüft der Secure Channel die Signatur.

Bei der Security Policy "Keine Security" passieren die Nachrichtenpakete den Secure Channel unverändert (die Nachrichten werden im Klartext empfangen und gesendet).

- **Session**

Die Session gibt die Nachrichten vom Secure Channel an die Anwendung weiter, bzw. erhält von der Anwendung Nachrichten, die gesendet werden sollen. Die Anwendung verwendet die Prozesswerte, bzw. stellt die Werte bereit.

Aufbau des Secure Channels

Der Aufbau des Secure Channels läuft folgendermaßen ab:

1. Der Server beginnt mit dem Aufbau des Secure Channels, wenn er eine Aufforderung dazu vom Client erhält. Dieser Request ist entweder signiert, signiert und verschlüsselt, oder die Nachricht wird im Klartext gesendet (Security-Modus des gewählten Server-Endpunkts). Bei "Signieren" und "Signieren & Verschlüsseln" sendet der Client ein "Geheimnis" (eine Zufallszahl) mit dem Request.
2. Der Server validiert das Client-Zertifikat (unverschlüsselt im Request enthalten) und überprüft die Identität des Clients. Vertraut der Server dem Client-Zertifikat, dann
 - entschlüsselt der Server die Nachricht und überprüft die Signatur ("Signieren & Verschlüsseln")
 - oder überprüft nur die Signatur ("Signieren")
 - oder lässt die Nachricht unverändert ("Keine Security").
3. Danach sendet der Server eine Antwort zum Client (gleichermaßen gesichert wie der Request). Im Response ist das Server-Geheimnis enthalten. Aus dem Client- und Server-Geheimnis errechnen Client und Server einen symmetrischen Schlüssel. Damit ist der Secure Channel aufgebaut.

Der symmetrische Schlüssel wird nun für das Signieren und Verschlüsseln von Nachrichten verwendet (anstelle der privaten und öffentlichen Schlüssel von Client und Server).

Aufbau der Session

Der Aufbau der Session läuft folgendermaßen ab:

1. Der Client startet den Session-Aufbau, indem er einen CreateSessionRequest an den Server schickt. Diese Nachricht enthält ein Nonce, eine nur einmal verwendete Zufallszahl. Der Server muss diese Zufallszahl (Nonce) signieren, um zu beweisen, dass er Inhaber des privaten Schlüssels ist. Der private Schlüssel gehört zu dem Zertifikat, den der Server beim Aufbau des Secure Channels verwendet. Diese Nachricht (und alle nachfolgenden) ist entsprechend den Sicherheitseinstellungen des gewählten Server-Endpunkts (ausgewählte Security Policies) gesichert.
2. Der Server antwortet mit der CreateSession Response. Diese Nachricht enthält den öffentlichen Schlüssel des Servers sowie das signierte Nonce. Der Client überprüft das signierte Nonce.
3. Wenn der Server den Test bestanden hat, dann sendet der Client einen SessionActivateRequest an den Server. Diese Nachricht enthält die Angaben, die für die Legitimierung des Anwenders erforderlich sind:
 - entweder Username und Passwort
 - oder das X.509-Zertifikat des Anwenders (in STEP 7 V15 nicht unterstützt)
 - oder keine Daten (wenn ein anonymer Zugang konfiguriert ist).
4. Wenn der Anwender über die notwendigen Rechte verfügt, sendet der Server eine Nachricht an den Client zurück (ActivateSessionResponse). Damit ist die Session aktiviert.

Die sichere Verbindung zwischen OPC UA-Client und -Server ist aufgebaut.

Aufbau einer Verbindung mit PLCopen Funktionsbausteinen

Die PLCopen Spezifikation hat eine Reihe von IEC 61131 Funktionsbausteinen für OPC UA-Clients definiert. Die Anweisung UA_Connect initiiert hierbei sowohl einen Secure Channel als auch eine Session nach oben beschriebenem Muster.

9.3 S7-1500 CPU als OPC UA-Server nutzen

9.3.1 Zugriff auf PLC-Variablen projektieren

9.3.1.1 Schreib- und Leserechte verwalten

PLC-Variablen und DB-Variablen für OPC UA frei geben

OPC UA-Clients können auf PLC-Variablen und DB-Variablen lesend und schreibend zugreifen, wenn die Variablen für OPC UA freigegeben sind (Voreinstellung). Bei freigegebenen Variablen ist das Optionskästchen bei "Erreichbar aus HMI/OPC UA" aktiviert.

Das folgende Beispiel zeigt einen Array-Datenbaustein:

MyDB					
Name	Datentyp	Erreichbar aus HMI/OPC UA	Schreibbar aus HMI/OPC UA	Sichtbar in HMI Engineering	
MyDB	Arr...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[0]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[1]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[2]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[3]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[4]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[5]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[6]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[7]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[8]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
MyDB[9]	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Bild 9-7 PLC-Variablen und DB-Variablen für OPC UA frei geben

Das Array kann von OPC UA-Clients komplett in einem Zug gelesen werden (siehe Knoten adressieren (Seite 141)). Bei allen Komponenten des Arrays sind die Optionskästchen "Erreichbar aus HMI/OPC UA" und "Schreibbar aus HMI/OPC UA" aktiviert.

Folge: OPC UA-Clients dürfen diese Komponenten sowohl lesen als auch schreiben.

Schreibrechte entziehen

Wenn Sie eine Variable gegen Schreibzugriffe schützen wollen, dann deaktivieren Sie bei dieser Variablen die Option "Schreibbar aus HMI/OPC UA". Dadurch entziehen Sie OPC UA-Clients und HMI-Geräten das Schreibrecht.

Folge: Ausschließlich lesende Zugriffe durch OPC UA-Clients und durch HMI-Geräte sind möglich. OPC UA-Clients können dieser Variable keine Werte zuweisen und dadurch keinen Einfluss auf den Ablauf des S7-Programms nehmen.

Schreib- und Leserechte entziehen

Um eine Variable gegen Schreib- und Lesezugriffe zu schützen, deaktivieren Sie bei dieser Variablen die Option "Erreichbar aus HMI/OPC UA" (Häkchen nicht gesetzt). Dadurch entfernt der OPC UA-Server diese Variable aus seinem Adressraum. OPC UA-Clients sehen diese CPU-Variable nicht mehr.

Folge: OPC UA-Clients und HMI-Geräte können diese Variable weder lesen noch schreiben.

Schreib- und Leserechte von Strukturen

Wenn Sie für eine Komponente einer Struktur das Schreib- oder Leserecht entziehen, dann kann die Struktur oder der Datenbaustein nicht mehr als Ganzes beschrieben oder gelesen werden.

Wenn Sie Schreib- und Leserechte einzelnen Komponenten eines PLC-Datentyps (UDT) entziehen, dann sind die Rechte auch in einem auf dem UDT basierenden Datenbaustein entzogen!

Sichtbar in HMI Engineering

Die Option "Sichtbar in HMI Engineering" bezieht sich auf Engineering-Tools von Siemens. Wenn Sie die Option "Sichtbar in HMI Engineering" deaktivieren (Häkchen nicht gesetzt), dann können Sie die Variable nicht mehr in WinCC (TIA Portal) projektieren.

Die Option hat keine Auswirkungen auf OPC UA.

Regeln

- Erlauben Sie in STEP 7 nur dann lesende Zugriffe auf PLC-Variablen und Variablen von Datenbausteinen, wenn es für die Kommunikation zu anderen Systemen (Steuerungen, eingebetteten Systemen, MES) erforderlich ist.
Andere PLC-Variablen sollten Sie nicht frei geben.
- Gewähren Sie nur dann schreibende Zugriffe über OPC UA, wenn Schreibrechte tatsächlich bei bestimmten PLC-Variablen und Variablen von Datenbausteinen erforderlich sind.
- Wenn Sie für alle Elemente eines Datenbausteins die Option "Erreichbar aus HMI/OPC UA" zurückgesetzt haben, dann ist der Datenbaustein für einen OPC UA-Client nicht mehr sichtbar im Adressraum des OPC UA-Servers der S7-1500 CPU.
- Sie können auch zentral den Zugriff auf einen gesamten Datenbaustein verhindern (siehe Wissenswertes zu OPC UA-Clients (Seite 150)). Diese Einstellung "überstimmt" die Einstellungen an den Komponenten im DB-Editor.

9.3.1.2 Schreib- und Leserechte für kompletten DB verwalten

DBs oder DB-Inhalte für OPC UA-Clients verbergen

Ab STEP 7 V15 haben Sie die Möglichkeit, den Zugriff auf einen kompletten Datenbaustein durch einen OPC UA-Client auf einfache Weise zu verhindern.

Auf diese Weise bleiben die Daten des entsprechenden DBs, auch Instanz-DBs von Funktionsbausteinen, für OPC UA-Clients verborgen.

Voreingestellt ist, dass Datenbausteine von OPC UA-Clients lesbar und schreibbar sind.

Vorgehen

Um einen Datenbaustein für OPC UA-Clients komplett zu verbergen bzw. um einen Datenbaustein vor Schreibzugriffen von OPC UA-Clients zu schützen, gehen Sie folgendermaßen vor:

1. Markieren Sie in der Projektnavigation den zu schützenden Datenbaustein.
2. Wählen Sie das Kontextmenü "Eigenschaften".
3. Wählen Sie den Bereich "Attribute".
4. Aktivieren/Deaktivieren Sie die Optionskästchen "DB erreichbar aus OPC UA" nach Ihren Erfordernissen.

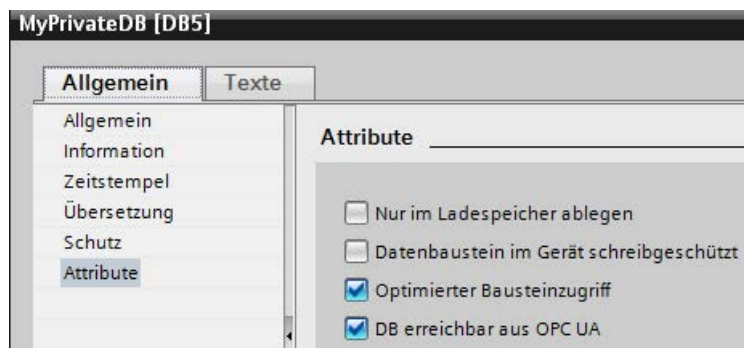


Bild 9-8 DBs oder DB-Inhalte für OPC UA-Clients verbergen

Hinweis

Beeinflussung der Einstellungen im DB-Editor

Wenn Sie über das hier beschriebene DB-Attribut einen DB verbergen, dann sind die Einstellungen an den Komponenten im DB-Editor nicht mehr relevant; einzelne Komponenten können nicht mehr erreicht oder beschrieben werden.

9.3.1.3 Zugriffsmöglichkeiten auf Daten des OPC UA-Servers

Hohe Performance abhängig vom Anwendungsfall

OPC UA ist für die Übertragung vieler Daten in kurzer Zeit ausgelegt. Sie können die Leistung deutlich steigern, wenn Sie nicht auf einzelne PLC-Variablen zugreifen, sondern Arrays und Strukturen als Ganzes lesen und schreiben.

Am schnellsten ist der Zugriff auf Arrays. Deshalb sollten Sie die Daten für OPC UA-Clients in Arrays zusammenfassen.

Empfehlungen für den Zugriff auf den OPC UA-Server durch den OPC UA-Client

- Nutzen Sie für den einmaligen oder seltenen Datenzugriff den normalen Read/Write-Zugriff.
- Nutzen Sie für den zyklischen Zugriff auf wenige Daten (bis ca. alle 5 Sekunden) Subscriptions.

Optimieren Sie am OPC UA-Server die Einstellungen für das kleinste Sendeintervall und das kleinste Abtastintervall.
- Nutzen Sie für einen regelmäßigen (wiederkehrenden) Zugriff auf bestimmte Variablen die Funktionen "RegisteredRead" und "RegisteredWrite".

Gewähren Sie der CPU mehr Kommunikationslast, indem Sie den Wert für "Zyklusbelastung durch Kommunikation" erhöhen. Vergewissern Sie sich, dass Ihre Anwendung mit den geänderten Einstellungen noch ordnungsgemäß funktioniert.

Vorgehensweise zum Anlegen eines Arrays-DBs

Arrays können sie z. B. in globalen Datenbausteinen, im Instanzdatenbaustein eines Funktionsbausteins oder als Array-DB anlegen. Im folgenden ist beschrieben, wie Sie einen Array-DB anlegen.

Um einen Datenbaustein mit einem Array (Array-Datenbaustein) anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Projektnavigation die CPU mit dem OPC UA-Server.
2. Doppelklicken Sie auf "Programmbausteine".
3. Doppelklicken Sie auf "Neuen Baustein hinzufügen".
4. Klicken Sie auf die Schaltfläche "Datenbaustein".
5. Wählen Sie einen eindeutigen Namen für den Datenbaustein oder übernehmen Sie den bereits eingetragenen Namen.
6. Wählen Sie in der Klappliste bei "Typ" den Eintrag "Array-DB".
7. Wählen Sie in der Klappliste bei "Array-Datentyp" den Datentyp für die einzelnen Komponenten des Arrays.
8. Tragen Sie die obere Grenze des Arrays bei "Array-Grenze" ein.
9. Klicken Sie auf die Schaltfläche "OK".

9.3.1.4 XML-Datei mit den freigegebenen PLC-Variablen exportieren

OPC UA-Exportdatei erzeugen

Für die Offline-Projektierung eines OPC UA-Clients können Sie eine XML-Datei nutzen, in der alle freigegebenen PLC-Variablen beschrieben sind.

Die OPC UA-XML-Datei können Sie mit STEP 7 (TIA Portal) aus Ihrem Projekt exportieren; die Datei ist gemäß der OPC UA-Spezifikation aufgebaut.

Um die XML-Datei zu erzeugen und zu exportieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU aus. Klicken Sie dazu auf das Symbol der CPU (z. B. in der Netzsicht).
2. Klicken Sie in den Eigenschaften der CPU auf "Allgemein > OPC UA > Server > Exportieren".
3. Klicken Sie auf die Schaltfläche "OPC UA XML-Datei exportieren":
4. Wählen Sie das Verzeichnis, in dem Sie die Exportdatei speichern wollen.
5. Wählen Sie einen neuen Namen für die Datei. Oder behalten Sie den Namen bei, der bereits eingetragen ist.
6. Klicken Sie auf "Speichern".

Hinweis

Server-Methoden sind nicht in der OPC UA-Exportdatei enthalten.

Alle Arrayelemente separat exportieren

Wenn in den CPU-Eigenschaften "OPC UA > Server > Exportieren" die Option "Alle Arrayelemente als separate Knoten exportieren" aktiviert ist, dann enthält die OPC UA XML-Datei alle Elemente von Arrays jeweils als einzelne XML-Elemente.

Die Option ist in der Voreinstellung aktiviert und nicht änderbar.

In der XML-Datei sind zudem die Arrays selbst jeweils in einem XML-Element beschrieben.

Wenn viele Array-Elemente in einem Array enthalten sind, kann die XML-Datei sehr umfangreich werden.

Tipp

In folgendem FAQ finden Sie einen Konverter, mit dem Sie die Exportdatei in das CSV-Format wandeln können. Sie erhalten damit eine Liste der für OPC UA erreichbaren Variablen der CPU.

Den FAQ finden Sie im Internet

(<https://support.industry.siemens.com/cs/ww/de/view/109742903>).

9.3.2 OPC UA-Server der S7-1500 CPU konfigurieren

9.3.2.1 OPC UA-Server aktivieren

Voraussetzung

- Sie haben eine Runtime-Lizenz für den Betrieb des OPC UA-Servers erworben, siehe Lizenzen für den OPC UA-Server (Seite 193).
- Wenn Sie Zertifikate zur gesicherten Kommunikation nutzen z. B. HTTPS, Secure OUC, OPC UA, dann achten Sie darauf, dass die betroffenen Baugruppen über die aktuelle Uhrzeit und das aktuelle Datum verfügen. Die Baugruppen werten die verwendeten Zertifikate sonst als ungültig und die gesicherte Kommunikation funktioniert nicht.

OPC UA-Server in Betrieb nehmen

In der Grundeinstellung ist der OPC UA-Server der CPU aus Sicherheitsgründen nicht freigegeben: OPC UA-Clients können weder schreibend noch lesend auf die S7-1500 CPU zugreifen.

Um den OPC UA-Server der CPU zu aktivieren, gehen Sie folgendermaßen vor.

1. Wählen Sie die CPU aus. Klicken Sie dazu auf das Symbol der CPU (z. B. in der Netzsicht).
2. Klicken Sie in den Eigenschaften der CPU auf "OPC UA > Server".
3. Aktivieren Sie den OPC UA-Server der CPU.
4. Bestätigen Sie die Sicherheitshinweise.
5. Wählen Sie bei den CPU-Eigenschaften den Bereich "Runtime-Lizenzen" und stellen die erworbene Runtime-Lizenz für den OPC UA-Server ein.
6. Kompilieren Sie das Projekt.
7. Laden Sie das Projekt in die CPU.

Der OPC UA-Server der CPU startet nun.

Einstellungen bleiben gespeichert

Falls Sie den Server bereits aktiviert und Einstellungen vorgenommen hatten, dann gehen diese Einstellungen nicht verloren, wenn Sie den Server deaktivieren. Die Einstellungen sind nach wie vor gespeichert und stehen wieder zur Verfügung, wenn Sie den Server wieder aktivieren.

Applikationsname

Der Applikationsname ist der Name der OPC UA-Applikation (Server). Der Name wird angezeigt unter "OPC UA > Allgemein":

- Die Voreinstellung für den Servernamen lautet: "SIMATIC.S7-1500.OPC-UAServer:PLC1".
- Die Voreinstellung setzt sich aus "SIMATIC.S7-1500.OPC-UAServer:" und dem Namen der CPU zusammen, wie er unter "Allgemein > Produktinformation > Name" gewählt wurde, hier "PLC_1".
- Clients identifizieren den Server über den Applikationsnamen.

Das folgende Beispiel stammt von UaExpert:



Wenn Sie den Server aktiviert haben, können Sie auch einen anderen Namen verwenden, der in Ihrem Projekt aussagekräftig ist.

Applikationsname ändern

Um den Namen des OPC UA-Servers zu ändern, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU aus. Klicken Sie dazu auf das Symbol der CPU (z. B. in der Netzansicht).
2. Klicken Sie in den Eigenschaften der CPU auf "OPC UA > Allgemein".
3. Tragen Sie einen aussagekräftigen Namen ein.

Beachten Sie, dass der Applikationsname auch im Zertifikat eingetragen ist (Subject Alternative Name) und Sie nach Änderung des Applikationsnamens gegebenenfalls ein zuvor erstelltes Zertifikat nochmals erstellen müssen.

9.3.2.2 Zugang zum OPC UA-Server

Server-Adressen

Der OPC UA-Server der S7-1500 CPU ist über alle internen PROFINET-Schnittstellen der CPU (ab Firmware V2.0) erreichbar, jedoch nicht über die PROFINET-Schnittstellen von CP/CM.

Bei SIMATIC S7-1500 SW Controllern ist der OPC UA-Server nur über die PROFINET-Schnittstellen erreichbar, die der Software-PLC zugewiesen sind.

Im Beispiel können über die folgenden URLs (Uniform Resource Locator) Verbindungen zu dem OPC UA-Server der CPU aufgebaut werden:

Erreichbarkeit des Servers	
Server-Adressen:	
Adresse	
opc.tcp://192.168.178.151:4840	
opc.tcp://192.168.1.1:4840	

Die URLs gliedern sich folgendermaßen:

- Protokollkennung "opc.tcp://"
- IP-Adresse
 - 192.168.178.151
Die IP-Adresse, über die OPC UA-Server aus dem Ethernet-Subnetz 192.168.178 erreichbar ist.
 - 192.168.1.1
Die IP-Adresse, über die OPC UA-Server aus dem Ethernet-Subnetz 192.168.1 erreichbar ist.
- TCP-Portnummer
 - Voreinstellung: 4840 (Standardport)
Die Portnummer kann geändert werden, unter "OPC UA > Server > Einstellungen > Port".

Dynamische IP-Adressen

Im folgenden Beispiel ist die IP-Adresse der PROFINET-Schnittstelle [X2] noch nicht festgelegt.

Erreichbarkeit des Servers	
Server-Adressen:	
Adresse	
opc.tcp://192.168.178.151:4840	
opc.tcp://<dynamically>:4840	

In der Tabelle erscheint der Platzhalter "<dynamically>".

Die IP-Adresse dieser PROFINET-Schnittstelle wird später am Gerät gesetzt. Welche IP-Adresse der PROFINET-Schnittstelle [X2] zugewiesen wurde (außerhalb des TIA Portals), erfahren Sie unter anderem, wenn Sie mit einem OPC UA-Client nach verfügbaren OPC UA-Servern suchen. Sie erhalten dann eine Liste der verfügbaren Server mit den IP-Adressen.

Standard-SIMATIC-Server-Schnittstelle aktivieren

Wenn die Option "Standard-SIMATIC-Server-Schnittstelle aktivieren" aktiviert ist, dann stellt der OPC UA-Server der CPU die freigegebenen PLC-Variablen und Server-Methoden den Clients zur Verfügung, wie es in der OPC UA-Spezifikation festgelegt ist.

In der Voreinstellung ist diese Option aktiviert.

Lassen Sie die Option aktiviert, damit OPC UA-Clients die Möglichkeit haben, sich automatisch mit dem OPC UA-Server der CPU zu verbinden und Daten auszutauschen.

Neben der Standard-SIMATIC-Server-Schnittstelle haben Sie die Möglichkeit, weitere Server-Schnittstellen zu importieren und zu aktivieren, siehe OPC UA-Informationsmodelle nutzen (Seite 218).

Unterstützung für OPC UA-Clients mit OPC UA V1.03 und älter

Die OPC UA-Spezifikation (<= V1.03) definiert Mechanismen, um mit Hilfe von TypeDictionaries Datentyp-Definitionen, z. B. für benutzerdefinierte Strukturen (UDTs) von einem Server auslesen zu können.

Zum Liefereinsatz von STEP 7 Professional (TIA Portal) V15 und S7-1500 CPUs mit Firmware Version 2.5 lag die OPC UA-Spezifikation als Release Candidate V1.04 vor. Diese aktualisierte Version der OPC UA-Spezifikation ermöglicht z. B. performanteren Zugriff auf Strukturen.

! WARNUNG

Zugriff auf Daten (z. B. Strukturen) mit Clients, welche die OPC UA-Spezifikation V1.04 nutzen

Der Zugriff über Mechanismen, die in der OPC UA Spezifikation V1.04 definiert sind, wird von S7-1500 CPUs mit Firmware 2.5 nicht unterstützt! Clients, die diese Mechanismen für Zugriffe auf Daten des OPC UA-Servers nutzen, arbeiten möglicherweise mit falschen Werten.

9.3.2.3 Allgemeine Einstellungen des OPC UA-Servers

TCP-Port für OPC UA

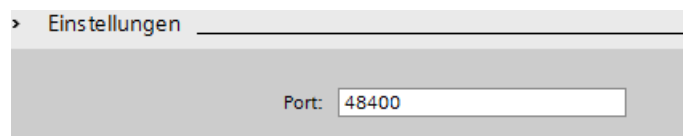
Standardmäßig verwendet OPC UA den TCP-Port 4840.

Sie können jedoch auch einen anderen Port wählen. Eingaben von 1024 bis 49151 sind möglich.

Sie müssen jedoch darauf achten, dass keine Konflikte mit anderen Anwendungen entstehen.

Den gewählten Port müssen OPC UA-Clients beim Verbindungsaufbau verwenden.

Im folgenden Beispiel wurde der Port 48400 gewählt:



Einstellungen für Sessions

- Maximales Timeout für Sessions

In diesem Feld legen Sie fest, wie lange die Zeitspanne höchstens sein darf, bis der OPC UA-Server eine Session ohne Datenaustausch abbaut.

Mögliche Werte zwischen 1 und 600000 Sekunden.

- Maximale Anzahl OPC UA-Sessions

In diesem Feld legen Sie fest, wie viele Sessions der OPC UA-Server der CPU höchstens aufbaut und gleichzeitig betreibt.

Die maximale Anzahl der Sessions ist abhängig von der Leistungsfähigkeit der CPU. Jede Session bindet Ressourcen.

Maximale Anzahl registrierter Knoten

In diesem Feld legen Sie fest, wie viele Knoten (Nodes) der OPC UA-Server höchstens registriert.

Die maximale Anzahl der registrierten Knoten ist abhängig von der Leistungsfähigkeit der CPU und wird beim Projektieren des Feldinhalts angezeigt (Mauszeiger in das Feld setzen). Jede Registrierung bindet Ressourcen.

Hinweis

Keine Fehlermeldung beim Versuch, mehr Knoten zu registrieren als die projektierte maximale Anzahl registrierbarer Knoten

Wenn ein Client zur Laufzeit mehr Knoten registrieren will als die projektierte maximale Anzahl registrierbarer Knoten, dann registriert der Server der S7-1500 CPU nur die projektierte maximale Anzahl. Der Server liefert dem Client ab der projektierten maximalen Anzahl registrierbarer Knoten die regulären String-Node-Ids zurück, so dass der Geschwindigkeitsvorteil durch Registrierung für diese Knoten entfällt. Der Client erhält keine Fehlermeldung.

Achten Sie beim Projektieren auf eine ausreichende Reserve oder lassen Sie den Client vor der Registrierung die maximale Anzahl registrierbarer Knoten ermitteln.

Siehe auch

Welche Ports werden von den verschiedenen Diensten für die Datenübertragung über TCP und UDP verwendet und was ist bei der Verwendung von Routern und Firewalls zu beachten? (<https://support.industry.siemens.com/cs/ww/de/view/8970169>)

9.3.2.4 Einstellungen des Servers für Subscriptions

Subscription statt zyklisches Abfragen

Eine Alternative zum zyklischen Abfragen einer PLC-Variablen (Polling) ist das Beobachten dieses Wertes. Nutzen Sie dazu eine Subscription (Abonnement): Wenn sich der Wert von PLC-Variablen geändert hat, informiert der Server den Client. Siehe "Der OPC UA-Client (Seite 150)".

Ein Server überwacht meist sehr viele PLC-Werte. Deshalb sendet der Server in regelmäßigen Abständen Nachrichten (Notifikation) an den Client, in denen die neuen Werte der PLC-Variablen enthalten sind.

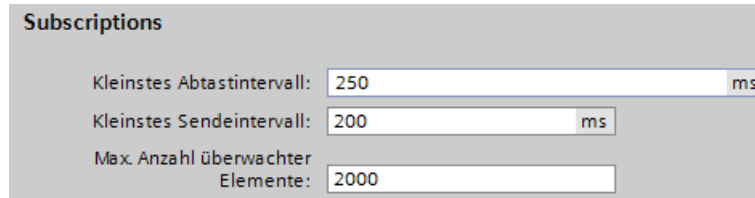
Wie oft sendet der Server Nachrichten?

Beim Anlegen einer Subscription gibt der OPC UA-Client seinen Wunsch an, in welchen Abständen der OPC UA-Client bei Wertänderung die neuen Werte erhalten möchte. Um die Kommunikationslast durch OPC UA zu begrenzen, legen Sie den Wert für den zeitlichen Abstand der Nachrichten fest, der nicht unterschritten werden darf. Dazu verwenden Sie die Parameter für das kleinste Sendeintervall und das kleinste Abtastintervall.

Kleinstes Sendeintervall

Bei "Kleinstes Sendeintervall" stellen Sie die zeitlichen Abstände (Intervalle) ein, in denen der Server bei Wertänderung eine Nachricht mit den neuen Werten an den Client schickt.

Im folgenden Bild wird als "Kleinstes Abtastintervall" der Wert 250 ms verwendet. Als "Kleinstes Sendeintervall" ist der Wert von 200 ms eingetragen.



Subscriptions	
Kleinstes Abtastintervall:	250 ms
Kleinstes Sendeintervall:	200 ms
Max. Anzahl überwachter Elemente:	2000

Im Beispiel sendet der OPC UA-Server bei Wertänderung alle 200 ms eine neue Nachricht, falls der OPC UA-Client eine Aktualisierung fordert.

Fordert der OPC UA-Client z. B. eine Aktualisierung alle 1000 ms, dann sendet der OPC UA-Server auch nur einmal in 1000 ms (eine Sekunde) eine Nachricht mit den neuen Werten.

Wenn der Client eine Aktualisierung alle 100 ms fordert, dann sendet der Server trotzdem nur alle 200 ms (kleinstes Sendeintervall).

Kleinstes Abtastintervall

Bei "Kleinstes Abtastintervall" stellen Sie die zeitlichen Abstände (Intervalle) ein, in denen der OPC UA-Server den Wert einer CPU-Variablen erfasst und mit dem bisherigen Wert vergleicht, um eine Wertänderung festzustellen.

Ist das Abtastintervall kleiner gewählt als das Sendeintervall und fordert ein OPC UA-Client für bestimmte PLC-Variablen eine so hohe Abtastrate, dann können pro Sendeintervall zwei oder mehr Werte anfallen.

In diesem Fall schreibt der OPC UA-Server die Wertänderungen in die Warteschlange und sendet nach Ablauf des Sendeintervalls alle Wertänderungen an den Client. Wenn mehr Wertänderungen im Sendeintervall anfallen als in die Warteschlange passen, dann überschreibt der OPC UA-Server die ältesten Werte (abhängig von der eingestellten "Discard Policy", die Option "Discard Oldest" muss in diesem Fall aktiviert sein). Die aktuellsten Werte werden an den Client gesendet.

Maximale Anzahl überwachter Elemente (Monitored Items)

In diesem Feld legen Sie die maximale Anzahl an Elemente fest, die der OPC UA-Server der CPU gleichzeitig auf Wertänderung überwacht.

Die Überwachung bindet Ressourcen. Die maximale Anzahl überwachter Elemente ist abhängig von der verwendeten CPU.

9.3.2.5 Handling der Client- und Server-Zertifikate

Eine gesicherte Verbindung zwischen OPC UA-Server und einem OPC UA-Client kommt nur dann zu Stande, wenn sich der Server gegenüber dem Client ausweisen kann. Dazu dient das Zertifikat des Servers.

Zertifikat des OPC UA-Servers

Wenn Sie den OPC UA-Server aktiviert und die Sicherheitshinweise bestätigt haben, erzeugt STEP 7 automatisch das Zertifikat für den Server und speichert es im lokalen Zertifikatsverzeichnis der CPU. Dieses Verzeichnis können Sie mit dem lokalen Zertifikatsmanager der CPU einsehen und verwalten (Zertifikate exportieren oder löschen).

Das folgende Bild zeigt den lokalen Zertifikatsmanager der CPU mit dem automatisch erzeugten Zertifikat für den OPC UA-Server:



Bild 9-9 Lokaler Zertifikatsmanager der CPU

Alternativ können Sie auch selbst ein Server-Zertifikat erzeugen.

Das Zertifikat des Servers wird während des Aufbaus einer Verbindung vom Server zum Client übertragen, der Client überprüft das Zertifikat.

Der Client-Anwender entscheidet, ob er dem Zertifikat des Servers vertraut

Auf der Client-Seite muss nun der Anwender entscheiden, ob er dem Server-Zertifikat vertraut. Wenn der Anwender dem Server-Zertifikat vertraut, dann speichert der Client das Server-Zertifikat in seinem Verzeichnis, das die vertrauenswürdigen Server-Zertifikate enthält.

Das folgende Beispiel zeigt einen Dialog des Clients "UA Sample Client". Wenn der Anwender auf die Schaltfläche "Ja" klickt, vertraut der Client dem Server-Zertifikat:

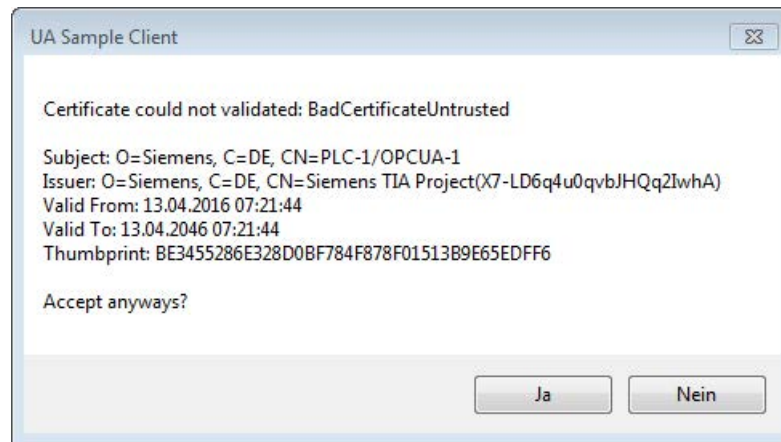


Bild 9-10 Dialog des Clients "UA Sample Client"

Siehe auch

Server-Zertifikate mit STEP 7 erzeugen (Seite 187)

Nachrichten gesichert übertragen (Seite 165)

Woher kommt das Zertifikat eines Clients?

Wenn Sie UA-Clients von Herstellern oder der OPC Foundation verwenden, wird bei der Installation oder beim ersten Programmaufruf automatisch ein Client-Zertifikat erzeugt. Diese Zertifikate müssen Sie über den globalen Zertifikatsmanager in STEP 7 importieren und für die jeweilige CPU verwenden (wie oben gezeigt).

Wenn Sie selbst einen OPC UA-Client programmieren, dann können Sie Zertifikate programmtechnisch erstellen, siehe "Instanz-Zertifikat für den Client (Seite 147)". Oder Sie erzeugen Zertifikate mit Tools, zum Beispiel mit OpenSSL oder dem Zertifikate-Generator der OPC Foundation:

- Wie Sie bei OpenSSL vorgehen, lesen Sie hier: "PKI-Schlüsselpaare und Zertifikate selbst erzeugen (Seite 162)".
- Wie Sie mit dem Zertifikate-Generator der OPC Foundation arbeiten, lesen Sie hier: "Selbst-signierte Zertifikate erzeugen (Seite 160)".

Client-Zertifikate dem Server bekanntmachen

Client-Zertifikate müssen Sie dem Server zur Verfügung stellen, damit eine gesicherte Verbindung aufgebaut werden kann.

Dazu gehen Sie folgendermaßen vor:

1. Aktivieren Sie im lokalen Zertifikatsmanager des Servers die Option "Globale Security-Einstellungen für den Zertifikatsmanager verwenden". Dadurch ist der globale Zertifikatsmanager verfügbar.

Sie finden diese Option in den Eigenschaften der CPU, die als Server dient, unter "Schutz & Security > Zertifikatsmanager".

Falls dieses Projekt noch nicht geschützt ist, dann klicken Sie in der "Projektnavigation" von STEP 7 unter "Security-Einstellungen > Einstellungen" auf die Schaltfläche "Dieses Projekt schützen" und melden Sie sich an.

STEP 7 zeigt nun in der "Projektnavigation" unter "Security-Einstellungen" der Eintrag "Globale Security-Einstellungen" an.

2. Doppelklicken Sie auf "Globale Security-Einstellungen".

3. Doppelklicken Sie auf "Zertifikatsmanager".

STEP 7 öffnet den globalen Zertifikatsmanager.

4. Klicken Sie auf das Register "Gerätezertifikate".

5. Klicken Sie mit der rechten Maustaste im Register auf eine freie Fläche (nicht auf ein Zertifikat).

6. Wählen Sie aus dem Kontextmenu den Eintrag "Importieren".

Der Dialog zum Importieren von Zertifikaten wird angezeigt.

7. Wählen Sie das Client-Zertifikat aus, dem der Server vertrauen soll.

8. Klicken Sie auf die Schaltfläche "Öffnen", um das Zertifikat zu importieren.

Das Zertifikat des Clients ist nun im globalen Zertifikatsmanager enthalten.

Merken Sie sich die ID des gerade importierten Client-Zertifikats.

9. Klicken Sie nun in den Eigenschaften der CPU, die als Server dient, auf das Register "Allgemein".

10. Klicken Sie auf den Bereich "OPC UA > Server > Security > Secure Channel".

11. Scrollen Sie im Dialog "Secure Channel" nach unten zum Abschnitt "Vertrauenswürdige Clients".

12. Doppelklicken Sie in der Tabelle auf die leere Zeile mit "<neu hinzufügen>". In der Zeile wird eine Schaltfläche mit drei Punkten angezeigt.

13. Klicken Sie auf diese Schaltfläche.

14. Wählen Sie das Client-Zertifikat aus, das Sie importiert haben.

15. Klicken Sie auf die Schaltfläche mit dem grünen Häkchen.

16. Übersetzen Sie das Projekt.

17. Laden Sie die Konfiguration in die S7-1500 CPU.

Ergebnis

Der Server vertraut nun dem Client. Wenn außerdem das Server-Zertifikat als vertrauenswürdig gilt, dann können Server und Client eine gesicherte Verbindung aufbauen.

Client-Zertifikate automatisch akzeptieren

Wenn Sie die Option "Client-Zertifikate zur Laufzeit automatisch akzeptieren" aktivieren (unterhalb der Liste "Vertrauenswürdige Clients"), dann akzeptiert der Server alle Client-Zertifikate.

ACHTUNG

Einstellung nach der Inbetriebnahme

Um Sicherheitsrisiken zu vermeiden, deaktivieren Sie die Option "Client-Zertifikate zur Laufzeit automatisch akzeptieren" wieder nach der Inbetriebnahme.

Security-Einstellungen des Servers konfigurieren

Das folgende Bild zeigt die verfügbaren Security-Einstellungen des Servers zum Signieren und Verschlüsseln von Nachrichten.

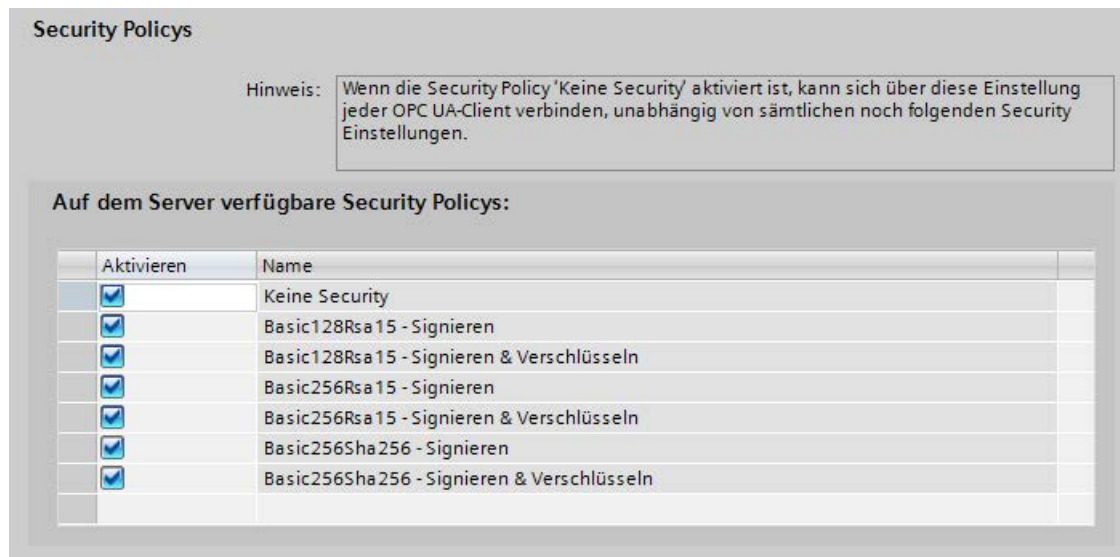


Bild 9-11 Security-Einstellungen des Servers konfigurieren

Per Voreinstellung wird ein Server-Zertifikat erstellt, welches SHA256-Signierung nutzt. Die folgenden Security Policys sind freigegeben:

- Keine
Ungesicherter Endpoint

Hinweis

Nicht erwünschte Security Policys deaktivieren

Wenn Sie bei den Secure-Channel-Einstellungen des S7-1500 OPC UA-Servers alle Security Policys aktiviert haben (Voreinstellung) - also auch den Endpunkt "Keine Security" - dann ist der Datenverkehr zwischen Server und Client auch ungesichert möglich (weder signiert noch verschlüsselt). Die Identität des Clients bleibt bei "Keine Security" unbekannt. Jeder OPC UA-Client kann sich dann mit dem Server verbinden, unabhängig von sämtlichen noch folgenden Security-Einstellungen.

Achten Sie bei der Projektierung des OPC UA-Servers darauf, dass nur Security Policys aktiviert sind, die mit dem Schutzkonzept für Ihre Maschine oder Anlage vereinbar sind. Alle anderen Security Policys sind zu deaktivieren.

Empfehlung: Verwenden Sie, wenn möglich, die Einstellung "Basic256Sha256".

- Basic128Rsa15 - Signieren
Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen, die den Hash-Algorithmus RSA15 und 128-Bit-Verschlüsselung verwenden.
Dieser Endpoint sichert die Integrität der Daten durch Signieren.
- Basic128Rsa15 - Signieren & Verschlüsseln
Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen, die den Hash-Algorithmus RSA15 und 128-Bit-Verschlüsselung verwenden.
Dieser Endpoint sichert die Integrität und Vertraulichkeit der Daten durch Signieren und Verschlüsseln.
- Basic256Rsa15 - Signieren
Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen, die den Hash-Algorithmus RSA15 und 256-Bit-Verschlüsselung verwenden.
Dieser Endpoint sichert die Integrität der Daten durch Signieren.
- Basic256Rsa15 - Signieren & Verschlüsseln
Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen, die den Hash-Algorithmus RSA15 und 256-Bit-Verschlüsselung verwenden.
Dieser Endpoint sichert die Integrität und Vertraulichkeit der Daten durch Signieren und Verschlüsseln.
- Basic256Sha256 - Signieren
Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen für 256-Bit-Hashing und 256-Bit-Verschlüsselung.
Dieser Endpoint sichert die Integrität der Daten durch Signieren.
- Basic256Sha256 - Signieren & Verschlüsseln
Gesicherter Endpoint, unterstützt eine Reihe von Algorithmen für 256-Bit-Hashing und 256-Bit-Verschlüsselung.
Dieser Endpoint sichert die Integrität und Vertraulichkeit der Daten durch Signieren und Verschlüsseln.

Um eine Security-Einstellung freizugeben, klicken Sie auf das Kästchen in der jeweiligen Zeile.

Hinweis

Wenn Sie die Einstellungen "Basic256Sha256 -Signieren" und "Basic256Sha256 -Signieren & Verschlüsseln" nutzen, dann müssen OPC UA-Server und OPC UA-Clients "SHA256"-signierte Zertifikate verwenden.

Bei den Einstellungen "Basic256Sha256 -Signieren" und "Basic256Sha256 -Signieren & Verschlüsseln" signiert die Zertifizierungsstelle von STEP 7 die Zertifikate automatisch mit "SHA256".

9.3.2.6 Handling der Client-Zertifikate der S7-1500 CPU

Woher kommt das Zertifikat des Clients?

Wenn Sie den OPC UA-Client einer S7-1500 CPU nutzen (OPC UA-Client aktiviert), dann können Sie mit STEP 7 ab V15 für diese Clients Zertifikate erzeugen wie in den folgenden Abschnitten beschrieben ist.

Wenn Sie UA-Clients von Herstellern oder der OPC Foundation verwenden, wird bei der Installation oder beim ersten Programmaufruf automatisch ein Client-Zertifikat erzeugt. Diese Zertifikate müssen Sie über den globalen Zertifikatsmanager in STEP 7 importieren und für die jeweilige CPU verwenden.

Wenn Sie selbst einen OPC UA-Client programmieren, dann können Sie Zertifikate programmtechnisch erstellen, siehe "Instanz-Zertifikat für den Client (Seite 147)". Oder Sie erzeugen Zertifikate mit Tools, zum Beispiel mit OpenSSL oder dem Zertifikate-Generator der OPC Foundation:

- Wie Sie bei OpenSSL vorgehen, lesen Sie hier: "PKI-Schlüsselpaare und Zertifikate selbst erzeugen (Seite 162)".
- Wie Sie mit dem Zertifikate-Generator der OPC Foundation arbeiten, lesen Sie hier: "Selbst-signierte Zertifikate erzeugen (Seite 160)".

Zertifikat des OPC UA-Clients der S7-1500 CPU

Eine gesicherte Verbindung zwischen OPC UA-Server und einem OPC UA-Client kommt nur dann zu Stande, wenn der Server das Zertifikat des Clients als vertrauenswürdig einstuft.

Dazu müssen Sie dem Server das Client-Zertifikat bekanntmachen.

Die folgenden Abschnitte beschreiben, wie Sie zunächst für den OPC UA-Client der S7-1500 CPU ein Zertifikat erzeugen und es dann dem Server zur Verfügung stellen.

1. Zertifikat für den Client erzeugen und exportieren

Für eine gesicherte Verbindung müssen Sie ein Client-Zertifikat erzeugen und das Zertifikat exportieren.

Dazu gehen Sie folgendermaßen vor:

1. In der "Projektnavigation" wählen Sie die CPU aus, die als Client fungiert.
2. Doppelklicken Sie auf "Gerätekongfiguration"
3. In den Eigenschaften der CPU klicken Sie auf "Schutz & Security > Zertifikatsmanager".
4. In der Tabelle "Gerätezertifikate" doppelklicken Sie auf "<neu hinzufügen>".

STEP 7 öffnet einen Dialog.

5. Klicken Sie auf die Schaltfläche "Hinzufügen".
6. Bei "Verwendungszweck" wählen Sie aus der Liste den Eintrag "OPC UA-Client".

Achtung:

Unter "Alternativer Name des Zertifikatsinhabers (SAN)" muss die IP-Adressen eingetragen sein, unter der die CPU in Ihrer Anlage erreichbar ist.

Sie müssen also die IP-Schnittstelle der CPU konfigurieren, bevor Sie ein Client-Zertifikat erzeugen.

7. Klicken Sie auf "OK".

STEP 7 zeigt nun das Client-Zertifikat in der Tabelle "Gerätezertifikate" an.

8. Klicken Sie mit der rechten Maustaste auf diese Zeile und wählen Sie aus dem Kontextmenü den Eintrag "Zertifikat exportieren".
9. Wählen Sie ein Verzeichnis aus, in dem Sie das Client-Zertifikat speichern.

2. Das Client-Zertifikat dem Server bekanntmachen

Das Client-Zertifikat müssen Sie dem Server zur Verfügung stellen, damit eine gesicherte Verbindung aufgebaut werden kann.

Dazu gehen Sie folgendermaßen vor:

1. Aktivieren Sie im lokalen Zertifikatsmanager des Servers die Option "Globale Security-Einstellungen für den Zertifikatsmanager verwenden". Dadurch ist der globale Zertifikatsmanager verfügbar.

Sie finden diese Option in den Eigenschaften der CPU, die als Server dient, unter "Schutz & Security > Zertifikatsmanager".

Falls dieses Projekt noch nicht geschützt ist, dann klicken Sie in der "Projektnavigation" von STEP 7 unter "Security-Einstellungen > Einstellungen" auf die Schaltfläche "Dieses Projekt schützen" und melden Sie sich an.

STEP 7 zeigt nun in der "Projektnavigation" unter "Security-Einstellungen" der Eintrag "Globale Security-Einstellungen" an.

2. Doppelklicken Sie auf "Globale Security-Einstellungen".
3. Doppelklicken Sie auf "Zertifikatsmanager".

STEP 7 öffnet den globalen Zertifikatsmanager.

4. Klicken Sie auf das Register "Gerätezertifikate".
5. Klicken Sie mit der rechten Maustaste im Register auf eine freie Fläche (nicht auf ein Zertifikat).
6. Wählen Sie aus dem Kontextmenu den Eintrag "Importieren".
Der Dialog zum Importieren von Zertifikaten wird angezeigt.
7. Wählen Sie das Client-Zertifikat aus, dem der Server vertrauen soll.
8. Klicken Sie auf die Schaltfläche "Öffnen", um das Zertifikat zu importieren.
Das Zertifikat des Clients ist nun im globalen Zertifikatsmanager enthalten.
Merken Sie sich die ID des gerade importierten Client-Zertifikats.
9. Klicken Sie nun in den Eigenschaften der CPU, die als Server dient, auf das Register "Allgemein".
10. Klicken Sie auf den Bereich "OPC UA > Server > Security > Secure Channel".
11. Scrollen Sie im Dialog "Secure Channel" nach unten zum Abschnitt "Vertrauenswürdige Clients".
12. Doppelklicken Sie in der Tabelle auf die leere Zeile mit "<neu hinzufügen>". In der Zeile wird eine Schaltfläche mit drei Punkten angezeigt.
13. Klicken Sie auf diese Schaltfläche.
14. Wählen Sie das Client-Zertifikat aus, das Sie importiert haben.
15. Klicken Sie auf die Schaltfläche mit dem grünen Häkchen.
16. Übersetzen Sie das Projekt.
17. Laden Sie die Konfiguration in die S7-1500 CPU.

Ergebnis

Der Server vertraut nun dem Client. Wenn außerdem das Server-Zertifikat als vertrauenswürdig gilt, dann können Server und Client eine gesicherte Verbindung aufbauen.

9.3.2.7 Server-Zertifikate mit STEP 7 erzeugen

Die folgende Beschreibung zeigt die Vorgehensweise zur Erzeugung neuer Zertifikate mit STEP 7 und gilt prinzipiell für verschiedene Verwendungen der Zertifikate. Je nachdem, über welchen Bereich der CPU-Eigenschaften Sie den folgenden Dialog starten, stellt STEP 7 den passenden Verwendungszweck bereits ein - hier "OPC UA-Client & -Server".

Empfehlung: Um die volle Funktionalität für die Security des OPC UA-Servers zu nutzen, verwenden Sie die globalen Security-Einstellungen.

Die globalen Security-Einstellungen aktivieren Sie in den CPU-Eigenschaften, Bereich "Schutz & Security > Zertifikatsmanager".

Server-Zertifikate individuell anpassen

STEP 7 erzeugt automatisch ein Zertifikat für den OPC UA-Server der S7-1500, wenn Sie den Server aktivieren (siehe "OPC UA-Server aktivieren (Seite 173)"). Dabei verwendet STEP 7 für die Parameter des Zertifikats voreingestellte Werte. Wenn Sie die Parameter ändern wollen, dann gehen Sie folgendermaßen vor:

1. Klicken Sie in den Eigenschaften der CPU unter "Allgemein > OPC UA > Server > Security > Secure Channel > Server-Zertifikat" auf die Drei-Punkte-Schaltfläche. Ein Dialog wird eingeblendet, der lokal vorhandene Zertifikate anzeigt.
2. Klicken Sie auf die Schaltfläche "Hinzufügen".
3. Der Dialog zum Erzeugen neuer Zertifikate wird angezeigt (folgendes Bild). Die Werte für ein Beispiel sind bereits eingetragen:

Neues Zertifikat erzeugen

CA

Auswählen, wie das neue Zertifikat signiert werden soll:

☐ Selbstsigniert

☒ Von Zertifizierungsstelle signiert

CA-Name: 1: Siemens TIA Project(Az5qDcOOukuE294)

Zertifikat-Parameter

Geben Sie die Parameter für das neue Zertifikat ein:

Zertifikatsinhaber: PLC-1/OPCUA-1-6

Signatur: sha256RSA

Gültig von: 11. Oktober. 2017 08:53:09

Gültig bis: 11. Oktober. 2037 00:00:00

Verwendungszweck: OPC UA-Server

Alternativer Name des Zertifikatsinhabers (SAN):

Typ	Wert
URI	urn:SIMATIC.S7-1500.OPC-U...
IP	192.168.0.1
IP	192.168.1.1
Neu hin...	

OK Abbrechen

Bild 9-12 Server-Zertifikate individuell anpassen

4. Verwenden Sie andere Parameter, falls dies nach den Sicherheitsvorgaben in Ihrem Unternehmen oder des Auftraggebers erforderlich ist.

Erläuterung der Felder für die Zertifikatserzeugung

- CA

Wählen Sie aus, ob das Zertifikat selbst-signiert sein soll oder von einem der CA-Zertifikate des TIA Portals. Die Zertifikate sind unter "Zertifikate bei OPC UA (Seite 159)" beschrieben. Wenn Sie ein Zertifikat erzeugen wollen, das von einem der CA-Zertifikate des TIA-Portals signiert sein soll, dann muss das Projekt geschützt sein und Sie müssen als Benutzer mit den erforderlichen Funktionsrechten angemeldet sein. Informationen hierzu erhalten Sie unter "Grundlagen der Benutzerverwaltung im TIA Portal".
- Zertifikatsinhaber

Die Voreinstellung setzt sich zusammen aus dem Namen des Projekts und "\OPCUA-1". Im Beispiel lautet der Projekt-Name "PLC1". In den Eigenschaften der CPU stellen Sie den Projektnamen ein unter "Allgemein > Projektinformation > Name". Behalten Sie die Voreinstellung bei oder tragen Sie bei "Zertifikatsinhaber" einen anderen Namen für den OPC UA-Server ein, der in Ihrem Projekt aussagekräftiger ist.
- Signatur

Wählen Sie hier das Hash- und Verschlüsselungsverfahren aus, das beim Signieren des Server-Zertifikats verwendet werden soll. Die folgenden Einträge sind verfügbar:

 - "sha1RSA",
 - "sha256RSA".
- Gültig von

Tragen Sie hier das Datum und die Uhrzeit ein für den Beginn der Gültigkeit des Server-Zertifikats.
- Gültig bis

Tragen Sie hier das Datum und die Uhrzeit ein für das Ende der Gültigkeit des Server-Zertifikats. Achten Sie darauf, dass das Zertifikat nicht nur ein Jahr oder wenige Jahre gültig ist. Im Beispiel ist das Zertifikat 30 Jahre gültig. Aus Sicherheitsgründen sollten Sie allerdings das Zertifikat in viel kürzeren Abständen erneuern. Die lange Gültigkeit überlässt aber Ihnen die Entscheidung, wann dazu der passende Zeitpunkt ist, etwa wenn eine Anlage gewartet werden muss.

- Verwendungszweck

Voreingestellt ist "OPC UA-Client & -Server". Behalten Sie diese Voreinstellung für den OPC UA-Server bei. Der Dialog "Neues Zertifikat erzeugen" kann von mehreren Stellen aus in STEP 7 aufgerufen werden. Wenn Sie zum Beispiel diesen Dialog für den Webserver der CPU aufrufen, dann wird unter "Verwendungszweck" "Webserver" eingetragen. In der Klappliste zum Verwendungszweck sind die folgenden Einträge verfügbar:

- "OPC UA-Client"
- "OPC UA-Client & -Server"
- "OPC UA-Server"
- "TLS"
- "Webserver"

- Alternativer Name des Zertifikatsinhabers

Im Beispiel oben ist eingetragen: "URI:urn:SIMATIC.S7-1500.OPC-UAServer:PLC1,IP:192.168.178.151,IP:192.168.1.1". Gültig wäre auch der folgende Eintrag: "IP: 192.168.178.151, IP: 192.168.1.1". Wichtig ist, dass hier die IP-Adressen eingetragen sind, über die der OPC UA-Server der CPU erreichbar ist (siehe "Zugang zum OPC UA-Server (Seite 175)"). Dadurch können OPC UA-Clients überprüfen, ob tatsächlich eine Verbindung zum OPC UA-Server der S7-1500 aufgebaut werden soll, oder möglicherweise ein Angreifer versucht, dem OPC UA-Client von einem anderen PC aus manipulierte Werte zuzusenden.

9.3.2.8 Authentifizierung des Benutzers

Arten der Benutzer-Authentifizierung

Sie können beim OPC UA-Server der S7-1500 einstellen, wie sich ein Benutzer des OPC UA-Clients legitimieren muss, wenn er auf den Server zugreifen will.

Dazu gibt es die folgenden Möglichkeiten:

- **Gast-Authentifizierung**

Der Anwender muss seine Berechtigung nicht nachweisen (anonymer Zugang). Der OPC UA-Server überprüft nicht die Berechtigung des Client-Anwenders.

Wenn Sie diese Art der Benutzer-Authentifizierung nutzen wollen, dann wählen Sie unter "OPC UA > Server > Security > Benutzer-Authentifizierung" die Option "Gast-Authentifizierung aktivieren".

Hinweis

Um die Security zu erhöhen, sollten Sie den Zugriff auf den OPC UA-Server nur mit Benutzer-Authentifizierung erlauben!

- **Authentifizierung über Benutzername und Passwort**

Der Anwender muss seine Berechtigung nachweisen (kein anonymer Zugang). Der OPC UA-Server überprüft, ob der Client-Anwender berechtigt ist, auf den Server zuzugreifen. Als Nachweis gilt der Benutzername mit dem richtigen Passwort.

Wenn Sie diese Art der Benutzer-Authentifizierung nutzen wollen, dann wählen Sie unter "OPC UA > Server > Security > Benutzer-Authentifizierung" die Option "Authentifizierung über Benutzername und Passwort aktivieren".

Deaktivieren Sie die Gast-Authentifizierung.

Tragen Sie in der Tabelle "Benutzerverwaltung" die Anwender ein.

Klicken Sie dazu jeweils auf den Eintrag "<Neuen Benutzer hinzufügen>" und fügen Sie den Benutzernamen und das zugehörige Passwort ein. Sie können maximal 21 Benutzer hinzufügen.

- **Zusätzliche Benutzerverwaltung über die Security-Einstellungen des Projekts**

Wenn Sie diese Option aktivieren, dann wird die Benutzerverwaltung des geöffneten Projekts auch für die Benutzer-Authentifizierung des OPC UA-Servers verwendet: Bei OPC UA sind dann dieselben Benutzernamen und Passwörter gültig wie im aktuellen Projekt.

Um die Benutzerverwaltung des Projekts zu aktivieren, gehen Sie folgendermaßen vor:

- Klicken Sie in der "Projektnavigation" auf "Security-Einstellungen > Einstellungen".
- Klicken Sie auf Schaltfläche "Dieses Projekt schützen".
- Tragen Sie Ihren Benutzernamen und Ihr Passwort ein.
- Unter "Security-Einstellungen > Benutzer und Rollen" tragen Sie weitere Benutzer ein.

- Wenn Sie in Ihrem Projekt einen weiteren OPC UA-Server projektieren, dann aktivieren Sie in dem Projekt ebenfalls die Option "Zusätzliche Benutzerverwaltung über die Security-Einstellungen des Projekts aktivieren". Dadurch ist eine erneute Eingabe von Benutzernamen und Passwörtern unnötig.

9.3.2.9 Benutzer und Rollen mit OPC UA-Funktionsrechten

Folgende Option zur Benutzer-Authentifizierung greift auf zentrale Projekteinstellungen für Projektbenutzer zurück:

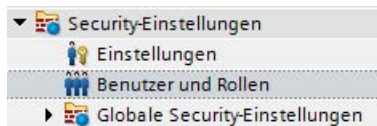
Beim Parametrieren der CPU-Eigenschaften (OPC UA > Server > Security > Benutzer-Authentifizierung). Dort ist es die Option "Zusätzliche Benutzerverwaltung über die Security-Einstellungen des Projekts aktivieren".

Voraussetzung

Um die Security-Einstellungen bearbeiten zu können, muss das Projekt geschützt sein und Sie sind mit ausreichenden Rechten, z. B. als Administrator, angemeldet.

Einstellungen in der Projektnavigation > "Security-Einstellungen"

Die zentralen Benutzereinstellungen und Rollen erreichen Sie im geschützten Projekt in der Projektnavigation, Bereich "Security-Einstellungen". Hier definieren Sie zentral Benutzer mit ihrem Benutzernamen, Passwort sowie Funktionsrechten. Diese Einstellungen können Sie an anderer Stelle einfach wiederverwenden.



Wiederverwenden zentraler Security-Einstellungen

Beispiel für Wiederverwendung:

- Benutzer-Auswahl für die Benutzer-Authentifizierung beim OPC UA-Server
Bei dieser Einstellung teilen Sie dem Server mit, welcher Client (Benutzer) mit welchem Benutzernamen und welchem Passwort überhaupt auf den Server zugreifen darf.

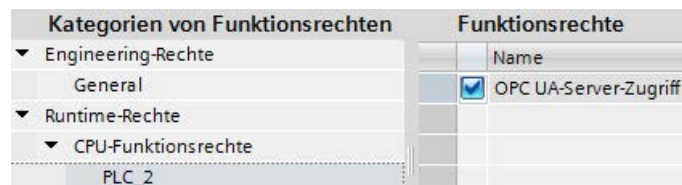
Funktionsrechte für den OPC UA-Server

Für die Benutzer der Server-Funktionalität einer S7-1500 CPU müssen auch die entsprechenden Funktionsrechte für den Server aktiviert sein! Es reicht nicht aus, nur Benutzernamen und Passwort zentral zu hinterlegen.

Ein Beispiel erläutert diese Art der Rechteverwertung.

1. Sie definieren im Bereich "Security-Einstellungen > Benutzer und Rollen" eine neue Rolle im Register "Rollen" mit dem Namen z. B. "PLC-opcua-role-all-inclusive".
2. Im Bereich "Kategorien von Funktionsrechten" navigieren Sie zu den Runtime-Rechten, dann zu den CPU-Funktionsrechten und markieren darunter die CPU, deren Funktionsrechte Sie einstellen wollen, z. B. PLC_2.
3. Im Bereich "Funktionsrechte" finden Sie folgendes Funktionsrecht:
 - OPC UA Server-Zugriff

Dieses Funktionsrecht wirkt am OPC UA-Server der S7-1500 CPU. Nur wenn diese Option markiert ist, hat ein Benutzer des Servers der CPU PLC_2, dem die Rolle "PLC-opcua-role-all-inclusive" zugewiesen ist, folgendes Recht: Er erzwingt für den Aufbau einer Session mit dem Server vom Client die Authentifizierung mit einem der zentral definierten (und in die CPU geladenen) Benutzernamen und zugehörigem Passwort.



Die Rolle "PLC-opcua-role-all-inclusive" müssen Sie noch den entsprechenden Benutzern zuweisen (Register "Benutzer" im Bereich "Security-Einstellungen" in der Projektnavigation).

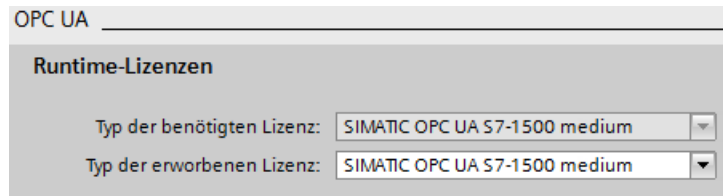
9.3.2.10 Lizenzen für den OPC UA-Server

Runtime-Lizenzen

Für den Betrieb des OPC UA-Servers der S7-1500 CPU ist eine Lizenz erforderlich. Der Typ der erforderlichen Lizenz ist abhängig von der Leistung der jeweiligen CPU. Die folgenden Lizenz-Typen werden unterschieden:

- SIMATIC OPC UA S7-1500 small (erforderlich für CPU 1511, CPU 1512, CPU 1513, ET 200SP CPUs, CPU 1515SP PC)
- SIMATIC OPC UA S7-1500 medium (erforderlich für CPU 1515, CPU 1516, Softwarecontroller CPU 1507, CPU 1516pro-2PN)
- SIMATIC OPC UA S7-1500 large (erforderlich für CPU 1517, CPU 1518)

Der erforderliche Lizenz-Typ wird angezeigt unter "Eigenschaften > Allgemein > Runtime-Lizenzen > OPC-UA > Typ der benötigten Lizenz":



Um den Erwerb der erforderlichen Lizenz zu bestätigen, gehen Sie folgendermaßen vor:

1. Klicken Sie in den Eigenschaften der CPU auf "Runtime-Lizenzen > OPC UA".
2. Wählen Sie in der Klappliste bei "Typ der erworbenen Lizenz" die notwendige Lizenz aus.

9.3.3 Methoden auf dem OPC UA-Server bereitstellen

9.3.3.1 Server-Methoden

Anwenderprogramm für Server-Methoden bereitstellen

Auf dem OPC UA-Server einer S7-1500 CPU (ab Firmware V2.5) haben Sie die Möglichkeit, Methoden über Ihr Anwenderprogramm bereitzustellen. Diese Methoden können von OPC UA-Clients genutzt werden, um z. B. einen Fertigungsauftrag über den Methoden-Aufruf von der S7-1500 CPU zu starten.

OPC UA-Methoden, eine Realisierung von "Remote Procedure Calls", bieten einen effizienten Mechanismus für die Interaktionen zwischen verschiedenen Kommunikationsteilnehmern. Der Mechanismus liefert sowohl eine Auftragsbestätigung als auch Rückgabewerte, sodass Sie auf die Ausprogrammierung von Handshaking-Mechanismen verzichten können.

Mit OPC UA-Methoden können Sie z. B. Daten konsistent ohne Triggerbits/Handshaking übertragen oder bestimmte Aktionen auf der Steuerung auslösen.

Wie funktioniert eine OPC UA-Methode?

Eine OPC UA-Methode funktioniert im Prinzip wie ein Know-how geschützter Funktionsbaustein, der von einem externen OPC UA-Client zur Laufzeit aufgerufen wird.

Der OPC UA-Client "sieht" lediglich die definierten Ein- und Ausgänge. Das Innere des Funktionsbausteins, die Methode oder der Algorithmus, bleibt dem externen OPC UA-Client verborgen. Der OPC UA-Client bekommt eine Rückmeldung über die erfolgreiche Ausführung und Rückgabewerte, die der Funktionsbaustein (Methode) liefert. Oder eine Fehlermeldung bei nicht erfolgreicher Ausführung.

Sie haben als Programmierer die vollständige Kontrolle und Verantwortung darüber, in welchem Programmkontext die OPC UA-Methode abläuft.

Regeln für die Programmierung einer Methode und Laufzeitverhalten

- Sorgen Sie dafür, dass die über die OPC UA-Methode gelieferten Rückgabewerte konsistent zu dem vom OPC UA-Client gelieferten Eingabewerte sind.
- Berücksichtigen Sie die Regeln zur Namensvergabe und Aufbau von Parametern sowie die verwendbaren Datentypen (siehe Beschreibung der OPC UA-Server-Anweisungen).
- Verhalten zur Laufzeit: Der OPC UA-Server nimmt **einen** Aufruf pro Instanz an. Erst wenn dieser Aufruf vom Anwenderprogramm abgearbeitet wurde oder Timeout hatte, ist diese Methoden-Instanz wieder für andere OPC UA-Clients aufrufbar.

Im Folgenden wird die prinzipielle Vorgehensweise gezeigt, mit der Sie ein Anwenderprogramm als Server-Methode implementieren.

Implementierung einer Server-Methode

Ein Programm (Funktionsbaustein) zur Implementierung einer Server-Methode hat folgenden Aufbau:

1. Aufruf der Server-Methode abfragen mit OPC-UA_ServerMethodPre

In Ihrem Anwenderprogramm (d. h. in Ihrer Server-Methode) rufen Sie zunächst die Anweisung "OPC-UA_ServerMethodPre" auf.

Diese Anweisung hat folgende Aufgaben:

- Mit dieser Anweisung fragen Sie beim OPC UA-Server der CPU nach, ob Ihre Server-Methode von einem OPC UA-Client aufgerufen wurde.
- Wenn die Methode aufgerufen wurde und die Server-Methode über Eingangsparameter verfügt, dann erhält Ihre Server-Methode nun die Eingangsparameter.

Die Eingangsparameter der Server-Methode stammen vom aufrufenden OPC UA-Client.

2. Server-Methode bearbeiten

In diesem Abschnitt der Server-Methode stellen Sie das eigentliche Anwenderprogramm zur Verfügung.

Sie haben die gleichen Möglichkeiten wie in anderen Anwenderprogrammen (zum Beispiel Zugriff auf andere Funktionsbausteine oder auf globale Datenbausteine).

Verwendet die Server-Methode Eingangsparameter, stehen Ihnen die Eingangsparameter der Server-Methode zur Verfügung.

Dieser Abschnitt der Server-Methode sollte nur dann ausgeführt werden, wenn ein OPC UA-Client die Server-Methode aufgerufen hat.

Nach der erfolgreichen Ausführung der Methode setzen Sie die Ausgangsparameter der Server-Methode, falls die Server-Methode Ausgangsparameter besitzt.

3. Server-Methode beantworten mit OPC-UA_ServerMethodPost

Um die Server-Methode abzuschließen, rufen Sie die Anweisung "OPC-UA_ServerMethodPost" auf.

Geben Sie der Anweisung "OPC-UA_ServerMethodPost" über die Parameter die Information mit, ob das Anwenderprogramm abgearbeitet ist oder nicht.

Wenn das Anwenderprogramm erfolgreich ausgeführt wurde, wird der OPC UA-Server über die einsprechenden Parameter informiert. Der OPC UA-Server sendet nun die Ausgangsparameter der Server-Methode an den OPC UA-Client.

Rufen Sie die Anweisungen "OPC-UA_ServerMethodPre" und "OPC-UA_ServerMethodPost" immer päirchenweise auf, unabhängig davon, ob das Anwenderprogramm zwischen den beiden Anweisungen abgearbeitet oder im nächsten Zyklus fortgesetzt wird.

Ein Beispiel für die Implementierung einer Server-Methode finden Sie im Kapitel zu den OPC UA-Server-Anweisungen .

Einbindung der Server-Methode

Die folgende Grafik zeigt, wie ein OPC UA-Client (A) die Server-Methode "Cool" aufruft:

Die CPU führt im zyklischen Anwenderprogramm die Instanz "Cool1" der Server-Methode "Cool" aus ⑥.

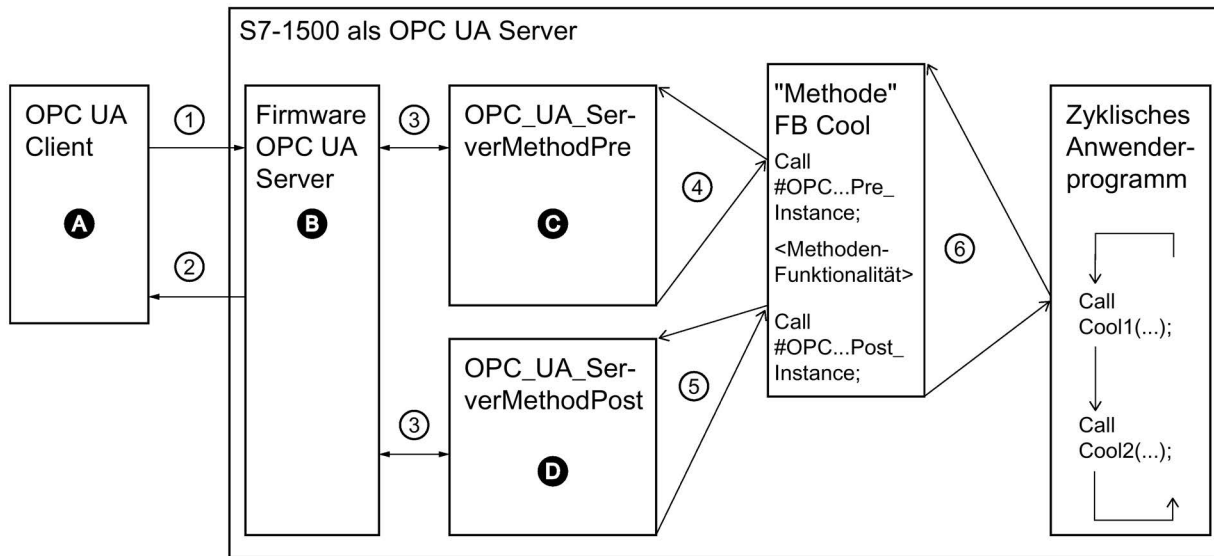
Die CPU fragt zunächst mit der Anweisung "OPC-UA_ServerMethodPre" nach ④, ob ein OPC UA-Client die Server-Methode "Cool" aufgerufen hat ①.

- Wenn die Server-Methode nicht aufgerufen wurde, kehrt die Programmausführung über ④ und ⑥ direkt zum zyklischen Anwenderprogramm zurück. Die CPU setzt das zyklische Anwenderprogramm nach "Cool1" fort.
- Wenn die Server-Methode aufgerufen wurde, gelangt diese Information über ④ zurück zur Server-Methode Cool. Dort wird nun die eigentliche Funktionalität ausgeführt, siehe "<Methoden-Funktionalität>" in der Grafik.

Anschließend teilt die Server-Methode über die Anweisung "OPC-UA_ServerMethodPost" ⑤ der Firmware (B) mit, dass sie ausgeführt wurde ③.

Die Firmware sendet diese Information über ② an den aufrufenden OPC UA-Client (A) zurück.

Die CPU setzt das zyklische Anwenderprogramm nach "Cool1" fort.



- A** Aufruf der Server-Methode und Management der "Done"-Information (Methode beendet)
- ① Asynchroner Aufruf der Server-Methode
- ② Asynchrone "Done"-Information der aufgerufenen Methode (Methode beendet)
- B** Warten auf OPC UA-Client-Aufrufe, Management von Aufrufen in der Warteschlange, "Done"-Information aus dem zyklischen Anwenderprogramm an den OPC UA-Client weiterleiten
- ③ Datentransfer vom OPC UA-Server zur Methoden-Instanz des Anwenderprogramms und umgekehrt
- C** Prüfen, ob Methode aufgerufen wurde.
Wenn ja, dann Eingangsdaten vom OPC UA-Server zur Methoden-Instanz des Anwenderprogramms weiterreichen und der Methoden-Instanz zurückmelden, dass die Methode aufgerufen wurde ("called")
- ④ Synchroner Aufruf der Anweisung OPC-UA-ServerMethodPre als Multiinstanz mit Angabe des Speicherbereichs für die Eingangsdaten vom OPC UA-Server.
Der Return-Value informiert, ob die Methode vom OPC UA-Client aufgerufen worden ist.
- ⑤ Prüfen, ob die Methode beendet wurde oder noch aktiv ist ("busy").
- D** Prüfen, ob die Methode beendet wurde.
Wenn ja, werden die Ausgangsdaten der Methoden-Instanz zum OPC UA Server weitergeleitet und der Methoden-Instanz zurückgemeldet, dass die Methode beendet ist. Der OPC UA-Server wird darüber informiert.
- ⑥ Aufruf des Methoden-FB (hier: FB Cool) mit der gewünschten Instanz und den Prozess-Parametern

Bild 9-13 Einbindung der Server-Methode

Hinweis

Methoden werden beim OPC UA-XML-Export nicht berücksichtigt

Wenn Sie das OPC UA-Informationsmodell exportieren, dann sind die nach dem oben beschriebenen Muster erstellten Methoden nicht in der exportierten XML-Datei enthalten.

Siehe auch

Beispielprogramm zum Bereitstellen einer Methode für OPC UA-Clients (Seite 205)

9.3.4 OPC UA-Server-Anweisungen für die Implementierung von Methoden

9.3.4.1 OPC_UA_ServerMethodPre

Einleitung

Dieses Kapitel beschreibt die Anweisung "OPC_UA_ServerMethodPre".

Weil die Anweisungen "OPC_UA_ServerMethodPre" und "OPC_UA_ServerMethodPost" immer paarweise im Anwenderprogramm aufzurufen sind, beachten Sie auch das Kapitel zur Anweisung "OPC_UA_ServerMethodPost".

Funktion der Anweisung

Die Anweisung "OPC_UA_ServerMethodPre" fragt beim Betriebssystem nach, ob die Server-Methode aufgerufen wurde.

Wenn die Server-Methode vom Client aufgerufen wurde, dann stellt die Anweisung "OPC_UA_ServerMethodPre" die Eingangsparameter für die Server-Methode bereit.

Deklaration der Variablen

Deklariieren Sie eine Instanz der Anweisung "OPC_UA_ServerMethodPre" sowie die Variablen, mit denen Sie die Parameter der Anweisung versorgen, siehe auch Beispielprogramm zum Bereitstellen einer Methode für OPC UA-Clients (Seite 205).

Für die Deklaration sind die folgenden Punkte wichtig:

- Legen Sie die Anweisung "OPC-UA_ServerMethodPre" als Multiinstanz im aufrufenden Funktionsbaustein an.

Hinweis

Name der Multiinstanz

Die Multiinstanz muss zwingend den Namen "OPC-UA_ServerMethodPre_Instance" haben, sonst wird keine Methode auf dem Server angelegt.

Dazu ziehen Sie per Drag & Drop die Anweisung aus dem Ordner "Anweisungen > Kommunikation > OPC UA > OPC UA Server" in den Editor.

Klicken Sie dann auf "Multiinstanz".

- Wenn die Server-Methode einen oder mehrere Eingangsparameter besitzt, dann müssen Sie eine Variable mit dem Namen "**UAMethod_InParameters**" deklarieren.

Legen Sie zunächst einen anwenderdefinierten Datentyp (UDT) für die Eingangsparameter der Server-Methode an.

Verwenden Sie dann diesen UDT für die Variable "UAMethod_InParameters".

Im Beispiel wird der Datentyp "UDT_OpenDoorInArguments" genannt und enthält das Element Number.

Alternative:

Sie können der Variablen "**UAMethod_InParameters**" auch den Datentyp "Struct" zuweisen. Legen Sie dann die Komponenten dieses Datentyps entsprechend der Eingangsparameter der Server-Methode an (gleiche Namen und Datentypen).

▼ OPC-UA_ServerMethodPre_Instance	OPC-UA_ServerMethodPre
■ Input	
■ ▼ Output	
■ Done	Bool
■ Busy	Bool
■ Error	Bool
■ Status	DWord
■ UAMethod_Called	Bool
■ ▼ InOut	
■ UAMethod_InParameters	Variant
■ Static	
▼ UAMethod_InParameters	*UDT_OpenDoorInArguments*
■ Number	Int

Parameter für "OPC-UA-ServerMethodPre"

Tabelle 9- 2 Die Parameter der Anweisung "OPC-UA-ServerMethodPre"

Parameter	Deklaration	Datentyp	Bedeutung
Done	Output	BOOL	Stand der Bearbeitung: <ul style="list-style-type: none"> 0: Ausführung der Anweisung abgebrochen, noch nicht abgeschlossen oder noch nicht begonnen 1: Ausführung der Anweisung ohne Fehler abgeschlossen
Busy	Output	BOOL	Parameter zum Stand der Bearbeitung: <ul style="list-style-type: none"> 0: Anweisung nicht in Bearbeitung 1: Anweisung momentan in Bearbeitung
Error	Output	BOOL	Fehleranzeige <ul style="list-style-type: none"> 0: Kein Fehler 1: Ein Fehler ist aufgetreten. Siehe Parameter "Status".
Status	Output	DWORD	Ursache für den Fehler, siehe unten "Fehlercodes für den Status".
UAMethodCalled	Output	BOOL	Die bereitgestellte Methode ist von einem OPC UA-Client aufgerufen worden.
UAMethod_InParameters	InOut	VARIANT	Zeiger auf eine Variable, die die Eingangsparameter für die bereitgestellte Methode enthält.

Fehlercodes für den Status

Der Parameter "Status" informiert Sie über Fehler, die bei der Ausführung der Anweisung auftreten können.

Die folgende Tabelle beschreibt die unterschiedlichen Kategorien von Fehlercodes.

Tabelle 9- 3 Fehlercodes für den Status

Fehlercode (hexadezimale Werte)	Erläuterung
0000_0000	Anweisung erfolgreich beendet.
8xxx_xxxx	OPC UA spezifischer Fehler
Axxx_xxxx	PLCopen spezifischer Fehler
B080_C300	Unzureichende Ressourcen
B08x_yz00	SIMATIC-spezifischer Fehler
Weitere Fehlercodes siehe Fehlercodes (Seite 209)	

Zuordnung von Datentypen (SIMATIC - OPC UA)

Für die Eingangs- und Ausgangsparameter von Methoden beachten Sie die Erläuterungen zu den Regeln verwendbarer Datentypen im Abschnitt "Mapping von Datentypen" (OPC UA Companion Spezifikationen verwenden (Seite 218)).

Versorgung von strukturierten Datentypen mit geschachtelten Arrays

Wenn ein strukturierter Datentyp (Struct/UDT) ein Array enthält, stellt der OPC UA-Server keine Information über die Länge dieses Arrays bereit.

Falls Sie eine solche Struktur z. B. als Eingangs- oder Ausgangsparameter einer Server-Methode verwenden, dann müssen Sie sicherstellen, dass das geschachtelte Array beim Aufruf der Methode mit der richtigen Länge versorgt wird.

Wenn Sie diese Regel nicht berücksichtigen, schlägt die Methode fehl mit dem Fehlercode "BadInvalidArgument".

9.3.4.2 OPC-UA_ServerMethodPost

Einleitung

Dieses Kapitel beschreibt die Anweisung "OPC-UA_ServerMethodPost".

Weil die Anweisungen "OPC-UA_ServerMethodPre" und "OPC-UA_ServerMethodPost" immer paarweise im Anwenderprogramm aufzurufen sind, beachten Sie auch das Kapitel zur Anweisung "OPC-UA_ServerMethodPre".

Funktion der Anweisung

Mit der Anweisung "OPC-UA_ServerMethodPost" informieren Sie das Betriebssystem darüber, dass die Server-Methode ausgeführt wurde und die Werte der Ausgangsparameter gültig sind.

Deklaration der Variablen

Deklarieren Sie eine Instanz der Anweisung "OPC-UA_ServerMethodPost" sowie die Variablen, mit denen Sie die Parameter der Anweisung versorgen, siehe auch Beispielprogramm zum Bereitstellen einer Methode für OPC UA-Clients (Seite 205).

Für die Deklaration sind die folgenden Punkte wichtig:

- Legen Sie die Anweisung "OPC-UA_ServerMethodPost" als Multiinstanz im aufrufenden Funktionsbaustein an.

Hinweis

Name der Multiinstanz

Die Multiinstanz muss zwingend den Namen "OPC-UA_ServerMethodPost_Instance" haben, sonst wird keine Methode auf dem Server angelegt.

Dazu ziehen Sie per Drag & Drop die Anweisung aus dem Ordner "Anweisungen > Kommunikation > OPC UA > OPC UA Server" in den Editor. Klicken Sie dann auf "Multiinstanz".

- Wenn die Server-Methode einen oder mehrere Ausgangsparameter besitzt, dann müssen Sie eine Variable mit dem Namen "**UAMethod_OutParameters**" deklarieren.

Legen Sie zunächst einen anwenderdefinierten Datentyp (UDT) für die Ausgangsparameter der Server-Methode an.

Verwenden Sie dann diesen UDT für die Variable "UAMethod_OutParameters".

Im Beispiel wird der Datentyp "UDT_OpenDoorOutArguments" genannt; einziger Ausgangsparameter ist Result.

Alternative:

Sie können der Variablen "**UAMethod_OutParameters**" auch den Datentyp "Struct" zuweisen. Legen Sie dann die Komponenten dieses Datentyps entsprechend der Ausgangsparameter der Server-Methode an (gleiche Namen und Datentypen).

▼ OPC-UA_ServerMethodPost_Instance	OPC-UA_ServerMethodPost
■ ▼ Input	
■ UAMethod_Result	DWord
■ UAMethod_Finished	Bool
■ ▼ Output	
■ Done	Bool
■ Busy	Bool
■ Error	Bool
■ Status	DWord
■ ▼ InOut	
■ UAMethod_OutParameters	Variant
■ Static	
▼ UAMethod_OutParameters	"UDT_OpenDoorOutArguments"
■ Result	Int

Bild 9-14 Deklaration der Variablen

Parameter für "OPC-UA-ServerMethodPost"

Tabelle 9- 4 Die Parameter der Anweisung "OPC-UA-ServerMethodPost"

Parameter	Deklaration	Datentyp	Bedeutung
Done	Output	BOOL	Stand der Bearbeitung: <ul style="list-style-type: none"> 0: Ausführung der Anweisung abgebrochen, noch nicht abgeschlossen oder noch nicht begonnen 1: Ausführung der Anweisung ohne Fehler abgeschlossen
Busy	Output	BOOL	Parameter zum Stand der Bearbeitung: <ul style="list-style-type: none"> 0: Anweisung nicht in Bearbeitung 1: Anweisung momentan in Bearbeitung
Error	Output	BOOL	Fehleranzeige <ul style="list-style-type: none"> 0: Kein Fehler 1: Ein Fehler ist aufgetreten. Siehe Parameter "Status".
Status	Output	DWORD	Ursache für den Fehler, siehe unten "Fehlercodes für den Status"
UAMethod_Result	Input	DWORD	Fehlercodes für den OPC UA-Server, bereitgestellt vom Anwenderprogramm. Empfehlung: Verwenden Sie für die Rückmeldung von Fehlern Codes, die mit 0xFF beginnen. Für OPC UA sind folgende Bereiche definiert: <ul style="list-style-type: none"> Good: 0x0000_0000 bis 0x3FFF_FFFF Uncertain: 0x4000_0000 bis 0x7FFF_FFFF Bad: 0x8000_0000 bis 0xFFFF_FFFF Je nach Client kann es sein, dass die Codes der Bereiche "Good" und "Uncertain" nicht ausgegeben werden.
UAMethod_Finished	Input	BOOL	Setzen Sie den Wert des Parameters auf TRUE, wenn die bereitgestellte Methode ausgeführt wurde.
UAMethod_OutParameters	InOut	VARIANT	Zeiger auf eine Variable, die die Ausgangsparameter der bereitgestellten Methode enthält.

Fehlercodes für den Status

Der Parameter "Status" informiert Sie über Fehler, die bei der Ausführung der Anweisung auftreten können.

Die folgende Tabelle beschreibt die unterschiedlichen Kategorien von Fehlercodes.

Tabelle 9- 5 Fehlercodes für den Status

Fehlercode (hexadezimale Werte)	Erläuterung
0000_0000	Anweisung erfolgreich beendet.
8xxx_xxxx	OPC UA spezifischer Fehler
Axxx_xxxx	PLCopen spezifischer Fehler
B080_C300	Unzureichende Ressourcen
B08x_yz00	SIMATIC-spezifischer Fehler
Weitere Fehlercodes siehe Fehlercodes (Seite 209)	

Zuordnung von Datentypen (SIMATIC - OPC UA)

Für die Eingangs- und Ausgangsparameter von Methoden beachten Sie die Erläuterungen zu den Regeln verwendbarer Datentypen im Abschnitt "Mapping von Datentypen" (OPC UA Companion Spezifikationen verwenden (Seite 218)).

Versorgung von strukturierten Datentypen mit geschachtelten Arrays

Wenn ein strukturierter Datentyp (Struct/UDT) ein Array enthält, stellt der OPC UA-Server keine Information über die Länge dieses Arrays bereit.

Falls Sie eine solche Struktur z. B. als Eingangs- oder Ausgangsparameter einer Server-Methode verwenden, dann müssen Sie sicherstellen, dass das geschachtelte Array beim Aufruf der Methode mit der richtigen Länge versorgt wird.

Wenn Sie diese Regel nicht berücksichtigen, schlägt die Methode fehl mit dem Fehlercode "BadInvalidArgument".

9.3.4.3 Beispielprogramm zum Bereitstellen einer Methode für OPC UA-Clients

Beispielprogramm für eine Server-Methode

Dieses Kapitel enthält den vollständigen Programmcode für das Beispielprogramm "OpenDoor".

Das Beispiel zeigt, wie ein Anwenderprogramm die Anweisungen "OPC-UA_ServerMethodPre" und "OPC-UA_ServerMethodPost" verwendet.

Die Anweisungen sind in den Kapiteln OPC-UA_ServerMethodPre (Seite 199) und OPC-UA_ServerMethodPost (Seite 202) beschrieben.

Das Programm stellt eine Server-Methode für OPC UA-Clients bereit: Das Programm setzt den Ausgangsparameter "Result" auf den Wert 1, wenn der Eingangsparameter "Number" den Wert 1 hat.

Um das Beispiel einfach und übersichtlich zu halten, wurde auf eine detaillierte Fehlerauswertung (Parameter "Status") verzichtet.

Programmstruktur

Das Programm gliedert sich in die folgenden Abschnitte:

1. Aufruf der Anweisung "OPC_UA_ServerMethodPre", um festzustellen, ob die Server-Methode von einem Client aufgerufen wurde.
2. Wurde die Server-Methode aufgerufen, dann wird die Server-Methode ausgeführt. Sie definiert die eigentliche Funktionalität bei einem Aufruf der Methode durch einen OPC UA-Client.
3. Wenn die Server-Methode beendet ist, dann wird die Anweisung "OPC_UA_ServerMethodPost" aufgerufen. Dieser Abschnitt informiert das Betriebssystem darüber, dass die Server-Methode ausgeführt wurde.

Deklaration

Das folgende Bild zeigt die Deklaration der lokalen Variablen für das Beispielprogramm:

▼ Static	
▶ UAMethod_OutParameters	"UDT_OpenDoorOutArguments"
▶ OPC_UA_ServerMethodPost_Instance	OPC_UA_ServerMethodPost
▶ UAMethod_InParameters	"UDT_OpenDoorInArguments"
▶ OPC_UA_ServerMethodPre_Instance	OPC_UA_ServerMethodPre
Method_Result	DWord
Method_Finished	Bool
Started	Bool
Error_Message	WString
▼ Temp	
Method_Called	Bool
Pre_Done	Bool
Pre_Error	Bool
Post_Done	Bool
Post_Error	Bool

Bild 9-15 Deklaration der lokalen Variablen

SCL-Programm

Das folgende Programm zeigt, wie Sie OPC UA-Anweisungen nutzen, um eine Methode für OPC UA-Clients bereitzustellen, die in Ihrem Anwenderprogramm ausgeführt wird (Server-Methode).

Aufruf der Anweisung "OPC_UA_ServerMethodPre"

Zuerst wird die Anweisung "OPC_UA_ServerMethodPre" aufgerufen, um beim Betriebssystem zu erfragen, ob die Server-Methode von einem OPC UA-Client aufgerufen wurde (Zeilen 2 bis 5).

Wurde die Server-Methode aufgerufen, dann ist die Variable "#Method_Called" gleich "TRUE" (in Zeile 3).

Wurde zudem die Anweisung "OPC_UA_ServerMethodPre" erfolgreich ausgeführt (#Pre_Done = TRUE), dann setzt die Zeile 10 die Variable "#Started" auf "TRUE".

Server-Methode

Wenn die Variable "#Started" den Wert "TRUE" hat, wird nun das eigentliche Anwendungsprogramm (Zeilen 15 bis 31) ausgeführt.

In diesem Abschnitt haben Sie alle programmtechnischen Möglichkeiten: Sie können Prozesswerte einlesen oder Prozesswerte ausgeben, auf globale Datenbausteine zugreifen, Funktionen und Funktionsbausteine aufrufen, usw.

Das Anwenderprogramm kann mehrere Zyklen dauern.

Um das Ende des Anwenderprogramms zu signalisieren, setzen Sie die Variable "#Method_Finished" auf "TRUE".

Mit "#Method_Result" übermitteln Sie einen selbstdefinierte Fehlercode, der am Bausteinparameter "UAMethod_Result" der Anweisung "OPC-UA_ServerMethodPost" verwendet wird.

Aufruf der Anweisung "OPC-UA_ServerMethodPost"

Die Variable "#Method_Finished" wurde in Zeile 21 bzw. 29 (Server-Methode) gesetzt, um den Zustand zu speichern, dass das Anwenderprogramm (Zeilen 15 bis 31) ausgeführt wurde.

Diese Variable wird in der Anweisung "OPC-UA_ServerMethodPost" genutzt, um beim Betriebssystem mitzuteilen, ob die Server-Methode ausgeführt wurde oder nicht (Zeilen 33 bis 45).

Das Betriebssystem übernimmt dann die Quittierung zum Client, der die Methode aufgerufen hat.

```

1  // Server Method called?
2  □ #OPC-UA_ServerMethodPre_Instance(UAMethod_InParameters := #UAMethod_InParameters,
3  |                                     UAMethod_Called => #Method_Called,
4  |                                     Error => #Pre_Error,
5  |                                     Done => #Pre_Done);
6  □ IF #Pre_Error THEN
7  |   #Error_Message := WString#'Error at OPC-UA_ServerMethodPre';
8  | END_IF;
9  □ IF #Pre_Done AND #Method_Called THEN
10 |   #Started := TRUE;
11 |   #Error_Message := WString#'';
12 | END_IF;
13
14 // The Server Method itself ...
15 □ IF #Started = TRUE THEN
16 |   #UAMethod_OutParameters.Result := 0;
17 |   IF #UAMethod_InParameters.Number = 1 THEN
18 |     #UAMethod_OutParameters.Result := 1;
19 |     "OpenDoorToProductionline" := TRUE;
20 |     #Method_Result := DWord#0000_0000;
21 |     #Method_Finished := TRUE;
22 |     #Started := FALSE;
23 |   ELSE
24 |     #Error_Message := WString#'Input-Parameter Number must be 1 to run this example server method';
25 |     //send back an error code to the calling OPC UA client
26 |     //for the example, we define 16#FFFF_FFFF = "Input-Parameter Number must be 1"
27 |     #Method_Result := 16#FFFF_FFFF;
28 |     #Started := FALSE;
29 |     #Method_Finished := TRUE;
30 |   END_IF;
31 | END_IF;
32
33 // Server Method finished?
34 □ #OPC-UA_ServerMethodPost_Instance(UAMethod_Result := #Method_Result,
35 |                                     UAMethod_Finished := #Method_Finished,
36 |                                     UAMethod_OutParameters := #UAMethod_OutParameters,
37 |                                     Done => #Post_Done,
38 |                                     Error => #Post_Error);
39 □ IF #Post_Error = TRUE THEN
40 |   #Error_Message := WString#'Error at OPC-UA_ServerMethodPost';
41 |   #Method_Finished := FALSE;
42 | END_IF;
43 □ IF #Post_Done = TRUE THEN
44 |   #Method_Finished := FALSE;
45 | END_IF;

```

Bild 9-16 Aufruf der Anweisung "OPC-UA_ServerMethodPost"

Methode löschen

Wenn Sie eine Server-Methode nicht mehr zur Verfügung stellen und löschen wollen, dann müssen Sie nicht nur den Aufruf des entsprechenden FBs entfernen, sondern auch den Instanz-Datenbaustein löschen.

Wird der Instanz-Datenbaustein nicht gelöscht, dann ist die Methode im Adressraum sichtbar, die Variablen werden nicht versorgt.

Siehe auch

Server-Methoden (Seite 194)

9.3.4.4 Fehlercodes

Fehlercodes von Siemens

SIMATIC-Fehlercodes für OPC UA-Anweisungen

In der folgenden Tabelle finden Sie die Fehlercodes von Siemens für die OPC UA-Anweisungen:

Tabelle 9- 6 SIMATIC-Fehlercodes für OPC UA-Anweisungen

Fehlercode (hex)	Beschreibung
B080_C300	InsufficientResources: Keine ausreichenden Ressourcen. Entweder (a) Fehler bei Speicherallokation oder (b) zu viele SFB-Instanzen
B080_0B00	ArrayElements_TooMany: Das Array besitzt zu viele Elemente.
B080_B000	TooManyMethods: Max. Anzahl Server-Methoden bzw. max. Anzahl Server-Methoden-Instanzen überschritten (Aufrufe der Anweisungen OPC_UA_ServerMethodPre, OPC_UA_ServerMethodPost): <ul style="list-style-type: none">• Für S7-1510 bis S7-1513 ...: max. 20• Für S7-1515, S7-1516...: max. 50• Für S7-1517, S7-1518, S7-1507S...: max. 100• (siehe auch Hinweise zu Mengengerüsten bei Nutzung von Server-Schnittstellen (Seite 233))

Fehlercodes der OPC Foundation

Fehlercodes

Die folgende Tabelle enthält Fehlercodes der OPC Foundation.

Die Namen und Erläuterungen der Fehler sind im Original aufgeführt. Sie sind nur in Englisch verfügbar.

Fehlercodes

Tabelle 9- 7 Fehlercodes der OPC Foundation

Status "Good" (0000_0000 - 3FFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
00000000	Good	Success, no error.
002D0000	GoodSubscriptionTransferred	The subscription was transferred to another session.
002E0000	GoodCompletesAsynchronously	The processing will complete asynchronously.
002F0000	GoodOverload	Sampling has slowed down due to resource limitations.
00300000	GoodClamped	The value written was accepted but was clamped.
00960000	GoodLocalOverride	The value has been overridden.
00A20000	GoodEntryInserted	The data or event was successfully inserted into the historical database.
00A30000	GoodEntryReplaced	The data or event field was successfully replaced in the historical database.
00A50000	GoodNoData	No data exists for the requested time range or event filter.
00A60000	GoodMoreData	The data or event field was successfully replaced in the historical database.
00A70000	GoodCommunicationEvent	The communication layer has raised an event.
00A80000	GoodShutdownEvent	The system is shutting down.
00A90000	GoodCallAgain	The operation is not finished and needs to be called again.
00AA0000	GoodNonCriticalTimeout	A non-critical timeout occurred.
00BA0000	GoodResultsMayBeIncomplete	The server should have followed a reference to a node in a remote server but did not. The result set may be incomplete.
00D90000	GoodDataIgnored	The request specifies fields which are not valid for the EventType or cannot be saved by the historian.
00DC0000	GoodEdited	The value does not come from the real source and has been edited by the server.
00DD0000	GoodPostActionFailed	There was an error in execution of these post-actions.
00E00000	GoodDependentValueChanged	A dependent value has been changed but the change has not been applied to the device.

Tabelle 9- 8 Fehlercodes der OPC Foundation

Status "Uncertain" (4000_0000 - 7FFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
406C0000	UncertainReferenceOutOfServer	One of the references to follow in the relative path references to a node in the address space in another server.
408F0000	UncertainNoCommunicationLastUsableValue	Communication to the data source has failed. The variable value is the last value that had a good quality.
40900000	UncertainLastUsableValue	Whatever was updating this value has stopped doing so.
40910000	UncertainSubstituteValue	The value is an operational value that was manually overwritten.
40920000	UncertainInitialValue	The value is an initial value for a variable that normally receives its value from another variable.
40930000	UncertainSensorNotAccurate	The value is at one of the sensor limits.
40940000	UncertainEngineeringUnitsExceeded	The value is outside of the range of values defined for this parameter.
40950000	UncertainSubNormal	The value is derived from multiple sources and has less than the required number of Good sources.
40A40000	UncertainDataSubNormal	The value is derived from multiple values and has less than the required number of Good values.
40BC0000	UncertainReferenceNotDeleted	The server was not able to delete all target references.
40C00000	UncertainNotAllNodesAvailable	The list of references may not be complete because the underlying system is not available.
40DE0000	UncertainDominantValueChanged	The related EngineeringUnit has been changed but the Variable Value is still provided based on the previous unit.
40E20000	UncertainDependentValueChanged	A dependent value has been changed but the change has not been applied to the device. The quality of the dominant variable is uncertain.

Tabelle 9- 9 Fehlercodes der OPC Foundation

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
80010000	BadUnexpectedError	An unexpected error occurred
80020000	BadInternalError	An internal error occurred as a result of a programming or configuration error.
80030000	BadOutOfMemory	Not enough memory to complete the operation.
80040000	BadResourceUnavailable	An operating system resource is not available
80050000	BadCommunicationError	A low level communication error occurred.
80060000	BadEncodingError	Encoding halted because of invalid data in the objects being serialized.
80070000	BadDecodingError	Decoding halted because of invalid data in the stream.
80080000	BadEncodingLimitsExceeded	The message encoding/decoding limits imposed by the stack have been exceeded.
80090000	BadUnknownResponse	An unrecognized response was received from the server.
80B80000	BadRequestTooLarge	The request message size exceeds limits set by the server.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
80B90000	BadResponseTooLarge	The response message size exceeds limits set by the client.
800A0000	BadTimeout	The operation timed out.
800B0000	BadServiceUnsupported	The server does not support the requested service.
800C0000	BadShutdown	The operation was cancelled because the application is shutting down.
800D0000	BadServerNotConnected	The operation could not complete because the client is not connected to the server.
800E0000	BadServerHalted	The server has stopped and cannot process any requests.
800F0000	BadNothingToDo	There was nothing to do because the client passed a list of operations with no elements.
80100000	BadTooManyOperations	The request could not be processed because it specified too many operations.
80110000	BadDataTypeIdUnknown	The extension object cannot be (de)serialized because the data type id is not recognized.
80120000	BadCertificateInvalid	The certificate provided as a parameter is not valid.
80130000	BadSecurityChecksFailed	An error occurred verifying security. The certificate provided as a parameter is not valid.
80140000	BadCertificateTimeInvalid	The Certificate has expired or is not yet valid.
80150000	BadCertificateIssuerTimeInvalid	An Issuer Certificate has expired or is not yet valid.
80160000	BadCertificateHostNameInvalid	The HostName used to connect to a Server does not match a HostName in the Certificate.
80170000	BadCertificateUriInvalid	The URI specified in the Application Description does not match the URI in the Certificate.
80180000	BadCertificateUseNotAllowed	The Certificate may not be used for the requested operation.
80190000	BadCertificateIssuerUseNotAllowed	The Issuer Certificate may not be used for the requested operation.
801A0000	BadCertificateUntrusted	The Certificate is not trusted.
801B0000	BadCertificateRevocationUnknown	It was not possible to determine if the Certificate has been revoked.
801C0000	BadCertificateIssuerRevocationUnknown	It was not possible to determine if the Issuer Certificate has been revoked.
801D0000	BadCertificateRevoked	The Certificate has been revoked.
801E0000	BadCertificateIssuerRevoked	The Issuer Certificate has been revoked.
801F0000	BadUserAccessDenied	User does not have permission to perform the requested operation.
80200000	BadIdentityTokenInvalid	The user identity token is not valid.
80210000	BadIdentityTokenRejected	The user identity token is valid but the server has rejected it.
80220000	BadSecureChannelIdInvalid	The specified secure channel is no longer valid.
80230000	BadInvalidTimestamp	The timestamp is outside the range allowed by the server.
80240000	BadNonceInvalid	The nonce does appear to be not a random value or it is not the correct length.
80250000	BadSessionIdInvalid	The session id is not valid.
80260000	BadSessionClosed	The session was closed by the client.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
80270000	BadSessionNotActivated	The session cannot be used because ActivateSession has not been called.
80280000	BadSubscriptionIdInvalid	The subscription id is not valid.
802A0000	BadRequestHeaderInvalid	The header for the request is missing or invalid.
802B0000	BadTimestampsToReturnInvalid	The timestamps to return parameter is invalid.
802C0000	BadRequestCancelledByClient	The request was cancelled by the client.
80310000	BadNoCommunication	Communication with the data source is defined, but not established, and there is no last known value available.
80320000	BadWaitingForInitialData	Waiting for the server to obtain values from the underlying data source.
80330000	BadNodeIdInvalid	The syntax of the node id is not valid.
80340000	BadNodeIdUnknown	The node id refers to a node that does not exist in the server address space.
80350000	BadAttributeIdInvalid	The attribute is not supported for the specified Node.
80360000	BadIndexRangeInvalid	The syntax of the index range parameter is invalid.
80370000	BadIndexRangeNoData	No data exists within the range of indexes specified.
80380000	BadDataEncodingInvalid	The data encoding is invalid.
80390000	BadDataEncodingUnsupported	The server does not support the requested data encoding for the node.
803A0000	BadNotReadable	The access level does not allow reading or subscribing to the Node.
803B0000	BadNotWritable	The access level does not allow writing to the Node.
803C0000	BadOutOfRange	The value was out of range.
803D0000	BadNotSupported	The requested operation is not supported.
803E0000	BadNotFound	A requested item was not found or a search operation ended without success.
803F0000	BadObjectDeleted	The object cannot be used because it has been deleted.
80400000	BadNotImplemented	Requested operation is not implemented.
80410000	BadMonitoringModeInvalid	The monitoring mode is invalid.
80420000	BadMonitoredItemIdInvalid	The monitoring item id does not refer to a valid monitored item.
80430000	BadMonitoredItemFilterInvalid	The monitored item filter parameter is not valid.
80440000	BadMonitoredItemFilterUnsupported	The server does not support the requested monitored item filter.
80450000	BadFilterNotAllowed	A monitoring filter cannot be used in combination with the attribute specified.
80460000	BadStructureMissing	A mandatory structured parameter was missing or null.
80470000	BadEventFilterInvalid	The event filter is not valid.
80480000	BadContentFilterInvalid	The content filter is not valid.
80490000	BadFilterOperandInvalid	The operand used in a content filter is not valid.
804A0000	BadContinuationPointInvalid	The continuation point provide is longer valid.
804B0000	BadNoContinuationPoints	The operation could not be processed because all continuation points have been allocated.
804C0000	BadReferenceTypeIdInvalid	The operation could not be processed because all continuation points have been allocated.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
804D0000	BadBrowseDirectionInvalid	The browse direction is not valid.
804E0000	BadNodeNotInView	The node is not part of the view.
804F0000	BadServerUriInvalid	The ServerUri is not a valid URI.
80500000	BadServerNameMissing	No ServerName was specified
80510000	BadDiscoveryUrlMissing	No DiscoveryUrl was specified.
80520000	BadSemaphoreFileMissing	The semaphore file specified by the client is not valid.
80530000	BadRequestTypeInvalid	The security token request type is not valid.
80540000	BadSecurityModeRejected	The security mode does not meet the requirements set by the Server.
80550000	BadSecurityPolicyRejected	The security policy does not meet the requirements set by the Server.
80560000	BadTooManySessions	The server has reached its maximum number of sessions.
80570000	BadUserSignatureInvalid	The user token signature is missing or invalid.
80580000	BadApplicationSignature Invalid	The signature generated with the client certificate is missing or invalid.
80590000	BadNoValidCertificates	The client did not provide at least one software certificate that is valid and meets the profile requirements for the server.
805A0000	BadRequestCancelled ByRequest	The request was cancelled by the client with the Cancel service.
805B0000	BadParentNodeIdInvalid	The parent node id does not to refer to a valid node.
805C0000	BadReferenceNotAllowed	The reference could not be created because it violates constraints imposed by the data model.
805D0000	BadNodeIdRejected	The requested node id was reject because it was either invalid or server does not allow node ids to be specified by the client.
805E0000	BadNodeIdExists	The requested node id is already used by another node.
805F0000	BadNodeClassInvalid	The node class is not valid.
80600000	BadBrowseNameInvalid	The browse name is invalid.
80610000	BadBrowseNameDuplicated	The browse name is not unique among nodes that share the same relationship with the parent.
80620000	BadNodeAttributesInvalid	The node attributes are not valid for the node class.
80630000	BadTypeDefinitionInvalid	The type definition node id does not reference an appropriate type node.
80640000	BadSourceNodeIdInvalid	The source node id does not reference a valid node.
80650000	BadTargetNodeIdInvalid	The target node id does not reference a valid node.
80660000	BadDuplicateReference NotAllowed	The reference type between the nodes is already defined.
80670000	BadInvalidSelfReference	The server does not allow this type of selfreference on this node.
80680000	BadReferenceLocalOnly	The reference type is not valid for a reference to a remote server.
80690000	BadNoDeleteRights	The server will not allow the node to be deleted.
806A0000	BadServerIndexInvalid	The server index is not valid.
806B0000	BadViewIdUnknown	The view id does not refer to a valid view node.
806D0000	BadTooManyMatches	The requested operation has too many matches to return.
806E0000	BadQueryTooComplex	The requested operation requires too many resources in the server.
806F0000	BadNoMatch	The requested operation has no match to return.
80700000	BadMaxAgeInvalid	The max age parameter is invalid.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
80710000	BadHistoryOperationInvalid	The history details parameter is not valid.
80720000	BadHistoryOperation Unsupported	The server does not support the requested operation.
80730000	BadWriteNotSupported	The server not does support writing the combination of value, status and timestamps provided.
80740000	BadTypeMismatch	The value supplied for the attribute is not of the same type as the attribute's value.
80750000	BadMethodInvalid	The method id does not refer to a method for the specified object.
80760000	BadArgumentsMissing	The client did not specify all of the input arguments for the method.
80770000	BadTooManySubscriptions	The server has reached its maximum number of subscriptions.
80780000	BadTooManyPublish Requests	The server has reached the maximum number of queued publish requests.
80790000	BadNoSubscription	There is no subscription available for this session.
807A0000	BadSequenceNumber Unknown	The sequence number is unknown to the server.
807B0000	BadMessageNotAvailable	The requested notification message is no longer available.
807C0000	BadInsufficientClientProfile	The Client of the current Session does not support one or more Profiles that are necessary for the Subscription.
80BF0000	BadStateNotActive	The sub-state machine is not currently active.
807D0000	BadTcpServerTooBusy	The server cannot process the request because it is too busy.
807E0000	BadTcpMessageTypeInvalid	The type of the message specified in the header invalid.
807F0000	BadTcpSecureChannel Unknown	The SecureChannelId and/or TokenId are not currently in use.
80800000	BadTcpMessageTooLarge	The size of the message specified in the header is too large.
80810000	BadTcpNotEnough Resources	There are not enough resources to process the request.
80820000	BadTcpInternalError	An internal error occurred.
80830000	BadTcpEndpointUrlInvalid	The Server does not recognize the QueryString specified.
80840000	BadRequestInterrupted	The request could not be sent because of a network interruption.
80850000	BadRequestTimeout	Timeout occurred while processing the request.
80860000	BadSecureChannelClosed	The secure channel has been closed.
80870000	BadSecureChannelToken Unknown	The token has expired or is not recognized.
80880000	BadSequenceNumberInvalid	The sequence number is not valid.
80890000	BadConfigurationError	There is a problem with the configuration that affects the usefulness of the value.
808A0000	BadNotConnected	The variable should receive its value from another variable, but has never been configured to do so.
808B0000	BadDeviceFailure	There has been a failure in the device/data source that generates the value that has affected the value.
808C0000	BadSensorFailure	There has been a failure in the sensor from which the value is derived by the device/data source.
808D0000	BadOutOfService	The source of the data is not operational.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
808E0000	BadDeadbandFilterInvalid	The dead band filter is not valid.
80970000	BadRefreshInProgress	This Condition refresh failed, a Condition refresh operation is already in progress.
80980000	BadConditionAlreadyDisabled	This condition has already been disabled.
80990000	BadConditionDisabled	Property not available, this condition is disabled.
809A0000	BadEventIdUnknown	The specified event id is not recognized.
809B0000	BadNoData	No data exists for the requested time range or event filter.
809D0000	BadDataLost	Data is missing due to collection started/stopped/lost.
809E0000	BadDataUnavailable	Expected data is unavailable for the requested time range due to an unmounted volume an off-line archive or tape or similar reason for temporary unavailability.
809F0000	BadEntryExists	The data or event was not successfully inserted because a matching entry exists.
80A00000	BadNoEntryExists	The data or event was not successfully updated because no matching entry exists.
80A10000	BadTimestampNotSupported	The client requested history using a timestamp format the server does not support (i. e. requested ServerTimestamp when server only supports SourceTimestamp).
80AB0000	BadInvalidArgument	One or more arguments are invalid.
80AC0000	BadConnectionRejected	Could not establish a network connection to remote server.
80AD0000	BadDisconnect	The server has disconnected from the client.
80AE0000	BadConnectionClosed	The network connection has been closed.
80AF0000	BadInvalidState	The operation cannot be completed because the object is closed uninitialized or in some other invalid state.
80B00000	BadEndOfStream	Cannot move beyond end of the stream.
80B10000	BadNoDataAvailable	No data is currently available for reading from a non-blocking stream.
80B20000	BadWaitingForResponse	The asynchronous operation is waiting for a response.
80B30000	BadOperationAbandoned	The asynchronous operation was abandoned by the caller.
80B40000	BadExpectedStreamToBlock	The stream did not return all data requested (possibly because it is a non-blocking stream).
80B50000	BadWouldBlock	Non-blocking behavior is required and the operation would block.
80B60000	BadSyntaxError	A value had an invalid syntax.
80B70000	BadMaxConnections Reached	The operation could not be finished because all available connections are in use.
80BB0000	BadEventNot Acknowledgeable	The event cannot be acknowledged.
80BD0000	BadInvalidTimestamp Argument	The defined timestamp to return was invalid.
80BE0000	BadProtocolVersion Unsupported	The applications do not have compatible protocol versions.
80C10000	BadFilterOperatorInvalid	An unrecognized operator was provided in a filter.
80C20000	BadFilterOperator Unsupported	A valid operator was provided, but the server does not provide support for this filter operator.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
80C30000	BadFilterOperandCountMismatch	The number of operands provided for the filter operator was less than expected for the operand provided.
80C40000	BadFilterElementInvalid	The referenced element is not a valid element in the content filter.
80C50000	BadFilterLiteralInvalid	The referenced literal is not a valid value.
80C60000	BadIdentityChangeNotSupported	The Server does not support changing the user identity assigned to the session.
80C80000	BadNotTypeDefinition	The provided Nodeid was not a type definition nodeid.
80C90000	BadViewTimestampInvalid	The view timestamp is not available or not supported.
80CA0000	BadViewParameterMismatch	The view parameters are not consistent with each other.
80CB0000	BadViewVersionInvalid	The view version is not available or not supported.
80CC0000	BadConditionAlreadyEnabled	This condition has already been enabled.
80CD0000	BadDialogNotActive	The dialog condition is not active.
80CE0000	BadDialogResponseInvalid	The response is not valid for the dialog.
80CF0000	BadConditionBranchAlreadyAked	The condition branch has already been acknowledged.
80D00000	BadConditionBranchAlreadyConfirmed	The condition branch has already been confirmed.
80D10000	BadConditionAlreadyShelved	The condition has already been shelved.
80D20000	BadConditionNotShelved	The condition is not currently shelved.
80D30000	BadShelvingTimeOutOfRange	The shelving time not within an acceptable range.
80D40000	BadAggregateListMismatch	The requested number of Aggregates does not match the requested number of NodeIds.
80D50000	BadAggregateNotSupported	The requested Aggregate is not support by the server.
80D60000	BadAggregateInvalidInputs	The aggregate value could not be derived due to invalid data inputs.
80DB0000	BadTooManyMonitoredItems	The request could not be processed because there are too many monitored items in the subscription.
80D70000	BadBoundNotFound	No data found to provide upper or lower bound value.
80D80000	BadBoundNotSupported	The server cannot retrieve a bound for the variable.
80DA0000	BadAggregateConfigurationRejected	The aggregate configuration is not valid for specified node.
80E10000	BadDominantValueChanged	The related EngineeringUnit has been changed but this change has not been applied to the device. The Variable Value is still dependent on the previous unit but its status is currently bad.
80E30000	BadDependentValueChanged	A dependent value has been changed but the change has not been applied to the device. The quality of the dominant variable is bad.
80E40000	BadRequestNotAllowed	The request was rejected by the server because it did not meet the criteria set by the server.
80E50000	BadTooManyArguments	Too many arguments were provided.
80E60000	BadSecurityModelInsufficient	The operation is not permitted over the current secure channel.
810D0000	BadCertificateChainIncomplete	The certificate chain is incomplete.

Status "Bad" (8000_0000 - FFFF_FFFF)		
Fehlernummer (hex)	Name	Bedeutung
810E0000	BadLicenseExpired	The server requires a license to operate in general or to perform a service or operation, but existing license is expired.
810F0000	BadLicenseLimitsExceeded	The server has limits on number of allowed operations / objects, based on installed licenses, and these limits were exceeded.
81100000	BadLicenseNotAvailable	The server does not have a license which is required to operate in general or to perform a service or operation.
81110000	BadNotExecutable	The executable attribute does not allow the execution of the method.
81120000	BadNumericOverflow	The number was not accepted because of a numeric overflow.
81130000	BadRequestNotComplete	The request has not been processed by the server yet.

9.3.5 OPC UA-Informationsmodelle nutzen

9.3.5.1 OPC UA Companion Spezifikationen verwenden

Einleitung

OPC UA ist universell einsetzbar: Der Standard selbst macht z. B. keine Aussagen darüber, wie PLC-Variablen zu benennen sind. Auch steht es im Ermessen des einzelnen Nutzers (Anwendungsentwicklers), Server-Methoden zu programmieren und zu benennen, die über OPC UA aufrufbar sind.

Informationsmodellierung und Standardisierung für Geräte und Branchen

Für gleichartige Anwendungen bietet es sich an, mit dem "OPC UA-Baukasten" sein Geräteinterface oder Maschineninterface zu standardisieren.

Viele Gremien und Arbeitskreise haben die Standardisierung vorangetrieben und unterschiedliche Companion Spezifikationen erarbeitet.

In diesen Spezifikationen ist festgelegt:

- Mit welchen Objekten, Methoden und Variablen ist ein typisches Gerät oder eine typische Maschine zu beschreiben.
- Welcher Namespace ist für die genannten Objekte vorgesehen.

Maschinen werden dabei typischerweise in funktionale bzw. technologische Einheiten strukturiert und diese Einheiten standardisiert.

Den Betreibern von Maschinen und Anlagen bieten Companion Spezifikationen den Vorteil einer einheitlichen Schnittstelle. So sind zum Beispiel alle RFID-Reader, welche die Spezifikation AutoID einhalten, auf die gleiche Weise integrierbar. Das heißt, alle RFID-Reader, die konform der Spezifikation AutoID sind, lassen sich auf die gleiche Weise von OPC UA-Clients ansprechen, unabhängig vom Hersteller der RFID-Reader.

Ein anderes Beispiel für eine Companion Spezifikation aus dem Bereich Spritzgießmaschinen ist die Euromap 77 Companion Spezifikation.

Das folgende Kapitel beschreibt am Beispiel von Euromap 77, wie Sie eine Companion Spezifikation in STEP 7 (TIA Portal) übernehmen und dafür PLC-Variablen anlegen.

Beispiel Euromap 77

Euromap 77 standardisiert den Datenaustausch zwischen Spritzgießmaschinen und dem übergelagerten MES (manufacturing execution system). Dadurch kann das MES alle unterlagerten Spritzgießmaschinen gleich anbinden.

Die einheitliche Datenschnittstelle erleichtert die Integration von Spritzgießmaschinen in eine Anlage.

Companion Spezifikation verwenden: Übersicht

Die Euromap 77 ist in der OPC UA-XML-Datei "Opc_Ua.EUROMAP77.NodeSet2.xml" beschrieben.

Hinweis

Euromap 77, Euromap 83 und OPC UA for Devices (DI)

Mit dem Release Candidate 2 wurde ein Teil der Definitionen von Euromap nach Euromap 83 übertragen. Deshalb müssen Sie auch die OPC UA-Server-Schnittstelle von Euromap 83 importieren.

"OPC UA for Devices" ist ein allgemein gültiges Informationsmodell für die Konfiguration von Hardware- und Softwarekomponenten. Das Informationsmodell dient auch als Basis für weitere Companion Standards und wird daher ebenfalls importiert.

Sie erhalten die OPC UA-XML-Dateien hier:

Euromap77 (<http://www.euromap.org/euromap77>)

Euromap83 (<http://www.euromap.org/euromap83>)

OPC UA for Devices (<https://opcfoundation.org/UA/schemas/DI/>)

Die Euromap-OPC UA-XML-Dateien definieren die Schnittstelle des OPC UA-Servers einer Spritzgießmaschine, die konform zum Euromap 77 Companion Standard ist.

Um den OPC UA-Server der S7-1500 CPU nach dem Euromap 77 Standard zu modellieren, gehen Sie folgendermaßen vor:

1. Erstellen Sie eine XML-Datei, in der Sie eine Instanz des Typs "IMM_MES_InterfaceType" anlegen.

"IMM_MES_InterfaceType" ist der oberste Knoten in der Euromap 77: Dieser Datentyp ist direkt vom OPC UA-Datentyp "BaseObjectType" abgeleitet.

Unterhalb von "IMM_MES_InterfaceType" sind alle Variablen und Methoden der Euromap 77 (83) definiert.

Wie Sie im Einzelnen vorgehen, ist unter "Schritt 1" beschrieben.

2. Ordnen Sie den Variablen und Methoden der Euromap 77 (dem Informationsmodell der Euromap 77) PLC-Variablen und FB-Instanzen (Server-Methoden) Ihrer S7-1500 CPU zu.

Wie Sie im Einzelnen vorgehen, ist unter "Schritt 2" beschrieben.

3. Importieren Sie die XML-Datei als "Server-Schnittstelle".

Wie Sie im Einzelnen vorgehen, ist unter "Schritt 3" beschrieben.

Übersetzen Sie das STEP 7-Projekt.

Laden Sie das Projekt in die CPU, die als Steuerung für eine Spritzgießmaschine fungiert.

4. Legen Sie in Ihrem STEP 7-Projekt die PLC-Variablen und Server-Methoden an, denen Sie unter Punkt 2 Variablen und Methoden der Euromap 77 zugeordnet haben.

Die PLC-Variablen müssen kompatible Datentypen besitzen, siehe unten "Mapping der Datentypen".

Erlauben Sie den Lese- und Schreibzugriff von OPC UA-Clients auf diese PLC-Variablen entsprechend Euromap 77.

Legen Sie die PLC-Variablen z. B. in einem Datenbaustein an.

Wie Sie im Einzelnen vorgehen, ist unter "Schritt 4" beschrieben.

Ergebnis: Die Variablen und Server-Methoden nach Euromap 77 sind im Adressraum des OPC UA-Servers Ihrer CPU für OPC UA-Clients verfügbar.

Schritt 1: Instanz erstellen

Die folgende beispielhafte Beschreibung verwendet das Programm "UaModeler" von Unified Automation. Sie können auch andere Tools verwenden, um die OPC UA-XML-Datei (Server-Schnittstelle) zu erstellen. Die Tools zum Entwerfen von Informationsmodellen werden ständig weiterentwickelt. Nutzen Sie daher für die Handhabung immer die vom Hersteller zur Verfügung gestellte Dokumentation. Diese Dokumentation hat Vorrang vor der hier vorliegenden Beschreibung.

Sie können das Programm "UaModeler" von hier laden: Unified_Automation_Download (<https://www.unified-automation.com/downloads/opc-ua-development.html>)

Mit Hilfe von UaModeler entwerfen Sie Informationsmodelle/Adressräume für Ihren OPC UA-Server und können neue Typen und Instanzen von OPC UA-Nodes erstellen. Sie können UaModeler auch dazu nutzen, Extensions zu UA-Variablen oder UA-Methoden zu definieren, um UA-Variablen und UA-Methoden auf PLC-Variablen und PLC-Funktionsbausteine (Instanzen) zu mappen.

Um eine XML-Datei mit einer Instanz von "IMM_MES_InterfaceType" anzulegen, gehen Sie folgendermaßen vor:

1. Laden Sie die Dateien "Opc_Ua.EUROMAP77.NodeSet2.xml" und "Opc_Ua_EUROMAP83_NodeSet2.xml" von der Euromap-Webseite (siehe oben).
2. Legen Sie in UaModeler ein neues Projekt an und tragen Sie einen Projektnamen ein.
Im Beispiel verwenden Sie "Demo" als Projektnamen.
3. Unter "Generate Code" klicken Sie auf "modeling > v1_0" und dann auf die Schaltfläche "Next".
4. Unter "Choose Base Models" klicken Sie auf die Schaltfläche "Find another model", falls die Datei "Ua.EUROMAP77.NodeSet2.ua" bzw. "Opc_Ua_EUROMAP83_NodeSet2.xml" nicht bereits geladen sind.

UaModeler speichert die XML-Dateien in einem internen Format mit der Endung ".ua".

5. Im Dialog "Add Information Model" wählen Sie "Information Model (*.xml)" als Dateityp aus. Anschließend wählen Sie die XML-Datei "Opc_Ua.EUROMAP77.NodeSet2.xml" und "Opc_Ua_EUROMAP83_NodeSet2.xml" im Dateisystem Ihres Rechners aus.

6. Eine Warnung wird angezeigt, dass das ausgewählte Modell auf mehr als einem Modell basiert. Klicken Sie auf "Yes".

Dadurch wird das zu Grunde liegende Modell "Opc.Ua.Di.Nodset2.ua" zusätzlich selektiert.

7. Der Dialog "New Model" wird angezeigt.

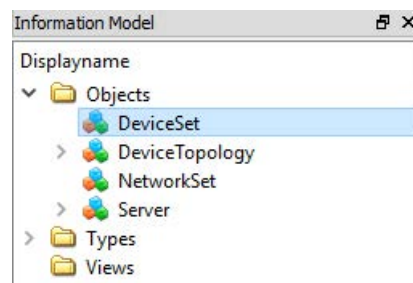
Im Beispiel ändern Sie die eingestellten Werte nicht und klicken auf "Finish", um die Werte zu übernehmen.

Unter "Namespace URI" ist "http://yourorganisation.org/Demo/" eingetragen.

Verwenden Sie in Ihrem Projekt einen aussagekräftigen Namensraum Ihres Unternehmens.

8. UaModeler öffnet das Projekt.

Unter "Information Model" klicken Sie mit der rechten Maustaste auf "DeviceSet" und dann auf "Add Instance".



9. Im Beispiel verwenden Sie Zeichenketten für Nodelds.

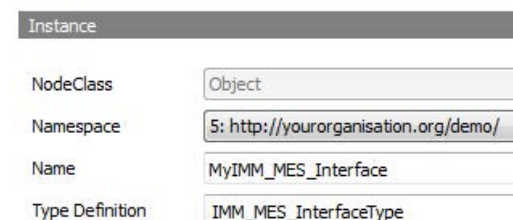
Deshalb wählen Sie unter "Additional Attributes" aus der Klappliste "String" aus. Falls Sie das Format für Nodelds nicht bedienbar ist, prüfen Sie die Einstellung "Enable editing of Nodelds" in den Settings; die Option muss aktiviert sein.



Als Nodeld tragen Sie im Beispiel "MyIMM_MES_Interface" ein.

10. Wählen Sie aus der Klappliste zu "Type Definition" den Eintrag "IMM_MES_InterfaceType" aus und tragen Sie einen Namen für die Instanz ein.

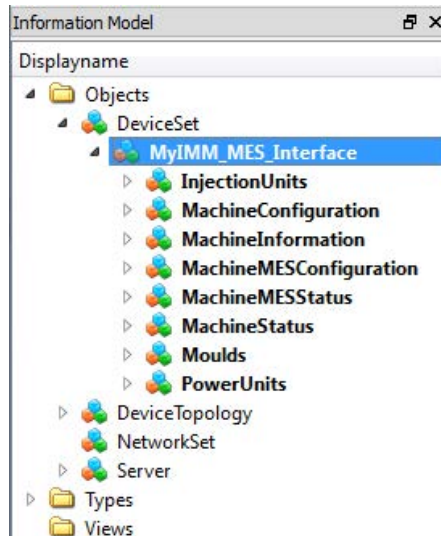
Im Beispiel verwenden Sie den Namen "MyIMM_MES_Interface":



11. Klicken Sie auf die Schaltfläche "OK" rechts unten im UaModeler.

12.Öffnen Sie den neuen Knoten "MyIMM_MES_Interface":

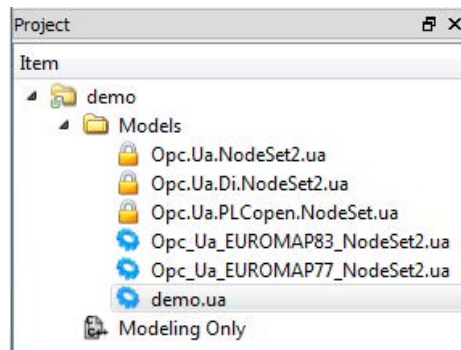
Dazu klicken Sie auf den Pfeil vor "MyIMM_MES_Interface":



Ergebnis: Im Adressraum des OPC UA-Servers befindet sich ein Objekt nach Euromap 77 bzw. Euromap 83 als Repräsentation der Spritzgießmaschine für z. B. ein MES (Management Execution System). Weitere Unterobjekte oder beliebige andere Konfigurationen können Sie auf gleiche Art und Weise erzeugen und damit den OPC UA-Server wie erforderlich gestalten.

13.Speichern Sie das Projekt.

14.Unter "Project" wählen Sie das Demo-Projekt aus:



15.Klicken Sie mit der rechten Maustaste auf "demo.ua" und wählen Sie aus dem Kontextmenu den Eintrag "Export XML".

UaModeler exportiert das Projekt in die Datei "demo.xml".

16.Schließen Sie UaModeler.

Schritt 2: PLC-Variablen zuordnen

Die folgende Beschreibung zeigt am Beispiel einer Variablen, wie Sie den Variablen und Methoden der Euromap 77 bestimmte PLC-Variablen und Server-Methoden zuordnen.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie mit einem Editor die Datei "demo.xml", die Sie im "Schritt 1" erstellt haben.

Suchen Sie in der XML-Datei, die UA-Variable mit der
NodeId="ns=1;s=MyIMM_MES_Interface.InjectionUnits.NodeVersion".

Dieser Variablen ordnen Sie im nächsten Schritt eine PLC-Variable zu.

2. Um dieser OPC UA-Variablen eine PLC-Variable zuzuordnen, fügen Sie in das XML-Element der OPC UA-Variablen eine Extension ein.

In diese Extension wiederum fügen Sie ein Element des Typs "<si:VariableMapping>" ein:

```
<UAVariable DataType="String" ParentNodeId="ns=1;s=MyIMM_MES_Interface.InjectionUnits" NodeId=
"ns=1;s=MyIMM_MES_Interface.InjectionUnits.NodeVersion" BrowseName="NodeVersion">
  <DisplayName>NodeVersion</DisplayName>
  <References>
    <Reference ReferenceType="HasProperty" IsForward="false">
      ns=1;s=MyIMM_MES_Interface.InjectionUnits</Reference>
    <Reference ReferenceType="HasTypeDefinition">i=68</Reference>
  </References>
  <Extensions>
    <Extension>
      <si:VariableMapping>"MyIMM_MES_Interface"."InjectionUnits.NodeVersion"
    </si:VariableMapping>
    </Extension>
  </Extensions>
  <Value>
</UAVariable>
```

3. Das XML-Element "<si:VariableMapping>" ist im XML-Namespace
"http://www.siemens.com/OPCUA/2017/SimaticNodeSetExtensions" definiert.

Deshalb müssen Sie dem XML-Element "<UANodeSet>" diesen Namensraum
hinzufügen, z. B. einmalig durch folgende Codezeile zu Beginn der XML-Datei:

```
<UANodeSet xmlns:si="http://www.siemens.com/OPCUA/2017/SimaticNodeSetExtensions"
```

Wenn Sie den Namensraum nicht hinzufügen, dann ist das Element
"<si:VariableMapping>" in der Datei nicht bekannt.

4. Speichern Sie die Datei "demo.xml" und schließen Sie den Editor.

Exkurs: PLC-Methoden zuordnen

Neben Variablen können Sie auch Methoden einer FB-Instanz (Anwenderprogramm bzw.
Funktionsbaustein als Repräsentation der Methode) zuordnen.

Um einer OPC UA-Methode eine FB-Instanz zuzuordnen, müssen Sie ebenfalls eine
Extension nach dem unten gezeigten Muster in der OPC UA-XML-Datei ergänzen. Beachten
Sie, dass das Suffix ".Method" ohne Anführungszeichen dem Instanznamen angehängt
werden muss.

In der Extension fügen Sie ein Element des Typs "<si:MethodMapping>" ein.

Die Eigenschaften einer OPC UA-Methode mit BrowseName "InputArguments" und "OutputArguments" sind OPC UA-Variablen und werden **nicht** zugeordnet.

```
<UAMethod ParentNodeId="ns=1;i=5003" NodeId=
"ns=1;s=MyIMM_MES_Interface.MachineConfiguration.SetServerTime" BrowseName="4:SetServerTime"
MethodDeclarationId="ns=2;i=7005">
  <DisplayName>SetServerTime</DisplayName>
  <References>
    <Reference ReferenceType="HasProperty">
      ns=1;s=MyIMM_MES_Interface.MachineConfiguration.SetServerTime.InputArguments</Reference>
    <Reference ReferenceType="HasComponent" IsForward="false">ns=1;i=5003</Reference>
  </References>
  <Extensions>
    <Extension>
      <si:MethodMapping>"MachineConfiguration.SetServerTime_IDB".Method</si:MethodMapping>
    </Extension>
  </Extensions>
</UAMethod>
```

Schritt 3: Server-Schnittstellen importieren

Um eine OPC UA-XML-Datei als "Server-Schnittstelle" zu importieren, gehen Sie folgendermaßen vor:

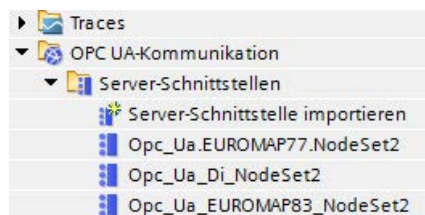
1. Öffnen Sie das STEP 7-Projekt.
2. Klicken Sie auf "OPC UA-Kommunikation > Server-Schnittstellen".
3. Doppelklicken Sie auf "Server-Schnittstelle importieren".
4. Im Dialog "Importieren" wählen Sie die Datei, die Sie als Server-Schnittstelle importieren wollen.

Im Beispiel ist das die Datei "demo.xml".

5. Klicken Sie auf die Schaltfläche "Import".
6. Importieren Sie auch die Dateien "Opc.Ua.Di.NodeSet2.xml", "Opc_Ua.EUROMAP77.NodeSet2.xml" und "Opc_Ua.EUROMAP83.NodeSet2.xml".

Die Datei "demo.xml" bezieht sich auf die erstgenannten Dateien.

Verwenden Sie stets aktuelle Versionen dieser Dateien, da STEP 7 keine fehlerhaften OPC UA-XML-Dateien übersetzt.



Die oben nach Euromap 77 Spezifikation erzeugte Abbildung der Maschine ist nun im Adressraum des OPC UA-Servers vorhanden.

Schritt 4: Im STEP 7-Projekt PLC-Variablen und Server-Methoden anlegen

Im Beispiel haben Sie der UA-Variablen mit der NodeId="ns=1;s=MyIMM_MES_Interface.InjectionUnits.NodeVersion" die PLC-Variable "MyIMM_MES_Interface"."InjectionUnits.NodeVersion" zugeordnet, siehe "Schritt 2: PLC-Variablen zuordnen".

Um die benötigte PLC-Variable mit STEP 7 (TIA Portal) anzulegen, gehen Sie folgendermaßen vor:

1. Erzeugen Sie einen Datenbaustein "MyIMM_MES_Interface".
2. Erzeugen Sie das DB-Element "InjectionUnits.NodeVersion". Verwenden Sie den kompatiblen SIMATIC-Datentyp zum OPC UA-Datentyp. Die UA-Variable des Beispiels besitzt den OPC UA-Datentyp "String" (DataType="String"). Der kompatible SIMATIC-Datentyp ist WSTRING.

MyIMM_MES_Interface				
Name	Datentyp	Startwert	Erreichbar aus HMI/OPC UA	Schreibbar aus HMI/OPC UA
▼ Static			<input type="checkbox"/>	<input type="checkbox"/>
InjectionUnits.NodeVersion	WString	WSTRING#"V2.5"	<input checked="" type="checkbox"/>	<input type="checkbox"/>

3. Übersetzen Sie das Projekt.
4. Laden Sie das Projekt in die CPU.

Import von exportieren OPC UA-XML-Dateien in eine S7-1500 CPU

Beachten Sie folgenden Hinweis, wenn Sie Server-Schnittstellen importieren, die aus dem OPC UA-XML-Export einer S7-1500 stammen.

Hinweis

Gesperrter Import für Namensraum "http://www.siemens.com/simatic-s7-opcua"

Sie können keine Server-Schnittstelle mit dem Namensraum "http://www.siemens.com/simatic-s7-opcua" in eine S7-1500 CPU importieren, da dieser Namensraum für S7-1500 CPUs reserviert (Standard-SIMATIC Server-Schnittstelle) und für den Import gesperrt ist.

Wenn Sie eine Server-Schnittstelle mit dem Namensraum "http://www.siemens.com/simatic-s7-opcua" importieren wollen, dann öffnen Sie die zu importierende Server-Schnittstelle (OPC UA-XML-Datei) und ändern den Namensraum an den entsprechenden Stellen. Die so geänderte Datei können Sie dann importieren.

Integrität der OPC UA-XML-Dateien

OPC UA-XML-Dateien repräsentieren den Server-Adressraum. Diese Dateien werden z. B. im Kontext von OPC UA Companion Spezifikationen von Ihnen nach Anpassung an die Applikation als Server-Schnittstelle importiert, mit der Hardware-Konfiguration in die S7-1500 CPU geladen und getestet.



WARNUNG

Keine Prüfung von importierten OPC UA-XML-Dateien

Schützen Sie diese OPC UA-XML-Dateien vor nicht autorisierten Manipulationen, da STEP 7 die Integrität dieser Dateien nicht prüft.

Empfehlung

Um bei einer Erweiterung bzw. Anpassung des Server-Adressraums Risiken zu minimieren, gehen Sie folgendermaßen vor:

1. Schützen Sie das Projekt (Projektnavigation: Security-Einstellungen > Einstellungen).
2. Exportieren Sie vor der Erweiterung oder Anpassung die entsprechende Server-Schnittstelle.
3. Überarbeiten Sie diese OPC UA-XML-Datei.
4. Importieren Sie die Datei erneut als Server-Schnittstelle.

Mapping der Datentypen

Die folgende Tabelle zeigt den kompatiblen SIMATIC-Datentyp zum jeweiligen OPC UA-Datentyp.

Ordnen Sie die Datentypen zu wie unten gezeigt (SIMATIC-Datentyp - OPC UA-Datentyp). Andere Zuordnungen sind nicht zugelassen. STEP 7 prüft nicht die Einhaltung dieser Regel und verhindert nicht eine falsche Zuordnung. Sie sind für die regelkonforme Auswahl und Zuordnung der Datentypen verantwortlich.

Die aufgelisteten Datentypen können Sie auch z. B. als Elemente von Strukturen/UDTs für Eingangs- und Ausgangsparameter von selbst erstellten Server-Methoden (UAMethod_InParameters und UAMethod_OutParameters) verwenden.

Tabelle 9- 10 Mapping der Datentypen

SIMATIC-Datentyp	OPC UA-Datentyp
BOOL	Boolean
SINT	SByte
INT	Int16
DINT	Int32
LINT	Int64
USINT	Byte
UINT	UInt16
UDINT	UInt32

SIMATIC-Datentyp	OPC UA-Datentyp
ULINT	UInt64
REAL	Float
LREAL	Double
LDT	DateTime
WSTRING	String
DINT	Enumeration (Encoding Int32) und alle davon abgeleiteten Datentypen
Anwenderdefinierter Datentyp erforderlich (UDT, user-defined data type) Der Anwenderdefinierte Datentyp muss mit dem Prefix "Union_" angelegt werden, z. B. "Union_MyDatatype", siehe Beispiel unter der Tabelle. Das erste Element (Selector) in diesem UDT muss den Datentyp "UDINT" besitzen.	UNION und alle davon abgeleiteten Datentypen

Anwenderdefinierter Datentyp für UNION erforderlich

Das folgende Bild zeigt die Variable "MyVariable", die den Datentyp "Union_MyDatatype" besitzt.

Dieser SIMATIC-Datentyp entspricht einer OPC UA-Variablen mit dem Datentyp UNION.

Das Bild zeigt ein Beispiel für die Deklaration: Bei Selector = 1, nimmt die Union einen ByteString auf, bei Selector = 2 einen WString.

Name	Datentyp
▼ Static	
■ ▼ MyVariable	"Union_MyDatatype"
■ Selector	UDInt
■ ▶ ByteString	Array[0..1] of Byte
■ WString	WString[42]

Weitere OPC UA Basisdatentypen nutzen

Neben den im Kapitel "Mapping von Datentypen" aufgeführten OPC UA-Datentypen und deren Entsprechungen auf SIMATIC-Seite gibt es noch folgende OPC UA-Basisdatentypen, die Sie ebenfalls nutzen können:

- OpcUa_NodeId
- OpcUa_QualifiedName
- OpcUa_Guid
- OpcUa_LocalizedText
- OpcUa_ByteString
- OpcUa_XmlElement

Voraussetzung für die Nutzung der oben genannten Basisdatentypen als Variablen im Anwenderprogramm: Die Basisdatentypen müssen als zusammengesetzte Datentypen vorliegen, die genauso strukturiert sind wie die entsprechenden OPC UA-Basisdatentypen.

- OpcUa_NodeId und OpcUa_QualifiedName liegen als Systemdatentyp vor; deshalb können Sie diese Datentypen für einzelne Variablen aber auch als Elemente einer Struktur nutzen.
- Für die übrigen Basisdatentypen müssen Sie einen PLC-Datentyp entsprechend der OPC UA-Spezifikation anlegen und anschließend als Element in einer Struktur verwenden, damit die Elemente über das Typedictionary aufgelöst werden können. Wie der PLC-Datentyp jeweils aussehen muss, ist im Folgenden für jeden einzelnen Basisdatentyp beschrieben. Ein Beispiel für eine Datenstruktur, in der z. B. der UDT "LocalizedText" verwendet wird, ist "EUInformation". EUInformation enthält Informationen zu EngineeringUnits. Ein Beispiel zur Umsetzung der Datenstruktur EUInformation finden Sie am Schluss der PLC-Datentypen-Beschreibungen.

Systemdatentyp "OPC-UA-NodeId"

Für den OPC UA Basisdatentyp "OpcUa_NodeId" entnehmen Sie der folgenden Tabelle die Bedeutung der Parameter. OPC-UA-NodeId nutzen Sie zur Identifizierung eines Knotens im OPC UA-Server.

Parameter	S7-Datentyp	Bedeutung
NamespaceIndex	UINT	Namensraumindex des Knotens im OPC UA-Server. Ein Knoten kann zum Beispiel eine Variable sein.
Identifizier	WSTRING[254]	Die Bezeichnung für den Knoten (Objekt oder Variable) ist abhängig vom Identifizier-Typ: <ul style="list-style-type: none"> • Numerischer Identifizier: Der Knoten wird mit einer Zahl bezeichnet, zum Beispiel "12345678". • String-Identifizier: Der Knoten wird mit einem Namen bezeichnet, zum Beispiel "MeineVariable". Groß- und Kleinschreibung wird unterschieden.
IdentifizierType	UDINT	Typ des Identifiziers <ul style="list-style-type: none"> • 0: Numerischer Identifizier • 1: String-Identifizier • 2: GUID • 3: Opaque

Systemdatentyp "OPC-UA-QualifiedName"

Entnehmen Sie der folgenden Tabelle den Aufbau des Systemdatentyps "OPC-UA-QualifiedName":

Name	S7-Datentyp	Bedeutung
NamespaceIndex	UINT	Der Namespaceindex des Namens.
Name	WSTRING[64]	Name des Knotens oder der Variablen.

UDT "Guid"

Legen Sie für den Basisdatentyp "Guid" folgenden PLC-Datentyp an. Die exemplarisch eingesetzten Defaultwerte können Sie auch anders setzen.

Guid			
	Name	Data type	Default value
1	Data 1	UDInt	16#11223344
2	Data 2	UInt	16#5566
3	Data 3	UInt	16#7788
4	Data 4	ULInt	16#99AABBCCDDEEFF11

UDT "LocalizedText"

Legen Sie für den Basisdatentyp "LocalizedText" folgenden PLC-Datentyp an:

LocalizedText			
	Name	Data type	Default value
1	EncodingByte	Byte	16#3
2	Locale	WString[5]	WSTRING#'de-DE'
3	Text	WString[255]	WSTRING#'Text'

Das EncodingByte gibt an, welche Felder (Locale bzw. Text) vorhanden sind:

EncodingByte	Bedeutung
0	Die Felder Locale und Text sind leer
1	Das Feld Locale hat Inhalt, das Feld Text ist leer
2	Das Feld Locale ist leer, das Feld Text hat Inhalt
3	Die Felder Locale und Text haben Inhalt

UDT "ByteString"

Legen Sie für den Basisdatentyp "ByteString" folgenden PLC-Datentyp an; hier ist z. B. ein ByteString-Array mit 12 Elementen gewählt:

ByteString			
	Name	Data type	Default value
1	ByteString	Array[0..11] of Byte	
2	ByteString[0]	Byte	16#0
3	ByteString[1]	Byte	16#0
4	ByteString[2]	Byte	16#0
5	ByteString[3]	Byte	16#0
6	ByteString[4]	Byte	16#0
7	ByteString[5]	Byte	16#0
8	ByteString[6]	Byte	16#0
9	ByteString[7]	Byte	16#0
10	ByteString[8]	Byte	16#0
11	ByteString[9]	Byte	16#0
12	ByteString[10]	Byte	16#0
13	ByteString[11]	Byte	16#0

UDT "XmlElement"

Ein XmlElement ist ein serialisiertes XML-Fragment (UTF-8-String).

Legen Sie für den Basisdatentyp "XmlElement" folgenden PLC-Datentyp an:

XmlElement			
	Name	Data type	Default value
1	XmlElement	WString	WSTRING#''

Beispiel: Struktur von EUInformation mit UDT "LocalizedText"

▼ Velocity	*EUInformation*		
■ NamespaceUri	WString[255]	WSTRING#'http://yourorganization.org'	Identifies the organization (company, standards organization) that defines the EUInformation
■ UnitId	DInt	1	Identifier for programmatic evaluation. -1 is used if a unitid is not available.
■ ▼ DisplayName	*LocalizedText*		
■ Encoding...	Byte	16#3	
■ Locale	WString[5]	WSTRING#'en-EN'	
■ Text	WString[255]	WSTRING#'m/s'	
■ ▼ Description	*LocalizedText*		
■ Encoding...	Byte	16#3	
■ Locale	WString[5]	WSTRING#'en-EN'	
■ Text	WString[255]	WSTRING#'meter per second'	

MinimumSamplingInterval-Attribut von Variablen

Neben "Value", "DataType" und "AccessLevel" können Sie in der XML-Datei, die den Server-Adressraum repräsentiert, auch das Attribut "MinimumSamplingInterval" für eine Variable setzen. Das Attribut gibt an, wie schnell der Server den Variablenwert abtasten kann.

Für den OPC UA-Server der S7-1500 CPU gilt folgende Regel für mögliche Werte von MinimumSamplingInterval:

- Verwenden Sie nur positive Ganzzahlen zwischen 0 und 4294967.

Kommazahlen (z. B. 0,55) sind nicht zulässig, da sie falsch interpretiert werden. Bei Kommazahlen bis zu 7 Stellen wird das Komma entfernt, bei Kommazahlen mit mehr als 7 Stellen wird der Wert auf das Maximum gestellt: 4294967,295.

Negative Zahlen werden auf den maximal möglichen Wert 4294967,295 gesetzt.

9.3.5.2 Schreib- und Leserechte für CPU-Variablen koordinieren

Definition von Schreib- und Leserechten im Informationsmodell (OPC UA-XML)

Im OPC UA-Informationsmodell regelt das Attribut "AccessLevel" den Zugriff auf Variablen.

AccessLevel ist bitweise definiert:

Bit 0 = CurrentRead und Bit 1 = CurrentWrite. Die Bedeutung der Bitkombinationen ergibt sich wie folgt:

- AccessLevel = 0: kein Zugriff
- AccessLevel = 1: read only
- AccessLevel = 2 write only
- AccessLevel = 3: read+write


Beispiel für die Vergabe von Schreib- und Leserechten (read+write)

```
<UAVariable NodeId="ns=3;s="Data_block_2";."Static_1";"
BrowseName="3:Static_1"
ParentNodeId="ns=3;s="Data_block_2";"
DataType="INT"
AccessLevel="3">
  <DisplayName>Static_1</DisplayName>
```

Definition von Schreib- und Leserechten in STEP 7

Beim Definieren von Variablen legen Sie die Zugriffsrechte fest mit den Eigenschaften "Erreichbar aus HMI/OPC UA" und "Schreibbar aus HMI/OPC UA".

Beispiel für die Vergabe von Schreib- und Leserechten

Name	Data type	Accessible from HMI/OPC UA	Writable from HMI/OPC UA
Static_1	Int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<Add new>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Zusammenspiel der Schreib- und Leserechte

Wenn Sie eine OPC UA-Server-Schnittstelle importiert haben und in dieser OPC UA-XML-Datei sind AccessLevel-Attribute gesetzt, dann resultieren die Schreib- und Leserechte aus der Regel: Wirksam sind die geringsten Zugriffsrechte aus beiden Einstellungen.

Beispiel

- AccessLevel = 1 (read only) in der OPC UA-Server-Schnittstelle
- Sowohl "Erreichbar aus HMI/OPC UA" als auch "Schreibbar aus HMI/OPC UA" ist in der PLC-Variablentabelle aktiviert

Ergebnis: Die Variable wird nur gelesen.

Regeln

Wenn Schreibrechte erforderlich sind:

- AccessLevel = 2 oder 3
- "Schreibbar aus HMI/OPC UA" aktiviert

Wenn Leserechte erforderlich sind:

- AccessLevel = 1 (AccessLevel 3 ist auch möglich, aber irreführend. Die Einstellung suggeriert, ein OPC UA-Client hat Schreib- und Leserechte)
- "Erreichbar aus HMI/OPC UA" aktiviert, "Schreibbar aus HMI/OPC UA" deaktiviert

Wenn weder Lese- noch Schreibrechte eingeräumt werden sollen (kein Zugriff):

- AccessLevel = 0
- "Erreichbar aus HMI/OPC UA" deaktiviert

Um jeglichen Zugriff zu sperren, muss nur eine der beiden Bedingungen erfüllt sein. Überdenken Sie in diesem Fall, ob die Variable in der OPC UA-Server-Schnittstelle überhaupt benötigt wird.

Zugriffstabelle

"Erreichbar aus HMI/OPC UA" muss gesetzt sein, damit überhaupt ein Zugriff per OPC UA möglich ist. "Schreibbar aus HMI/OPC UA" muss gesetzt sein, damit ein OPC UA-Client auf eine Variable / ein DB-Element schreiben kann.

Der Tabelle entnehmen Sie das resultierende Zugriffsrecht.

Tabelle 9- 11 Zugriffstabelle

OPC UA-XML	STEP 7 (TIA Portal) z. B. Variablen-tabelle		
AccessLevel	Erreichbar aus HMI/OPC UA	Schreibbar aus HMI/OPC UA	Resultierendes Zugriffsrecht
0	x	x	Kein Zugriff
x	0	x	Kein Zugriff
1	aktiviert	x	Read only
2	aktiviert	deaktiviert	Kein Zugriff
3	aktiviert	deaktiviert	Read only
2	aktiviert	aktiviert	Write only
3	aktiviert	aktiviert	Read+write

(x = don't care)

9.3.5.3 Hinweise zu Mengengerüsten bei Nutzung von Server-Schnittstellen

Wenn Sie OPC UA-Server-Schnittstellen verwenden, dann müssen Sie abhängig von Leistungsklasse der S7-1500 CPU Obergrenzen für folgende Objekte berücksichtigen:

- Anzahl der Server-Interfaces
- Anzahl der OPC UA Nodes
- Datenmenge der Ladeobjekte
- Falls Sie Methoden implementiert haben: Anzahl Server-Methoden bzw. Server-Methoden-Instanzen

Mengengerüste für OPC UA-Server-Schnittstellen

In folgender Tabelle sind die Mengengerüste der S7-1500 CPUs dokumentiert, die auch beim Übersetzen und Laden einer Konfiguration berücksichtigt werden.

Eine Verletzung der Mengengerüste wird mit einer Fehlermeldung quittiert.

Tabelle 9- 12 Mengengerüste für OPC UA-Server-Schnittstellen

Technisches Datum	CPU 1510SP (F) CPU 1511 (C/F/T/TF) CPU 1512C CPU 1512SP (F) CPU 1513 (F)	CPU 1505 (S/SP/SP F/SP T/SP TF) CPU 1515 (F/T/TF) CPU 1515 SP PC (F/T/TF) CPU 1516 (F/T/TF)	CPU 1507S (F) CPU 1517 (F/T/TF) CPU 1518 (F)
Maximale Anzahl OPC UA-Server-Schnittstellen (Informationsmodelle)	10	10	10
Maximale Anzahl von OPC UA Nodes	1000	5000	30000
Maximale Größe ladbarer OPC UA-Server-Schnittstellen	1024 KB	5120 KB	15360 KB
Maximale Anzahl nutzbarer Server-Methoden bzw. max. Anzahl Server-Methoden-Instanzen (Anweisungen OPC-UA_ServerMethodPre, OPC-UA_ServerMethodPost)	20	50	100

Hinweis

Memory Card mit ausreichender Kapazität verwenden

Je nach Anzahl und Umfang der zu ladenden Server-Schnittstellen kann der Fall eintreten, dass das Laden der Hardware-Konfiguration in die CPU mit einer Fehlermeldung abgebrochen wird.

Die Ursache des Abbruchs wird nicht im Diagnosepuffer angezeigt.

Verwenden Sie eine Memory Card mit ausreichender Kapazität und deaktivieren Sie nicht benötigte Server-Schnittstellen (Eigenschaften des Server-Interfaces, Option "Server-Schnittstelle aktivieren und zur CPU laden").

Routing

10.1 S7-Routing

Definition S7-Routing

S7-Routing ist die Übertragung von Daten über S7-Subnetzgrenzen hinweg. Hierbei können Sie Informationen von einem Sender über verschiedene S7-Subnetze hinweg zu einem Empfänger verschicken. Der Übergang von einem S7-Subnetz zu einem oder mehreren anderen Subnetzen erfolgt im S7-Router. Der S7-Router ist ein Gerät, welches über die Schnittstellen zu den betreffenden S7-Subnetzen verfügt. S7-Routing ist über verschiedene S7-Subnetze (PROFINET/Industrial Ethernet und/oder PROFIBUS) möglich.

Voraussetzungen für S7-Routing

- Alle erreichbaren Geräte in einem Netz sind innerhalb eines Projekts in STEP 7 konfiguriert und geladen worden.
- Alle am S7-Routing beteiligten Geräte müssen Informationen darüber erhalten, welche S7-Subnetze über welche S7-Router erreicht werden können (= Routing-Information). Die Routing-Information erhalten die Geräte durch das Laden der Hardwarekonfiguration in die CPUs, da die CPUs die Rolle eines S7-Routers spielt.

Bei einer Topologie mit mehreren hintereinanderliegenden S7-Subnetzen müssen Sie folgende Reihenfolge beim Laden einhalten: zunächst laden Sie die Hardwarekonfiguration in die CPU(s), die direkt mit dem selben S7-Subnetz wie das PG/PC verbunden sind, dann laden Sie nacheinander die CPUs der dahinterliegenden S7-Subnetzen, vom nächsten S7-Subnetz bis zum am weitesten entfernten S7-Subnetz.
- Das PG/PC, mit dem Sie eine Verbindung über einen S7-Router herstellen wollen, muss dem S7-Subnetz zugeordnet sein, an dem es auch tatsächlich physikalisch angeschlossen ist. Das PG/PC können Sie in STEP 7 unter Online & Diagnose > Online-Zugänge > Verbindung mit Schnittstelle/Subnetz einen PG/PC zuordnen.
- Für S7-Subnetze vom Typ PROFIBUS: Die CPU muss entweder als DP-Master konfiguriert sein oder wenn sie als DP-Slave konfiguriert ist, muss in den Eigenschaften der DP-Schnittstelle des DP-Slaves das Kontrollkästchen "Test, Inbetriebnahme und Routing" aktiviert sein.
- S7-Routing für HMI-Verbindungen ist ab STEP 7 V13 SP1 möglich.

Hinweis

Firewall und S7-Routing

Eine Firewall erkennt die IP-Adresse des Senders beim S7-Routing nicht, wenn der Sender sich außerhalb des an die Firewall angrenzenden S7-Subnetzes befindet.

Einen Überblick, welche Geräte die Funktion "S7-Routing" unterstützen, finden Sie in diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/584459>).

S7-Routing für Online-Verbindungen

Sie können mit dem PG/PC Geräte über S7-Subnetze hinweg erreichen, um beispielsweise:

- Anwenderprogramme zu laden
- eine Hardware-Konfiguration zu laden
- um Test- und Diagnosefunktionen ausführen zu können

Im folgenden Bild ist die CPU 1 S7-Router zwischen S7-Subnetz 1 und S7-Subnetz 2.

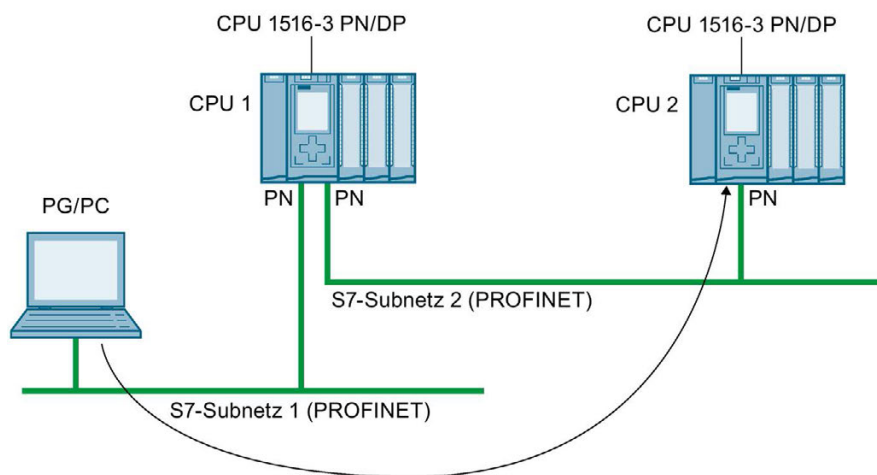


Bild 10-1 S7-Routing: PROFINET - PROFINET

Im folgenden Bild ist der Zugriff von einem PG über PROFINET nach PROFIBUS dargestellt. Die CPU 1 ist S7-Router zwischen S7-Subnetz 1 und S7-Subnetz 2; die CPU 2 ist S7-Router zwischen S7-Subnetz 2 und S7-Subnetz 3.

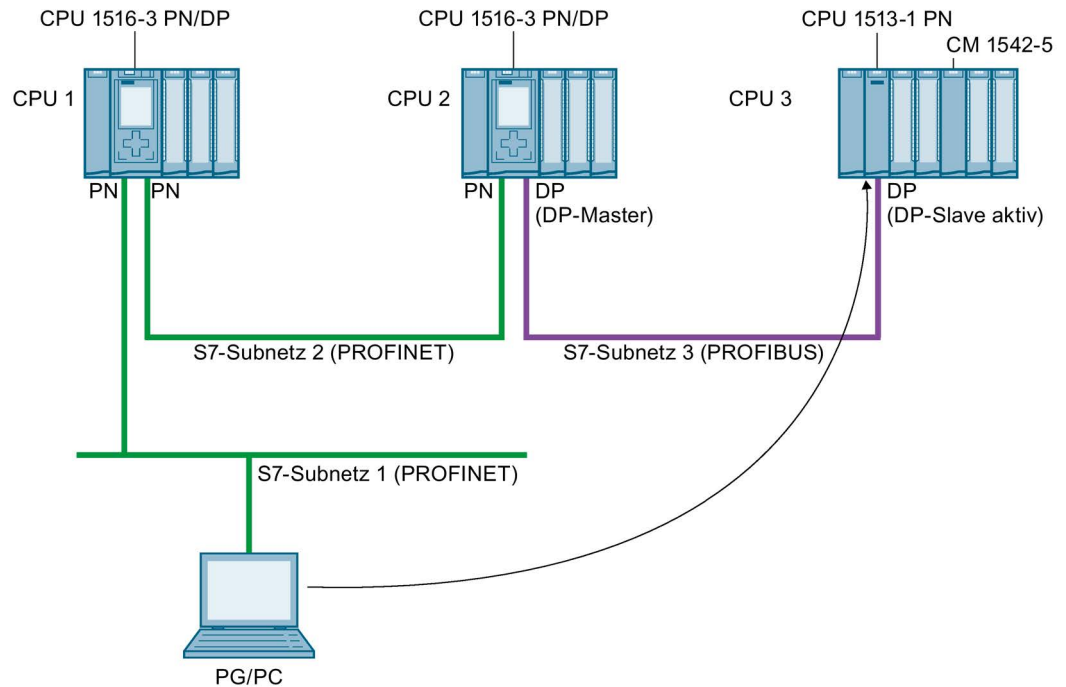


Bild 10-2 S7-Routing: PROFINET - PROFIBUS

S7-Routing für HMI-Verbindungen

Sie haben die Möglichkeit, eine S7-Verbindung von einem HMI zu einer CPU über unterschiedliche Subnetze (PROFIBUS und PROFINET bzw. Industrial Ethernet) einzurichten. Im folgenden Bild ist die CPU 1 S7-Router zwischen S7-Subnetz 1 und S7-Subnetz 2.

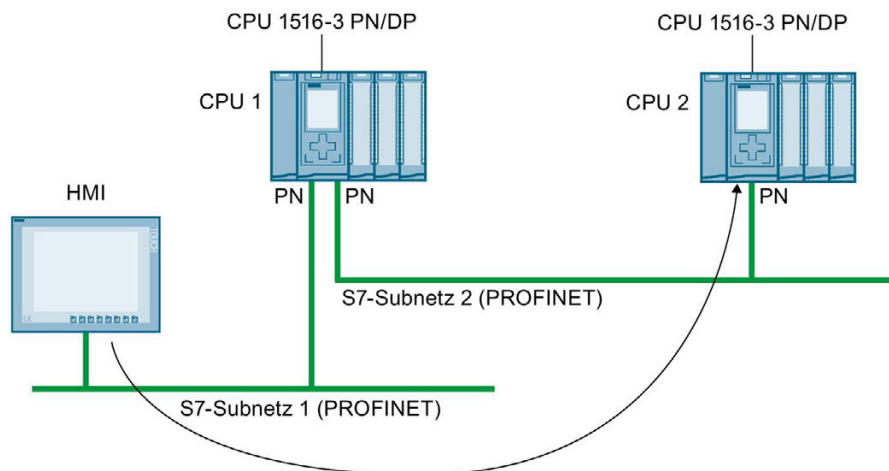


Bild 10-3 S7-Routing über HMI-Verbindung

S7-Routing für CPU-CPU-Kommunikation

Sie haben die Möglichkeit, eine S7-Verbindung von einer CPU zu einer anderen über unterschiedliche Subnetze (PROFIBUS und PROFINET bzw. Industrial Ethernet) einzurichten. Das Vorgehen ist an Beispielen im Kapitel S7-Kommunikation (Seite 121) beschrieben.

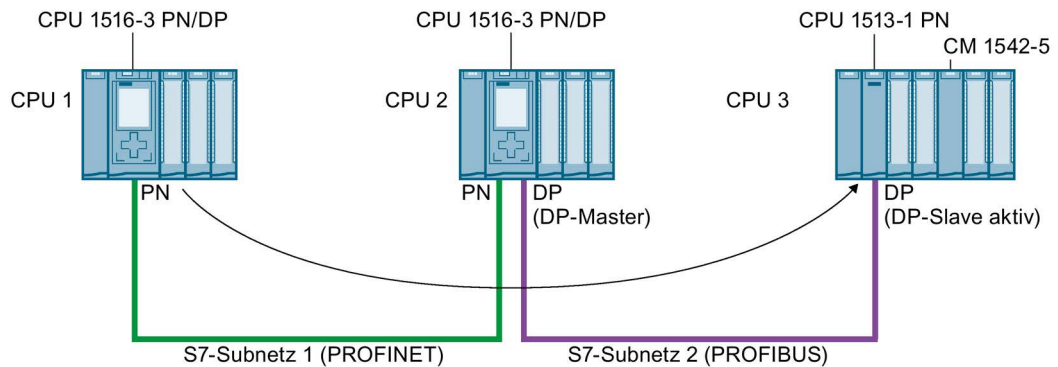


Bild 10-4 S7-Routing über CPU-CPU-Kommunikation

S7-Routing nutzen

Für die CPU wählen Sie im Dialog "Online verbinden" von STEP 7 die PG/PC-Schnittstelle und das S7-Subnetz aus. Das S7-Routing wird automatisch durchgeführt.

Anzahl der Verbindungen für S7-Routing

Die Anzahl der Verbindungen, die für S7-Routing in den S7-Routern (CPUs, CMs bzw. CPs) zur Verfügung stehen, finden Sie in den Technischen Daten, in den Gerätehandbüchern der jeweiligen CPU/CM/CP.

S7-Routing: Applikationsbeispiel

Das folgende Bild zeigt Ihnen als Applikationsbeispiel die Fernwartung einer Anlage durch ein PG. Die Verbindung kommt hierbei über zwei S7-Subnetz hinweg über eine Modemverbindung zu Stande.

Eine Fernverbindung über TeleService projektieren Sie in STEP 7 über "Online-Zugänge" bzw. "Online verbinden".

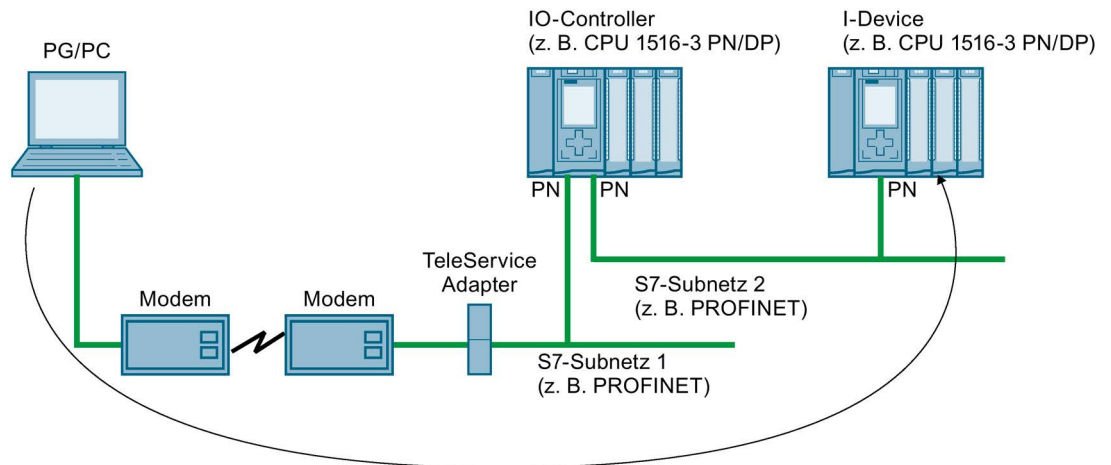


Bild 10-5 Fernwartung einer Anlage über TeleService

Weitere Informationen

- Die Belegung von Verbindungsressourcen beim S7-Routing ist beschrieben im Kapitel Belegung von Verbindungsressourcen (Seite 246).
- Ausführliche Informationen zum Einrichten von TeleService finden Sie in der Online-Hilfe STEP 7.
- Weitere Informationen zu S7-Routing und TeleService Adaptern finden Sie über die Suche im Internet unter folgenden Links:
 - Gerätehandbuch Industrie Software Engineering Tools TS Adapter IE Basic (<http://support.automation.siemens.com/WW/view/de/51311100>)
 - Downloads zum TS Adapter (<http://support.automation.siemens.com/WW/view/de/10805406/133100>)

Siehe auch

HMI-Kommunikation (Seite 68)

10.2 Datensatz-Routing

Definition Datensatz-Routing

Daten können von einer Engineering Station von PROFINET aus über verschiedene Netzwerke hinweg an Feldgeräte gesendet werden. Da die Engineering Station die Feldgeräte über genormte Datensätze anspricht und diese Datensätze über S7-Geräte geroutet werden, hat sich für diese Art des Routings der Begriff "Datensatz-Routing" etabliert.

Die Daten, die beim Datensatz-Routing versendet werden, beinhalten außer der Parametrierung für die beteiligten Feldgeräte (DP-Slaves) auch gerätespezifische Informationen, z. B. Sollwerte, Grenzwerte.

Datensatz-Routing wird z. B. eingesetzt, wenn Feldgeräte verschiedener Hersteller zum Einsatz kommen. Die Feldgeräte werden zur Parametrierung und Diagnose über genormte Datensätze (PROFINET) angesprochen.

Datensatz-Routing mit STEP 7

Sie können mit STEP 7 Datensatz-Routing durchführen, indem Sie über die TCI-Schnittstelle (Tool Calling Interface) ein Gerätetool (z. B. PCT) aufrufen und Aufrufparameter übergeben. Das Gerätetool nutzt für die Kommunikation mit dem Feldgerät die Kommunikationswege, die auch STEP 7 nutzt.

Für diese Art des Routings ist außer der Installation des TCI-Tools auf dem STEP 7-Rechner keine Projektierung erforderlich.

Beispiel: Datensatz-Routing mit dem Port Configuration Tool (PCT)

Mit dem Port Configuration Tool (PCT) ist es möglich, die IO-Link Master der ET200 zu konfigurieren und daran angeschlossene IO-Link Devices zu parametrieren. Die Subnetze sind über Datensatz-Router verbunden. Datensatz-Router sind z. B. CPUs, CPs, IMs, IO-Link Master.

Welche Konstellationen von Datensatz- Routern das PCT unterstützt, finden Sie in diesem FAQ (<http://support.automation.siemens.com/WW/view/de/87611392>).

Das folgende Bild zeigt eine Beispielkonfiguration für Datensatz-Routing mit PCT.

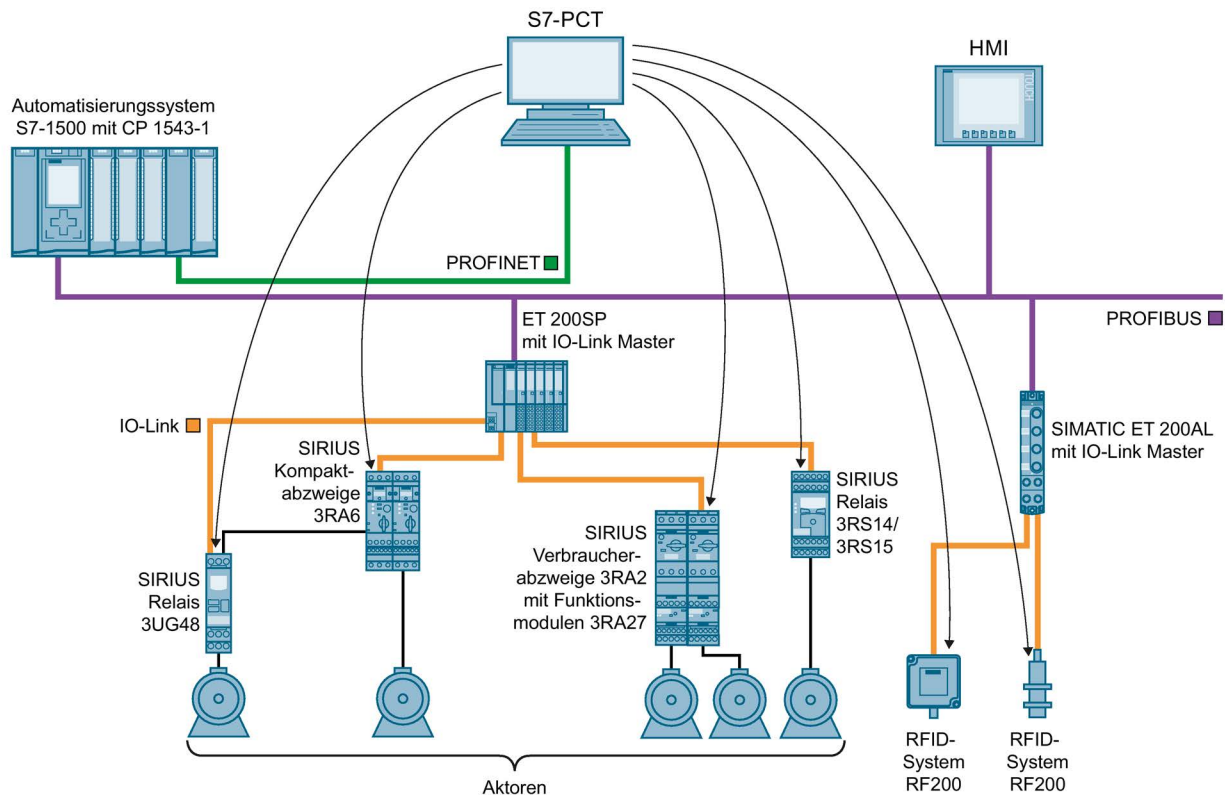


Bild 10-6 Beispielkonfiguration für Datensatz-Routing mit PCT

Weitere Informationen

- Welcher Unterschied zwischen "normalen Routing und Datensatzrouting besteht, finden Sie in diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/7000978>).
- Ob die eingesetzte CPU, der CP oder das CM Datensatz-Routing unterstützt, finden Sie in den entsprechenden Gerätehandbüchern beschrieben.
- Die Belegung von Verbindungsressourcen beim Datensatz-Routing ist beschrieben im Kapitel Belegung von Verbindungsressourcen (Seite 246).
- Weitere Informationen zur Konfiguration mit STEP 7 finden Sie in der Onlinehilfe STEP 7.

Verbindungsressourcen

11.1 Verbindungsressourcen einer Station

Einleitung

Einige Kommunikationsdienste benötigen Verbindungen. Verbindungen belegen im Automatisierungssystem (Station) Ressourcen. Die Verbindungsressourcen bekommt die Station von den CPUs, Kommunikationsprozessoren (CP) und Kommunikationsmodulen (CM) zur Verfügung gestellt.

Verbindungsressourcen einer Station

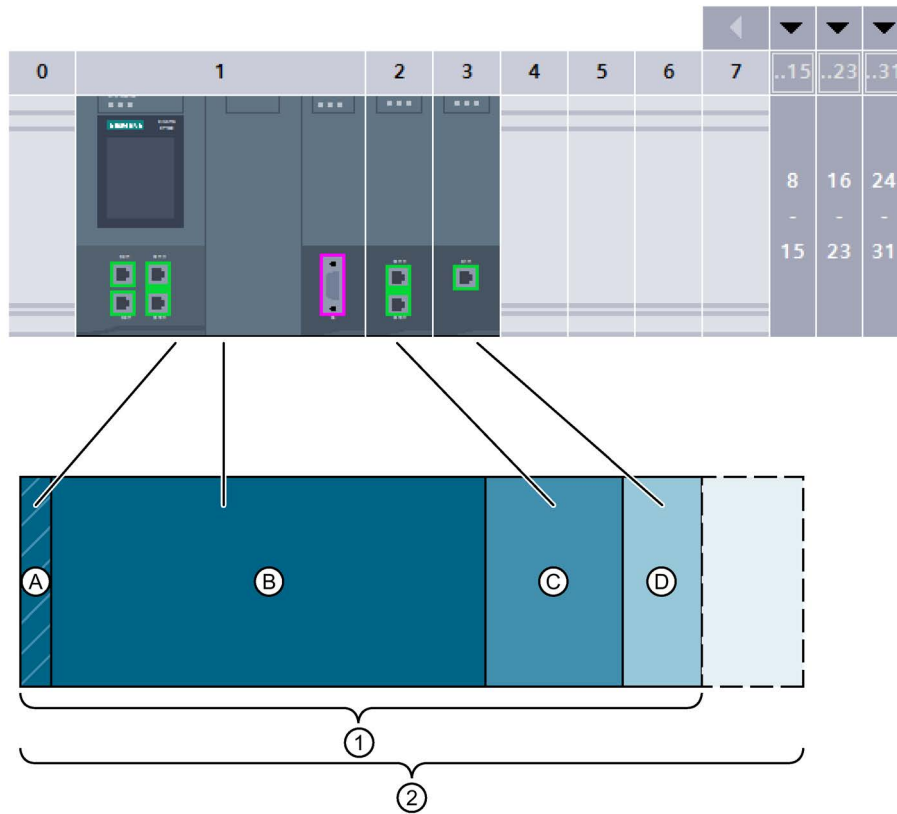
Die zur Verfügung stehenden Verbindungsressourcen sind abhängig von den eingesetzten CPUs, CPs und CMs und dürfen eine maximale Anzahl pro Station nicht überschreiten.

Die maximale Anzahl der Ressourcen einer Station wird durch die CPU bestimmt.

Jede CPU bringt reservierte Verbindungsressourcen für PG-, HMI- und Webserver-Kommunikation mit. Damit ist z. B. sichergestellt, dass ein PG immer mindestens eine Online-Verbindung mit der CPU aufbauen kann, unabhängig davon, wie viele andere Kommunikationsdienste bereits Verbindungsressourcen belegen.

Daneben gibt es dynamische Ressourcen. Die Differenz zwischen der maximalen Anzahl der Verbindungsressourcen und der Anzahl reservierter Verbindungsressourcen ist die maximale Anzahl dynamischer Verbindungsressourcen. Aus dem Pool der dynamischen Verbindungsressourcen bedienen sich die Kommunikationsdienste PG-Kommunikation, HMI-Kommunikation, S7-Kommunikation, Open User Communication, Web-Kommunikation und sonstige Kommunikation (z. B. OPC UA).

Das folgende Bild zeigt beispielhaft, wie einzelne Komponenten Verbindungsressourcen einer S7-1500-Station zur Verfügung stellen.



- ① Verfügbare Verbindungsressourcen der Station, davon
- A Reservierte Verbindungsressourcen der Station
 - A + B Verbindungsressourcen der CPU 1518
 - C Verbindungsressourcen des Kommunikationsmoduls CM 1542-1
 - D Verbindungsressourcen des Kommunikationsprozessors CP 1543-1
- ② Maximale Verbindungsressourcen der Station am Beispiel einer Konfiguration aus CPU 1518, CM 1542-1 und CP 1543-1

Bild 11-1 Verbindungsressourcen einer Station

Anzahl Verbindungsressourcen einer Station

Tabelle 11- 1 Maximal unterstützte Verbindungsressourcen für einige CPU-Typen

Verbindungsressourcen einer Station	1511 1511C	1512C 1513	1515	1516	1517	1518
Maximal Verbindungsressourcen der Station	96	128	192	256	320	384
davon reserviert	10					
davon dynamisch	86	118	182	246	310	374
Verbindungsressourcen der CPU	64	88	108	128	160	192
Max. zusätzlich nutzbare Verbindungsressourcen durch Stecken von CMs/CPs	32	40	84	128	160	192
Zusätzliche Verbindungsressourcen CM 1542-1	64					
Zusätzliche Verbindungsressourcen CP 1543-1	118					
Zusätzliche Verbindungsressourcen CM 1542-5	40					
Zusätzliche Verbindungsressourcen CP 1542-5	16					

Wieviele Verbindungsressourcen eine CPU bzw. ein Kommunikationsmodul unterstützt, finden Sie in den Gerätehandbüchern in den Technischen Daten beschrieben.

Beispiel

Sie haben eine CPU 1518-4PN/DP mit einem Kommunikationsmodul CM 1542-1 und einem Kommunikationsprozessor CP 1542-5 konfiguriert.

- Maximale Verbindungsressourcen der Station: **384**
- Verfügbare Verbindungsressourcen:
 - CPU 1518-4 PN/DP: 192
 - CM 1542-1: 64
 - CP 1542-5: 16
 - Gesamt: **272**

Der Aufbau stellt 272 Verbindungsressourcen zur Verfügung. Durch Hinzufügen weiterer Kommunikationsmodule kann die Station maximal 112 weitere Verbindungsressourcen unterstützen.

Reservierte Verbindungsressourcen

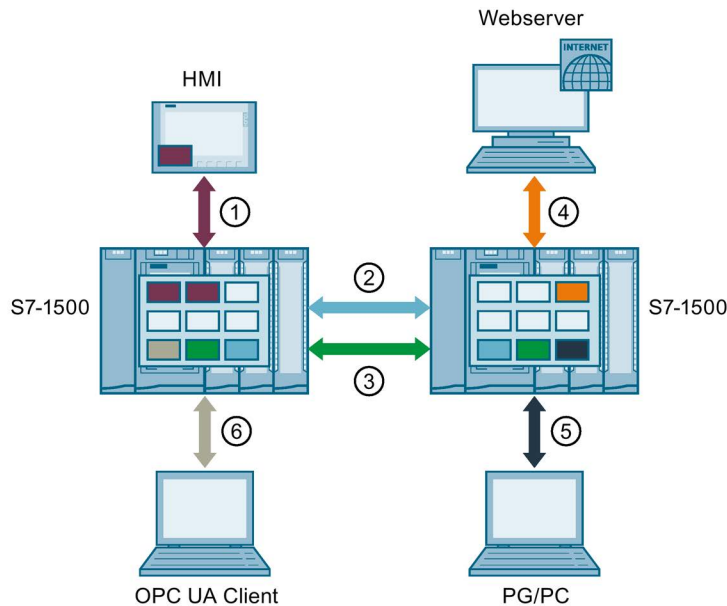
Für Stationen mit S7-1500 CPU, ET 200SP CPU und ET 200pro CPU auf Basis S7-1500 sind 10 Verbindungsressourcen reserviert:

- 4 für PG-Kommunikation, die von STEP 7 benötigt werden für z. B. Test- und Diagnosefunktionen oder das Laden in die CPU
- 4 für HMI-Kommunikation, die durch die ersten, in STEP 7 projektierten HMI-Verbindungen belegt werden
- 2 für Kommunikation zum Webserver

11.2 Belegung von Verbindungsressourcen

Überblick - Belegung von Verbindungsressourcen

Das folgende Bild zeigt, wie verschiedene Verbindungen die Ressourcen der S7-1500 belegen.



- ① HMI-Kommunikation: siehe unten
- ② Open User Communication: Verbindungen der Open User Communication belegen in jedem Endpunkt eine Verbindungsressource.
- ③ S7-Kommunikation: Verbindungen der S7-Kommunikation belegen in jedem Endpunkt eine Verbindungsressource.
- ④ Web-Kommunikation: Die Webserver-Verbindung belegt in der Station mindestens eine Verbindungsressource. Die Anzahl der belegten Verbindungen hängt vom Browser ab.
- ⑤ PG-Kommunikation: Die PG-Verbindung belegt in der Station eine Verbindungsressource.
- ⑥ OPC UA-Kommunikation: Jede Session, die der OPC UA-Server der CPU mit einem OPC UA-Client aufbaut, belegt in der Regel eine Verbindungsressource (Sonstige Kommunikation) in der Station.

- Verbindungsressource für HMI-Kommunikation
- Verbindungsressource für OpenUser-Kommunikation
- Verbindungsressource für S7-Kommunikation
- Verbindungsressource für Web-Kommunikation
- Verbindungsressource für PG-Kommunikation
- Verbindungsressource für Sonstige Kommunikation (z. B. OPC UA)

Bild 11-2 Belegung von Verbindungsressourcen

Verbindungsressourcen für HMI-Kommunikation

Bei HMI-Kommunikation ist die Belegung von Verbindungsressourcen in der Station abhängig vom eingesetzten HMI-Gerät.

Tabelle 11- 2 Maximal belegte Verbindungsressourcen für verschiedene HMI-Geräte

HMI-Gerät	max. belegte Verbindungsressourcen der Station pro HMI-Verbindung
Basic Panel	1
Comfort Panel	2 ¹
RT Advanced	2 ¹
RT Professional	3

¹ Wenn Sie keine Systemdiagnose und keine Meldungsprojektierung nutzen, dann belegt die Station pro HMI-Verbindung nur eine Verbindungsressource.

Beispiel: Sie haben für eine CPU 1516-3 PN/DP folgende HMI-Verbindungen konfiguriert:

- Zwei HMI-Verbindungen zu einem HMI TP700 Comfort. (jeweils 2 Verbindungsressourcen)
- Eine HMI-Verbindung zu einem HMI KTP1000 Basic. (1 Verbindungsressource)

Insgesamt werden in der CPU 5 Verbindungsressourcen für HMI-Kommunikation belegt.

Verbindungsressourcen für Routing

Für die Übertragung von Daten über S7-Subnetze hinweg ("S7-Routing") wird eine S7-Verbindung zwischen zwei CPUs aufgebaut. Die S7-Subnetze sind über Netzübergänge, sogenannte S7-Router miteinander verbunden. CPUs, CMs und CPs in S7-1500 sind S7-Router.

Für eine geroutete S7-Verbindung gilt folgendes:

- Eine geroutete Verbindung belegt in beiden Endpunkten jeweils eine Verbindungsressource, STEP 7 zeigt diese Verbindungsressourcen in der Tabelle "Verbindungsressourcen" an.
- Im S7-Router werden zwei spezielle Verbindungsressourcen für S7-Routing belegt. STEP 7 zeigt die speziellen Verbindungsressourcen für S7-Routing nicht in der Tabelle "Verbindungsressourcen" an. Die Anzahl der Ressourcen für S7-Routing ist CPU abhängig. Sie finden der Ressourcen für S7-Routing in den Technischen Daten der CPU unter "Anzahl S7-Routing Verbindungen".

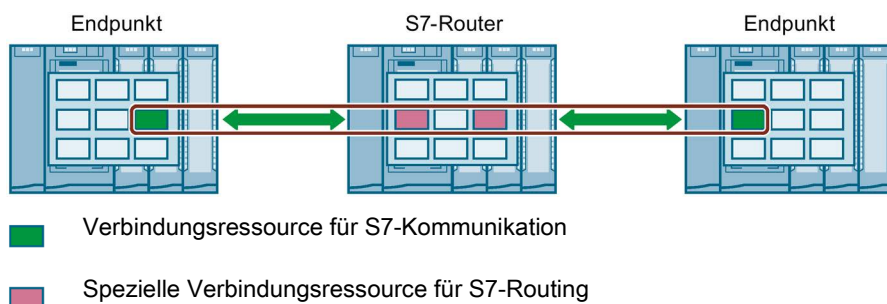


Bild 11-3 Verbindungsressourcen beim S7-Routing

Datensatz-Routing ermöglicht ebenfalls die Übertragung von Daten über S7-Subnetze hinweg, von einer Engineering Station, die am PROFINET angeschlossen ist, über PROFIBUS an diverse Feldgeräte.

Wie beim S7-Routing werden auch beim Datensatz-Routing in jedem Datensatz-Router zwei der speziellen Verbindungsressourcen für S7-Routing belegt.

Hinweis

Verbindungsressourcen beim Datensatz-Routing

Im Datensatz-Router werden beim Datensatz-Routing zwei spezielle Verbindungsressourcen für S7-Routing belegt. Weder die Datensatzverbindung, noch die belegten Verbindungsressourcen werden in der Tabelle Verbindungsressourcen angezeigt.

Wann werden Verbindungsressourcen belegt?

Der Zeitpunkt für die Belegung der Verbindungsressourcen hängt davon ab, wie die Verbindung eingerichtet wird (siehe Kapitel Einrichten einer Verbindung (Seite 31)).

- **Programmiertes Einrichten einer Verbindung:**

Sobald im Anwenderprogramm der CPU eine Anweisung zum Aufbau einer Verbindung (TSEND_C/TRCV_C oder TCON) aufgerufen wird, wird eine Verbindungsressource belegt.

Durch entsprechende Parametrierung des CONT Parameters der Anweisungen TSEND_C/TRCV_C bzw. den Aufruf der Anweisung TDISCON kann die Verbindung nach der Datenübertragung abgebaut und die Verbindungsressource freigegeben werden. Wenn die Verbindung abgebaut ist, stehen die Verbindungsressourcen in der CPU/CP/CM wieder zur Verfügung.

- **Konfigurierte Verbindungen** (z. B. HMI-Verbindung):

Wenn Sie eine Verbindung in STEP 7 konfiguriert haben, dann wird die Verbindungsressource belegt, sobald die Hardware-Konfiguration in die CPU geladen wurde.

Nach der Nutzung einer projektierten Verbindung zur Datenübertragung wird die Verbindung nicht abgebaut. Die Verbindungsressource bleibt dauerhaft belegt. Um die Verbindungsressource wieder frei zu geben, müssen Sie die projektierte Verbindung in STEP 7 löschen und die geänderte Projektierung in die CPU laden.

- **PG-Verbindung:**

Sobald Sie das PG mit einer CPU online in STEP 7 verbunden haben, werden Verbindungsressourcen belegt.

- **Webserver:**

Solange Sie den Webserver der CPU in einem Browser geöffnet haben, werden Verbindungsressourcen in der CPU belegt.

- **OPC UA-Server:**

Solange eine Session zwischen dem OPC UA Server der CPU und einem OPC UA Client besteht, wird eine Verbindungsressource in der CPU belegt.

Überwachung der maximal möglichen Anzahl Verbindungsressourcen

Offline

STEP 7 überwacht beim Konfigurieren von Verbindungen die Belegung von Verbindungsressourcen. Ein Überschreiten der maximal möglichen Anzahl von Verbindungsressourcen meldet STEP 7 mit einer entsprechenden Warnung.

Online

Die CPU überwacht den Verbrauch von Verbindungsressourcen im Automatisierungssystem. Wenn Sie im Anwenderprogramm mehr Verbindungen aufbauen, als das Automatisierungssystem Verbindungsressourcen bereitstellt, dann quittiert die CPU die Anweisung zum Aufbau der Verbindung mit einem Fehler.

11.3 Anzeige der Verbindungsressourcen

Anzeige der Verbindungsressourcen in STEP 7 (Offline-Sicht)

Sie können sich die Verbindungsressourcen eines Automatisierungssystems in der Hardware-Konfiguration anzeigen lassen. Sie finden die Verbindungsressourcen im Inspektorfenster in den Eigenschaften der CPU.


Verbindungsressourcen									
		① Ressourcen der Station			② Ressourcen ...				
		Reserviert		Dynamis.. 	PLC_1 [CPU ...	CP 1543-1_...	Ressourcen ...		
Maximale Anzahl der Ressourcen:		10		246	128	118	48		
		Maximum	Konfigurierte	Konfigurierte	Konfigurierte	Konfigurierte	Konfigurierte		
PG-Kommunikation:		4	-	-	-	-	-		
HMI-Kommunikation:		4	4	6	6	4	0		
S7-Kommunikation:		0	-	23	2	0	21		
Open User Communication:		0	-	52	39	13	0		
Web-Kommunikation:		2	-	-	-	-	-		
Sonstige Kommunikation:		-	-	0	0	0	0		
Insgesamt verwendete Ressourcen:		4		81	47	17	21		
Verfügbare Ressourcen:		6		165	81	101	27		

Bild 11-4 Beispiel: Reservierte und verfügbare Verbindungsressourcen (Offline-Sicht)

① Stationsspezifische Verbindungsressourcen

Die Spalten der stationsspezifischen Verbindungsressourcen geben Informationen über die verwendeten und verfügbaren Verbindungsressourcen der Station.

Im Beispiel stehen für das Automatisierungssystem maximal 256 stationsspezifische Verbindungsressourcen zur Verfügung:

- 10 reservierte Verbindungsressourcen, davon sind 4 bereits verwendet und 6 noch verfügbar.
Die verwendeten Ressourcen teilen sich wie folgt auf:
 - 4 Ressourcen für HMI-Kommunikation
- 246 dynamische Verbindungsressourcen, davon sind 81 bereits verwendet und 165 noch verfügbar.
Die verwendeten Ressourcen teilen sich wie folgt auf:
 - 6 Ressourcen für HMI-Kommunikation
 - 23 Ressourcen für S7-Kommunikation
 - 52 Ressourcen für Open User Communication

Das Warndreieck in der Spalte der dynamischen Stationsressourcen wird deshalb angezeigt, weil die Summe der max. verfügbaren Verbindungsressourcen von CPU, CP und CM (= 294 Verbindungsressourcen) die Stationsgrenze von 256 überschreitet.

Hinweis**Überschreiten der verfügbaren Verbindungsressourcen**

Ein Überschreiten der stationsspezifischen Verbindungsressourcen meldet STEP 7 mit einer Warnung. Wenn Sie die verfügbaren Verbindungsressourcen aus CPU, CP und CM voll ausnutzen wollen, dann müssen Sie entweder eine CPU mit einer größeren max. Anzahl verfügbarer stationsspezifischer Verbindungsressourcen einsetzen oder die Anzahl der Kommunikationsverbindungen reduzieren.

② Modulspezifische Verbindungsressourcen

Die Spalten der modulspezifischen Verbindungsressourcen geben Informationen über die Ressourcenverwendung in den CPUs, CPs und CMs eines Automatisierungssystems:

Die Anzeige ist modul- und nicht schnittstellengranular.

Im Beispiel stellt die CPU maximal 128 Verbindungsressourcen zur Verfügung, davon sind 47 bereits verwendet und 81 noch verfügbar:

Die verwendeten Ressourcen teilen sich wie folgt auf:

- 6 Ressourcen für HMI-Kommunikation
- 2 Ressourcen für S7-Kommunikation
- 39 Ressourcen für Open User Communication

Anzeige der Verbindungsressourcen in STEP 7 (Online-Sicht)

Wenn Sie online mit der CPU verbunden sind, dann können Sie sich unter "Verbindungsinformation" zusätzlich anzeigen lassen, wie viele Ressourcen jeweils aktuell verwendet werden.

PLC_2 [CPU 1516-3 PN/DP]								
Eigenschaften Info Diagnose								
Geräte-Information Verbindungsinformation Meldungsanzeige								
Verbindungsressourcen								
Maximale Anzahl der Ressourcen:	Ressourcen der Station						Ressourcen des Moduls	
	Reserviert			Dynamisch			PLC_2 [CPU 1516-3 PN/DP]	
	Maximum	Konfigurierte	Verwendet	Konfigurierte	Verwendet		Konfigurierte	Verwendet
PG-Kommunikation:	4	-	4	-	0		-	0
HMI-Kommunikation:	4	4	4	4	4		8	8
S7-Kommunikation:	0	0	0	72	68		34	34
Open User Communication:	0	0	0	118	118		45	45
Web-Kommunikation:	2	-	2	-	0		-	0
Sonstige Kommunikation:	-	-	0	-	0		-	0
Insgesamt verwendete Ressourc...		4	10	194	190		82	82
Verfügbare Ressourcen:		6	0	0	4		4	4

Bild 11-5 Verbindungsressourcen - online

Die Online-Sicht der Tabelle "Verbindungsressourcen" enthält zusätzlich zur Offline-Sicht Spalten mit den aktuell verwendeten Verbindungsressourcen. In der Online-Sicht werden **alle** verwendeten Verbindungsressourcen im Automatisierungssystem angezeigt, unabhängig davon, auf welche Art die Verbindung eingerichtet wurde.

In der Zeile "Sonstige Kommunikation" werden belegte Verbindungsressourcen für Kommunikation zu Fremdgeräten angezeigt. Die Tabelle wird automatisch aktualisiert.

Hinweis

Wenn eine geroutete S7-Verbindung über eine CPU geht, dann werden die dafür benötigten Verbindungsressourcen der CPU nicht in der Tabelle der Verbindungsressourcen angezeigt!

Anzeige der Verbindungsressourcen für HMI

Informationen über die Verfügbarkeit und Belegung von Verbindungsressourcen für HMI-Verbindungen finden Sie in der Offline-Sicht im Kontext des HMI-Geräts (im Inspektorfenster, in den Eigenschaften im Bereich "Verbindungsressourcen").

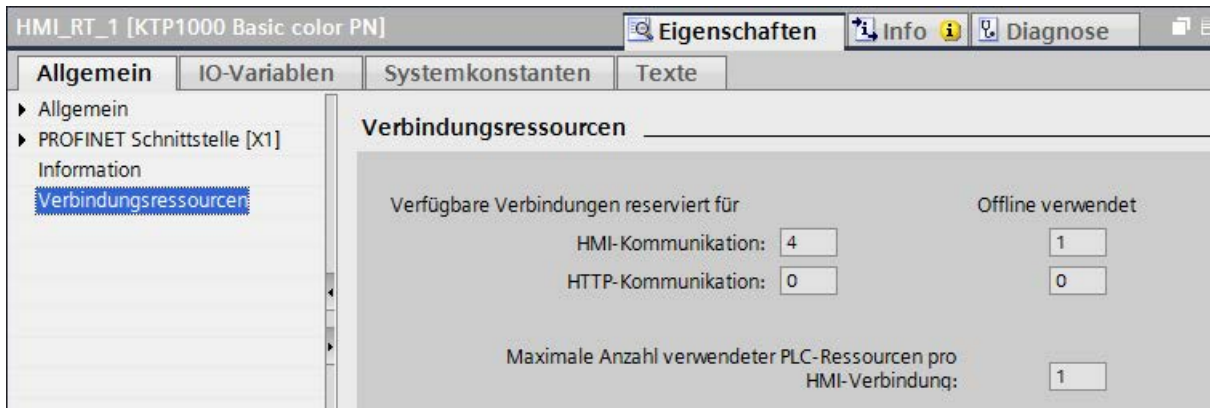


Bild 11-6 Verbindungsressourcen - HMI-Kommunikation

Im Bereich Verbindungsressourcen werden angezeigt:

- Anzahl der im HMI verfügbaren Verbindungen reserviert für HMI-Kommunikation und HTTP-Kommunikation
- Anzahl der offline im HMI verwendeten Verbindungsressourcen für HMI-Kommunikation und HTTP-Kommunikation

Falls die Anzahl der maximal verfügbaren Verbindungsressourcen für ein HMI-Gerät überschritten ist, wird eine entsprechende Meldung von STEP 7 ausgegeben.

- "Maximale Anzahl verwendeter PLC-Ressourcen pro HMI-Verbindung": Dieser Parameter ist ein Faktor, der mit der Anzahl der offline verwendeten HMI-Verbindungen zu multiplizieren ist. Das Produkt ergibt die Anzahl der in der CPU belegten HMI-Ressourcen.

Anzeige der Verbindungsressourcen im Webserver

Sie können sich die Verbindungsressourcen nicht nur in STEP 7 anzeigen lassen, sondern auch mit einem Browser, der die entsprechende Seite des Webserverns anzeigt.

Informationen zur Anzeige der Verbindungsressourcen im Webserver finden Sie im Funktionshandbuch Webserver

(<http://support.automation.siemens.com/WW/view/de/59193560>).

Diagnose von Verbindungen

Verbindungstabelle in der Online-Sicht

Für eine im Hardware- und Netzwerkeditor von STEP 7 angewählte CPU erhalten Sie in der Online-Sicht der Verbindungstabelle den Status ihrer Verbindungen angezeigt.

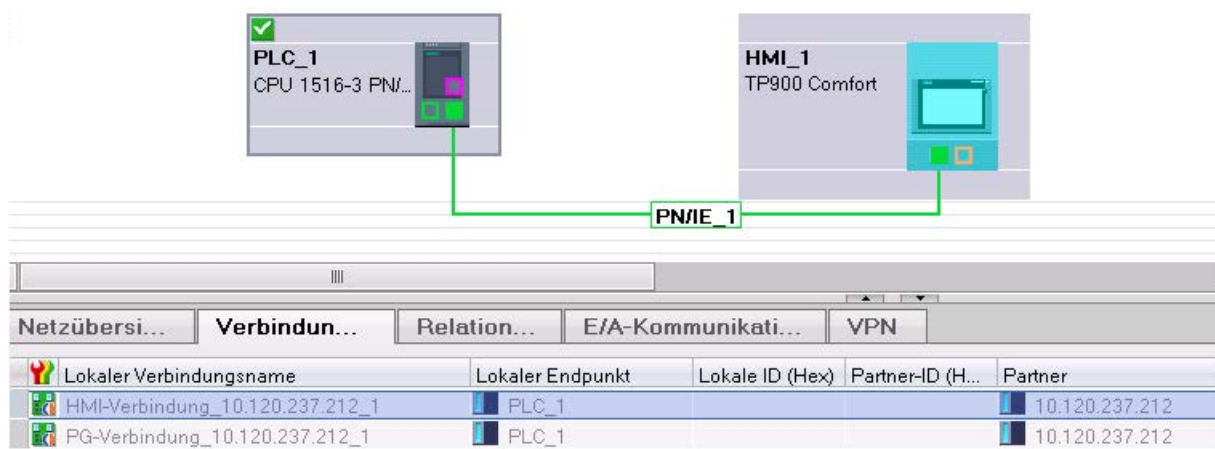


Bild 12-1 Online-Sicht der Verbindungstabelle

Für die angewählte Verbindung in der Verbindungstabelle erhalten Sie detaillierte Diagnoseinformationen im Register "Verbindungsinformationen".

Register "Verbindungsinformationen": Verbindungsdetails

The screenshot shows the 'Verbindungsinformationen' (Connection Information) window with the 'Verbindungsdetails' (Connection Details) tab selected. The window displays a table of connections and a detailed view of the selected connection.

Lokaler Verbindungsname	Lokaler Endpunkt	Lokale ID (Hex)	Partner-ID (H...	Partner	Verbindungstyp
HMI-Verbindung_10.120.237.212_1	PLC_1			10.120.237.212	HMI-Verbindung
PG-Verbindung_10.120.237.212_1	PLC_1			10.120.237.212	PG-Verbindung

The 'Verbindungsdetails' (Connection Details) section shows the following information:

- Verbindungsname: HMI-Verbindung_10.120.237.212_1 (ConnEnd_187)
- Lokale ID (hex):
- Verbindungstyp: Verbindung von HMI-Client
- Protokoll: ISO-on-TCP
- Online-Status: Verbunden
- Details: Aufgebaut: Verbindung existiert nur Online. Verbindung ist aufgebaut.

Bild 12-2 Diagnose von Verbindungen - Verbindungsdetails

Register "Verbindungsinformationen": Adressdetails

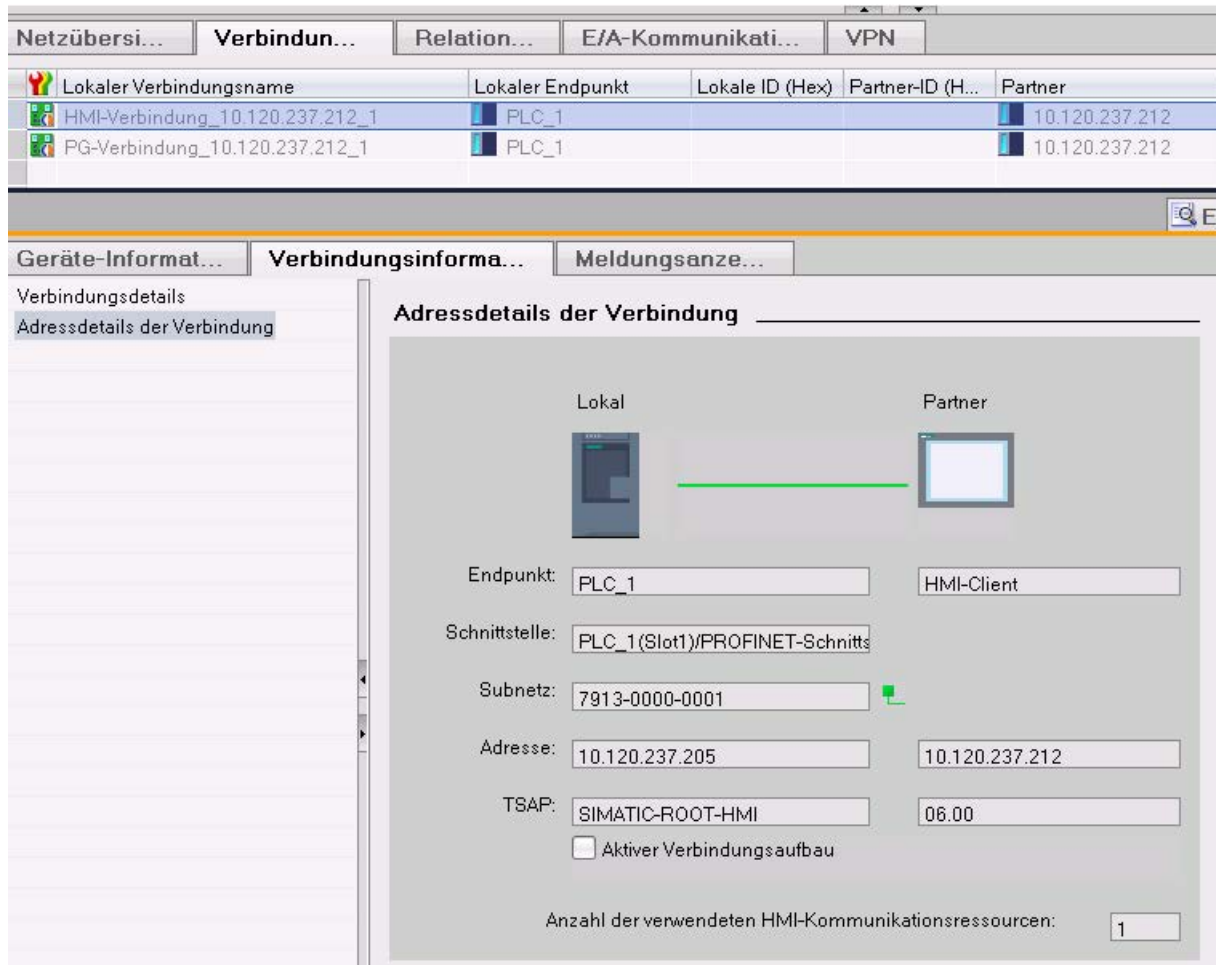


Bild 12-3 Diagnose von Verbindungen - Adressdetails

Diagnose über Webserver

Über den integrierten Webserver einer CPU haben Sie die Möglichkeit, Diagnoseinformationen von der CPU über einen Webbrowser auszuwerten.

Auf der Webseite "Kommunikation" finden Sie in verschiedenen Registern folgende Informationen zur Kommunikation über PROFINET:

- Informationen zu den PROFINET-Schnittstellen der CPU (z. B. Adressen, Subnetze, physikalische Eigenschaften)
- Informationen zur Qualität der Datenübertragung (z. B. Anzahl fehlerfrei gesendeter/empfangener Datenpakete)
- Informationen zur Belegung/Verfügbarkeit von Verbindungsressourcen
- Die Seite "Verbindungen" ist ähnlich der Online-Sicht in STEP 7 und gibt auch eine Übersicht aller Verbindungen mit Detailansicht

Diagnose über Anwenderprogramm

Wenn Sie die Anweisung T_DIAG programmieren, können Sie über das Anwenderprogramm Diagnoseinformationen über die projektierten und programmierten Verbindungen von der CPU auswerten.

Weitere Informationen

Die Beschreibung der Webserverfunktionalität finden Sie im Funktionshandbuch Webserver (<http://support.automation.siemens.com/WW/view/de/59193560>).

Industrial Ethernet Security mit CP 1543-1

Umfassender Schutz - Aufgabe von Industrial Ethernet Security

Mit Industrial Ethernet Security können einzelne Geräte, Automatisierungszellen oder Netzsegmente eines Ethernet-Netzwerks abgesichert werden. Zusätzlich kann die Datenübertragung durch die Kombination unterschiedlicher Sicherheitsmaßnahmen geschützt werden vor:

- Datenspionage
- Datenmanipulation
- unberechtigten Zugriffen

Sicherheitsmaßnahmen

- Firewall
 - IP-Firewall mit Stateful Packet Inspection (Layer 3 und 4)
 - Firewall auch für Ethernet-"Non-IP"-Telegramme gemäß IEEE 802.3 (Layer2)
 - Bandbreitenbegrenzung
 - Globale Firewall-Regeln

Alle Netzknoten, die sich im internen Netzsegment eines CP 1543-1 befinden, werden durch dessen Firewall geschützt.

- Logging

Zur Überwachung lassen sich Ereignisse in Log-Dateien speichern, die mit Hilfe des Projektierwerkzeugs ausgelesen werden oder automatisch an einen Syslog-Server gesendet werden können.
- HTTPS

Zur verschlüsselten Übertragung von Webseiten, z. B. bei der Prozesskontrolle.
- FTPS (expliziter Modus)

Zur verschlüsselten Übertragung von Dateien.
- Gesichertes NTP

Zur sicheren Uhrzeitsynchronisierung und -übertragung.
- SNMPv3

Zur abhörsicheren Übertragung von Netzwerkanalyseinformationen.

- **VPN-Gruppen**
Den CP 1543-1 können Sie mit anderen Security-Baugruppen per Projektierung zu VPN-Gruppen zusammenfassen. Zwischen allen Security-Baugruppen einer VPN-Gruppe werden IPsec-Tunnel aufgebaut (VPN). Alle internen Knoten dieser Security-Baugruppen können mittels dieser Tunnel gesichert miteinander kommunizieren.
- **Schutz für Geräte und Netzsegmente**
Die Schutzfunktionen Firewall und VPN-Gruppen kann sich über den Betrieb einzelner Geräte, mehrerer Geräte wie auch ganzer Netzsegmente erstrecken.

Weitere Informationen

Eine Übersicht mit Links zu den wichtigsten Beiträgen zu Industrial Security finden Sie im diesem FAQ (<https://support.industry.siemens.com/cs/ww/de/view/92651441>).

13.1 Firewall

Aufgaben der Firewall

Die Firewall-Funktionalität hat die Aufgabe, Netze und Stationen vor Fremdbeeinflussungen und Störungen zu schützen. Das bedeutet, dass nur bestimmte, vorher festgelegte Kommunikationsbeziehungen erlaubt werden.

Zur Filterung des Datenverkehrs können u. a. IPv4-Adressen, IPv4-Subnetze, Portnummern oder MAC-Adressen verwendet werden.

Die Firewall-Funktionalität kann für folgende Protokollebenen konfiguriert werden:

- IP-Firewall mit Stateful Packet Inspection (Layer 3 und 4)
- Firewall auch für Ethernet-"Non-IP"-Telegramme gemäß IEEE 802.3 (Layer 2)

Firewall-Regeln

Firewall-Regeln beschreiben, welche Pakete in welche Richtung erlaubt bzw. verboten werden.

13.2 Logging

Funktionalität

Zu Test- und Überwachungszwecken verfügt das Security-Modul über Diagnose- und Logging-Funktionen.

- Diagnosefunktionen

Hierunter sind verschiedene System- und Statusfunktionen zu verstehen, die Sie im Online-Modus anwenden können.

- Logging-Funktionen

Hierbei geht es um die Aufzeichnung von System- und Sicherheitsereignissen. Die Aufzeichnung erfolgt je nach Ereignistyp in flüchtige oder dauerhafte lokale Pufferbereiche des CP 1543-1. Alternativ kann auch eine Aufzeichnung in einem Netzwerk-Server erfolgen.

Die Parametrierung und Auswertung dieser Funktionen setzt eine Netzwerkverbindung voraus.

Ereignisse mit Logging-Funktionen aufzeichnen

Welche Ereignisse aufgezeichnet werden sollen, legen Sie mit den Log-Einstellungen fest. Dabei können Sie für die Aufzeichnung folgende Varianten konfigurieren:

- Lokales Logging

Bei dieser Variante zeichnen Sie die Ereignisse in lokalen Puffern des CP 1543-1 auf. Im Online-Dialog des Security Configuration Tool können Sie dann auf diese Aufzeichnungen zugreifen, diese sichtbar machen und in der Service-Station archivieren.

- Netzwerk Syslog

Beim Netzwerk Syslog nutzen Sie einen im Netz vorhandenen Syslog-Server. Dieser zeichnet die Ereignisse entsprechend der Konfiguration in den Log-Einstellungen auf.

13.3 NTP-Client

Funktionalität

Zur Überprüfung der zeitlichen Gültigkeit eines Zertifikats und für die Zeitstempel von Log-Einträgen werden auf dem CP 1543-1, genauso wie auf der CPU, Datum und Uhrzeit geführt. Diese Uhrzeit ist per NTP synchronisierbar. Der CP 1543-1 leitet die synchronisierte Uhrzeit über den Rückwandbus des Automatisierungssystems an die CPU weiter. So erhält auch die CPU eine synchronisierte Uhrzeit für die Zeitereignisse in der Anwenderprogrammbearbeitung.

Das automatische Stellen und der periodische Abgleich der Uhrzeit wird über einen gesicherten oder ungesicherten NTP-Server realisiert. Sie können dem CP 1543-1 max. 4 NTP-Server zuweisen. Eine gemischte Konfiguration von ungesicherten und gesicherten NTP-Servern ist nicht möglich.

13.4 SNMP

Funktionalität

Der CP 1543-1 unterstützt, genauso wie die CPU, die Übertragung von Managementinformationen über das Simple Network Management Protocol (SNMP). Dafür ist auf dem CP/der CPU ein "SNMP-Agent" installiert, der die SNMP-Anfragen entgegennimmt und beantwortet. Informationen über die Eigenschaften von SNMP-fähigen Geräten sind in sogenannten MIB-Dateien (Management Information Base) hinterlegt, für die der Benutzer die notwendigen Rechte haben muss.

Beim SNMPv1 wird der "Community String" mitgesendet. Der "Community String" ist wie ein Passwort, das zusammen mit der SNMP-Anfrage verschickt wird. Wenn der Community String korrekt ist, wird die angeforderte Information gesendet. Wenn der String falsch ist, wird die Anfrage verworfen.

Bei SNMPv3 können die Daten verschlüsselt übertragen werden. Dazu wählen Sie entweder ein Authentifizierungsverfahren oder ein Authentifizierungs- und Verschlüsselungsverfahren aus.

Mögliche Auswahl:

- Authentifizierungs-Algorithmus: keine, MD5, SHA-1
- Verschlüsselungs-Algorithmus: keine, AES-128, DES

Sie können die Verwendung von SNMP für den CP/die CPU deaktivieren. Deaktivieren Sie SNMP, wenn die Sicherheitsrichtlinien in Ihrem Netzwerk kein SNMP zulassen oder Sie eine eigene SNMP-Lösung verwenden.

Wie Sie SNMP für die CPU deaktivieren, finden Sie im Kapitel SNMP deaktivieren (Seite 61).

13.5 VPN

Funktionalität

Für Security-Baugruppen, die das interne Netz schützen, stellen VPN-Tunnel (Virtual Private Network) eine gesicherte Datenverbindung durch das unsichere externe Netz zur Verfügung.

Die Baugruppe verwendet für die Tunnelung das IPsec-Protokoll (Tunnelmodus von IPsec).

In STEP 7 können Sie Security-Baugruppen VPN-Gruppen zuordnen. Zwischen allen Baugruppen einer VPN-Gruppe werden automatisch VPN-Tunnel aufgebaut. Dabei kann eine Baugruppe in einem Projekt parallel mehreren verschiedenen VPN-Gruppen angehören.

Glossar

Anweisung

Die kleinste selbstständige Einheit eines Anwenderprogramms, durch Struktur, Funktion oder Verwendungszweck als abgegrenzter Teil des Anwenderprogramms charakterisiert. Die Anweisung stellt eine Arbeitsvorschrift für den Prozessor dar.

Anwenderprogramm

Bei SIMATIC wird zwischen dem Betriebssystem der CPU und Anwenderprogrammen unterschieden. Das Anwenderprogramm enthält alle Anweisungen, Deklarationen und Daten, durch die eine Anlage oder ein Prozess gesteuert werden können. Das Anwenderprogramm ist einem programmierbaren Modul (z. B. CPU, FM) zugeordnet und ist in kleinere Einheiten strukturierbar.

Automatisierungssystem

Speicherprogrammierbare Steuerung für die Regelung und Steuerung von Prozessketten der verfahrenstechnischen Industrie und der Fertigungstechnik. Je nach Automatisierungsaufgabe setzt sich das Automatisierungssystem aus unterschiedlichen Komponenten und integrierten Systemfunktionen zusammen.

Baumtopologie

Netzwerktopologie, gekennzeichnet von einer verzweigten Struktur: An jeden Busteilnehmer werden zwei oder weitere Busteilnehmer angeschlossen.

Betriebssystem

Software, die die Verwendung und den Betrieb eines Computers ermöglicht. Das Betriebssystem verwaltet Betriebsmittel wie Speicher, Ein- und Ausgabegeräte und steuert die Ausführung von Programmen.

Bus

Übertragungsmedium, das mehrere Teilnehmer miteinander verbindet. Die Datenübertragung kann elektrisch oder über Lichtwellenleiter sowohl seriell als auch parallel erfolgen.

Client

Teilnehmer in einem Netz, der von einem anderen Teilnehmer am Netz (Server) einen Dienst anfordert.

CM

→ *Kommunikationsmodul*

CP

→ *Kommunikationsprozessor*

CPU

Central Processing Unit - Zentralbaugruppe des S7-Automatisierungssystems mit Steuer- und Rechenwerk, Speicher, Betriebssystem und Schnittstelle für Programmiergerät.

DP-Master

Innerhalb von PROFIBUS DP ein Master in der Dezentralen Peripherie, der sich nach der Norm EN 50170, Teil 3, verhält.

→ *Siehe auch DP-Slave*

DP-Slave

Slave in der Dezentralen Peripherie, der am PROFIBUS mit dem Protokoll PROFIBUS DP betrieben wird und sich nach der Norm EN 50170, Teil 3, verhält.

→ *Siehe auch DP-Master*

Duplex

Datenübertragungsverfahren; es wird zwischen Voll- und Halbduplexverfahren unterschieden.

Halbduplex: Ein Kanal zum abwechselnden Datenaustausch steht zur Verfügung (abwechselnd Senden und Empfangen in jeweils eine Richtung).

Vollduplex: Zwei Kanäle zum gleichzeitigen Datenaustausch in beide Richtungen stehen zur Verfügung (gleichzeitiges Senden und Empfangen in beide Richtungen).

End-Entity-Zertifikat

→ *Siehe auch Gerätezertifikat*

Ethernet

Internationale Standardtechnologie für lokale Netzwerke (LAN), basierend auf Frames. Sie definiert Kabeltypen und Signalisierung für die Bitübertragungsschicht sowie Paketformate und Protokolle für die Medienzugriffskontrolle.

Ethernet-Netzwerkkarte

Elektronische Schaltung zur Verbindung eines Computers mit einem Ethernet-Netzwerk. Sie ermöglicht den Austausch von Daten/die Kommunikation innerhalb des Netzes.

Feldgerät

→ *Gerät*

FETCH/WRITE

Server-Dienste über TCP/IP, ISO-on-TCP und ISO für den Zugriff auf Systemspeicherbereiche von S7-CPU's. Der Zugriff (Client-Funktion) ist von einer SIMATIC S5 oder einem Fremdgerät/PC aus möglich. FETCH: Daten direkt lesen; WRITE: Daten direkt schreiben.

Freeport

Frei programmierbares ASCII-Protokoll; hier zur Datenübertragung über Punkt-zu-Punkt-Kopplung.

FTP

File Transfer Protocol; ein Netzwerkprotokoll zur Übertragung von Dateien über IP-Netzwerke. FTP wird benutzt, um Dateien vom Server zum Client herunter zu laden oder vom Client zum Server hochzuladen. Außerdem können über FTP Verzeichnisse angelegt und ausgelesen, sowie Verzeichnisse und Dateien umbenannt oder gelöscht werden.

Gerät

Oberbegriff für:

- Automatisierungssysteme (z. B. SPS, PC)
- Dezentrale Peripheriesysteme
- Feldgeräte (z. B. SPS, PC, Hydraulikgeräte, Pneumatikgeräte) und
- Aktive Netzkomponenten (z. B. Switches, Router)
- Netzübergänge zu PROFIBUS, AS-Interface oder anderen Feldbussystemen

Gerätezertifikate

Solche Zertifikate werden von einer Zertifizierungsstelle (CA) signiert.

Die Signatur eines End-Entity-Zertifikats wird mit dem öffentlichen Schlüssel des Zertifikats der Zertifizierungsstelle überprüft.

Die Attribute "Antragsteller" und "Aussteller" dürfen nicht identisch sein.

Bei "Antragsteller" steht zum Beispiel der Name eines Programms, wie beim OPC UA Applikations-Zertifikat.

Bei "Aussteller" steht die Zertifizierungsstelle, die dieses Zertifikat signiert hat.

Das Feld "CA" muss auf "False" stehen.

HMI

Human Maschine Interface, Gerät zur Visualisierung und Steuerung von Automatisierungsprozessen.

IE

→ *Industrial Ethernet*

IM

→ *Interfacemodul*

Industrial Ethernet

Richtlinie zum Aufbau eines Ethernets in industrieller Umgebung. Der wesentliche Unterschied zum Standard-Ethernet liegt in der mechanischen Belastbarkeit und Störunempfindlichkeit der einzelnen Komponenten.

Interfacemodul

Modul im Dezentralen Peripheriesystem. Das Interfacemodul verbindet das Dezentrale Peripheriesystem über einen Feldbus mit der CPU (IO-Controller/DP-Master) und bereitet die Daten für die Peripheriemodule auf.

Intermediate CA-Zertifikat

Das ist das Zertifikat einer Zertifizierungsstelle, das mit dem privaten Schlüssel einer Root-Zertifizierungsstelle signiert ist.

Eine solche zwischengeschaltete Zertifizierungsstelle signiert mit ihrem privaten Schlüssel End-Entity-Zertifikate.

Die Signatur dieser End-Entity-Zertifikate wird mit dem öffentlichen Schlüssel der Intermediate-Zertifizierungsstelle überprüft.

Die Attribute "Antragsteller" und "Aussteller" des Intermediate CA-Zertifikats dürfen nicht identisch sein: Diese Zertifizierungsstelle hat ihr Zertifikat ja nicht selbst signiert.

Das Feld "CA" muss auf "True" stehen.

IO-Controller, PROFINET IO-Controller

Zentrales Gerät in einem PROFINET-System, meist eine klassische speicher-programmierbare Steuerung oder ein PC. Der IO-Controller richtet Verbindungen zu den IO-Devices ein, tauscht Daten mit ihnen aus, steuert und kontrolliert also das System.

IO-Device, PROFINET IO-Device

Gerät der Dezentralen Peripherie eines PROFINET-Systems, das von einem IO-Controller kontrolliert und gesteuert wird (z. B. dezentrale Ein-/Ausgänge, Ventilinseln, Frequenzumrichter, Switches).

IP-Adresse

Binäre Zahl, die im Zusammenhang mit dem Internetprotokoll (IP) als eindeutige Adresse in Computernetzwerken verwendet wird. Dadurch werden diese Geräte eindeutig adressierbar und individuell erreichbar. Eine IPv4-Adresse kann mit Hilfe einer binären Subnetz-Maske so bewertet werden, dass sich ein Netzanteil bzw. ein Hostanteil als Struktur ergibt. Die textuelle Darstellung einer IPv4-Adresse besteht beispielsweise aus 4 Dezimalzahlen mit dem Wertebereich 0 bis 255. Die Dezimalzahlen sind durch einen Punkt voneinander getrennt.

IPv4-Subnetzmaske

Binäre Maske, mit der eine IPv4-Adresse (als binäre Zahl) in einen "Netzanteil" und einen "Hostanteil" aufteilt.

ISO-on-TCP-Protokoll

S7-Routing-fähiges Kommunikationsprotokoll für paketorientierte Übertragung von Daten im Ethernet; bietet eine Netzwerkadressierung. ISO-on-TCP-Protokoll ist für mittlere und große Datenmengen geeignet und erlaubt dynamische Datenlängen.

ISO-Protokoll

Kommunikationsprotokoll für nachrichten- bzw. paketorientierte Übertragung von Daten im Ethernet. Dieses Protokoll ist hardwarenahe, sehr schnell und lässt dynamische Datenlängen zu. Das ISO-Protokoll eignet sich für mittlere und große Datenmengen.

Kommunikationsmodul

Baugruppe für Kommunikationsaufgaben, die in einem Automatisierungssystem als Schnittstellenerweiterung der CPU (z. B. PROFIBUS) verwendet wird bzw. zusätzliche Kommunikationsmöglichkeiten (PtP) bietet.

Kommunikationsprozessor

Baugruppe für erweiterte Kommunikationsaufgaben, die spezielle Anwendungsfälle, z. B. im Bereich Security, abdeckt.

Konsistente Daten

Daten, die inhaltlich zusammengehören und beim Übertragen nicht getrennt werden dürfen.

Linientopologie

Netzwerktopologie, gekennzeichnet durch die Anordnung der Busteilnehmer in einer Reihe.

MAC-Adresse

Weltweit eindeutige Geräte-Identifikation für alle Ethernet-Geräte. Die MAC-Adresse wird bereits vom Hersteller vergeben und hat 3 byte Herstellerkennung und 3 byte Geräte-kennung als laufende Nummer.

Master

Übergeordneter, aktiver Teilnehmer an der Kommunikation/am PROFIBUS-Subnetz. Besitz der Master besitzt Buszugriffsrechte (Token), kann der Daten anfordern und verschicken.

→ *Siehe auch DP-Master*

Modbus RTU

Remote Terminal Unit; Offenes Kommunikationsprotokoll für serielle Schnittstellen, welches auf einer Master-/Slave-Architektur basiert.

Modbus TCP

Transmission Control Protokoll; Offenes Kommunikationsprotokoll für Ethernet, welches auf einer Master/Slave-Architektur basiert. Die Daten werden als TCP/IP-Pakete übertragen.

Netz

Ein Netz besteht aus einem oder mehreren verknüpften Subnetzen mit einer beliebigen Zahl von Teilnehmern. Mehrere Netze können nebeneinander bestehen.

NTP

Das **Network Time Protocol** (NTP) ist ein Standard zur Synchronisierung von Uhren in Automatisierungssystemen über Industrial Ethernet. NTP verwendet das verbindungslose Transportprotokoll UDP für das Internet.

OPC UA

OPC Unified Automation ist ein Protokoll für die Kommunikation zwischen Maschinen, entwickelt von der OPC Foundation.

PG

→ *Programmiergerät*

PNO

→ *PROFIBUS-Nutzerorganisation*

Port

Physikalische Anschlussmöglichkeit für Geräte, die PROFINET-Teilnehmer sind. PROFINET-Schnittstellen verfügen über einen oder mehrere Ports.

PROFIBUS

Process Field Bus - europäische Feldbusnorm.

PROFIBUS DP

Ein PROFIBUS mit dem Protokoll DP, der sich konform zur EN 50170 verhält. DP steht für Dezentrale Peripherie (schnell, echtzeitfähig, zyklischer Datenaustausch). Aus Sicht des Anwenderprogramms wird die dezentrale Peripherie genauso angesprochen wie die zentrale Peripherie.

PROFIBUS-Adresse

Eindeutige Kennung eines am PROFIBUS angeschlossenen Teilnehmers. Zur Adressierung eines Teilnehmers wird die PROFIBUS-Adresse im Telegramm übertragen.

PROFIBUS-Gerät

Gerät mit mindestens einer PROFIBUS-Schnittstelle, entweder elektrisch (z. B. RS485) oder optisch (z. B. Polymer Optical Fiber).

PROFIBUS-Nutzerorganisation

Technisches Komitee, das den PROFIBUS- und PROFINET-Standard definiert und weiterentwickelt.

PROFINET

Offenes komponentenbasiertes industrielles Kommunikationssystem auf Ethernet-Basis für verteilte Automatisierungssysteme. Von der PROFIBUS-Nutzerorganisation geförderte Kommunikationstechnologie.

PROFINET IO

IO steht für Input/Output; Dezentrale Peripherie (schnell, echtzeitfähig, zyklischer Datenaustausch). Aus Sicht des Anwenderprogramms wird die dezentrale Peripherie genauso angesprochen wie die zentrale Peripherie.

PROFINET IO als Ethernet-basierter Automatisierungsstandard von PROFIBUS & PROFINET International definiert damit ein herstellerübergreifendes Kommunikations-, Automatisierungs- und Engineering-Modell.

Bei PROFINET IO wird eine Switching-Technologie eingesetzt, die es jedem Teilnehmer ermöglicht, zu jedem Zeitpunkt auf das Netz zuzugreifen. Damit kann das Netz durch gleichzeitige Datenübertragung mehrerer Teilnehmer wesentlich effektiver genutzt werden. Gleichzeitiges Senden und Empfangen wird durch den Vollduplex-Betrieb von Switched-Ethernet ermöglicht.

PROFINET IO basiert auf Switched-Ethernet mit Vollduplex-Betrieb und einer Übertragungsbandbreite von 100 Mbit/s.

PROFINET-Gerät

Gerät, das immer über eine PROFINET-Schnittstelle (elektrisch, optisch, drahtlos) verfügt.

PROFINET-Schnittstelle

Schnittstelle eines kommunikationsfähigen Moduls (z. B. CPU, CP) mit einem oder mehreren Ports. Bereits ab Werk ist der Schnittstelle eine MAC-Adresse zugewiesen. Zusammen mit der IP-Adresse und dem Gerätenamen (aus der individuellen Konfiguration) gewährleistet diese Adresse der Schnittstelle eine eindeutige Identifizierung des PROFINET-Geräts im Netz. Die Schnittstelle kann elektrisch, optisch oder drahtlos sein.

Programmiergerät

Programmiergeräte sind im Kern Personal Computer, die industrietauglich, kompakt und transportabel sind. Sie sind gekennzeichnet durch eine spezielle Hardware- und Software-Ausstattung für speicherprogrammierbare Steuerungen.

Protokoll

Vereinbarung, nach welchen Regeln die Kommunikation zwischen zwei oder mehreren Kommunikationspartnern abläuft.

Prozessabbild

Adressbereich einer speicherprogrammierbaren Steuerung (SPS), in dem die Signalzustände der Eingänge und die logischen Zustände der Ausgänge aus den angeschlossenen Modulen digital abgelegt sind.

PtP

Point-to-Point, Schnittstelle und/oder Übertragungsprotokoll für bidirektionalen Datenaustausch zwischen genau zwei Kommunikationspartnern.

Punkt-zu-Punkt-Kopplung

Bidirektionaler Datenaustausch über Kommunikationsmodule mit serieller Schnittstelle zwischen genau zwei Kommunikationspartnern.

Ringtopologie

Alle Teilnehmer eines Netzes sind in einem Ring zusammengeschlossen.

Root-CA-Zertifikate

→ *Siehe auch Stammzertifikat*

Router

Netzwerkknoten mit eindeutiger Kennzeichnung (Namen und Adresse), der Subnetze miteinander verbindet und den Datentransport zu eindeutig gekennzeichneten Kommunikationsteilnehmern im Netz leistet.

RS232, RS422 und RS485

Standard für serielle Schnittstellen.

RTU

Modbus RTU (RTU: **R**emote **T**erminal **U**nit, entfernte Terminaleinheit) überträgt die Daten in binärer Form; ermöglicht einen guten Datendurchsatz. Die Daten müssen in ein lesbares Format umgesetzt werden, bevor sie ausgewertet werden können.

S7-Routing

Kommunikation zwischen S7-Automatisierungssystemen, S7-Anwendungen oder PC-Stationen in verschiedenen S7-Subnetzen über einen oder mehrere Netzwerkknoten, die als S7-Router fungieren.

SDA-Dienst

Send Data with Acknowledge. SDA ist ein elementarer Dienst, mit dem ein Initiator (z. B. DP-Master) eine Nachricht an einen anderen Teilnehmer abschicken kann und dafür unmittelbar eine Empfangsbestätigung erhält.

SDN-Dienst

Send Data with No Acknowledge. Dieser Dienst wird vorwiegend dafür eingesetzt, an mehrere Stationen Daten zu verschicken, der Dienst bleibt deswegen unquittiert. Geeignet für Synchronisationsaufgaben und Zustandsmeldungen.

Security

Oberbegriff für alle Maßnahmen zum Schutz vor

- Verlust der Vertraulichkeit durch unberechtigten Zugriff auf Daten
- Verlust der Integrität durch Manipulation von Daten
- Verlust der Verfügbarkeit durch Zerstörung von Daten

Selbst-signierte Zertifikate

Das sind Zertifikate, die Sie mit Ihrem privaten Schlüssel signieren und als End-Entität-Zertifikate verwenden.

Die Signatur dieser Zertifikate wird mit Ihrem öffentlichen Schlüssel überprüft.

Die Attribute "Antragsteller" und "Aussteller" von selbst-signierten Zertifikaten müssen identisch sein: Sie haben Ihr Zertifikat ja selbst signiert.

Das Feld "CA" muss auf "False" stehen.

Sie können selbst-signierte Zertifikate zum Beispiel als Applikations-Zertifikat für einen OPC UA-Client verwenden.

Wie Sie ein selbst-signiertes Zertifikat mit dem Zertifikate-Generator der OPC Foundation erzeugen, ist hier (Seite) beschrieben.

Server

Gerät oder allgemein ein Objekt, das bestimmte Dienste erbringen kann; aufgrund der Anforderung durch einen Client wird der Dienst erbracht.

Slave

Dezentrales Gerät in einem Feldbussystem, das nur nach Aufforderung durch einen Master Daten mit diesem austauschen darf.

→ *Siehe auch DP-Slave*

SNMP

Simple Network Management Protocol nutzt das verbindungslose Transportprotokoll UDP. SNMP funktioniert ähnlich dem Client/Server-Modell. Der SNMP Manager überwacht die Netzwerkknoten. Die SNMP Agenten sammeln in den einzelnen Netzwerkknoten verschiedene netzwerkspezifische Informationen und machen diese Informationen in strukturierter Form in der MIB (**M**anagement **I**nformation **B**ase) abrufbar und steuerbar. Mit Hilfe dieser Informationen kann ein Netzwerkmanagementsystem eine ausführliche Netzwerkd Diagnose durchführen.

Stammzertifikat

Das ist das Zertifikat einer Zertifizierungsstelle: Sie signiert mit ihrem privaten Schlüssel End-Entity-Zertifikate und Intermediate CA-Zertifikate.

Die Attribute "Antragsteller" und "Aussteller" dieses Zertifikats müssen identisch sein: Diese Zertifizierungsstelle hat ihr Zertifikat selbst signiert.

Das Feld "CA" muss auf "True" stehen.

Das TIA Portal V14 besitzt ein solches Root-CA-Zertifikat:

Wenn Sie im TIA Portal den OPC UA Server einer S7-1500 konfigurieren, dann erzeugt das TIA Portal für den OPC UA-Server ein End-Entity-Zertifikat und signiert dieses Zertifikat mit dem eigenen privaten Schlüssel.

Die Signatur dieses End-Entity-Zertifikats ist überprüfbar mit dem öffentlichen Schlüssel des TIA Portals. Dieser Schlüssel ist im Root-CA-Zertifikat des TIA Portals enthalten.

Subnetz

Teil eines Netzes, dessen Parameter bei den Teilnehmern (z. B. bei PROFINET) abgeglichen sein müssen. Ein Subnetz umfasst die Buskomponenten und alle angeschlossenen Stationen. Subnetze können beispielsweise mittels Gateways oder Routern zu einem Netz gekoppelt werden.

Switch

Netzwerk-Komponente zur Verbindung mehrerer Endgeräte bzw. Netz-Segmente in einem lokalen Netz (LAN).

TCP/IP

Transmission Control Protocol/Internet Protocol, verbindungsorientiertes Netzwerk Protokoll, allgemein anerkannter Standard für den Datenaustausch in heterogenen Netzen.

Twisted Pair

Fast Ethernet über Twisted Pair-Leitungen basiert auf dem Standard IEEE 802.3u (100 Base-TX). Übertragungsmedium ist eine 2x2-adrige, verdrehte und geschirmte Leitung mit einem Wellenwiderstand von 100 Ohm (AWG 22). Die Übertragungseigenschaften dieser Leitung müssen die Anforderungen der Kategorie 5 erfüllen.

Die Maximallänge der Verbindung zwischen Endgerät und Netzkomponente darf 100 m nicht überschreiten. Die Anschlüsse erfolgen nach 100 Base-TX-Standard mit dem RJ45-Steckverbindingssystem.

UDP

User-Datagram-Protokoll; Kommunikationsprotokoll zur schnellen und unkomplizierten Datenübertragung, ohne Quittierung. Auf Sicherungsmechanismen, wie sie bei TCP/IP vorhanden sind, wird verzichtet.

Uhrzeitsynchronisation

Fähigkeit zur Übertragung einer Standardsystemzeit von einer einzelnen Quelle an alle Geräte im System, so dass deren Uhren entsprechend der Standardzeit eingestellt werden können.

USS

Universelles serielles Schnittstellen-Protokoll; definiert ein Zugriffsverfahren nach dem Master-Slave-Prinzip für die Kommunikation über einen seriellen Bus.

Webserver

Software/Kommunikationsdienst zum Datenaustausch über das Internet. Der Webserver überträgt die Dokumente über standardisierte Übertragungsprotokolle (HTTP, HTTPS) an einen Webbrowser. Dokumente können statisch sein oder dynamisch bei Anforderung durch den Webbrowser aus unterschiedlichen Quellen durch den Webserver zusammengesetzt werden.

Index

A

Advanced Encryption Algorithm, 41
AES, 41
Antragsteller, 44
Asymmetrische Verschlüsselung, 42
Auf- und Abbau einer Kommunikation, 97

B

Belegung von Verbindungsressourcen, 249
BRCV, 122
BSEND, 122

C

CM, 16
CP, 16

D

Datenkonsistenz, 35
Datensatz-Routing, 240
Diagnose von Verbindung, 254
Digitale Zertifikate, 44

E

E-Mail, 21, 72, 93
End-Entity-Zertifikat, 47
Exportdatei für OPC UA, 172

F

FDL, 72
Fetch, 21
Firewall, 260
Freeport-Protokoll, 130
FTP, 21, 72, 93, 94

G

GET, 122

H

Handshake Protocol, 43
HMI-Kommunikation, 21, 68

I

IM, 20
Industrial Ethernet Security, 258
Interfacemodul, 20
ISO, 21, 71
ISO-on-TCP, 71, 80

K

Kommunikation
 Auf- und Abbau, 97
 Datensatz-Routing, 240
 HMI-Kommunikation, 68
 Kommunikationsprotokolle, 71
 Offene Kommunikation, 70
 Open User Communication, 70
 PG-Kommunikation, 65
 Punkt-zu-Punkt-Kopplung, 130
 S7-Kommunikation, 121
 S7-Routing, 235
Kommunikation über PUT/GET-Anweisung
 Verbindung anlegen und parametrieren, 123
Kommunikationsdienste
 Verbindungsressourcen, 30
Kommunikationsmodul, 16
Kommunikationsmöglichkeiten
 Überblick, 21
Kommunikationsprozessor, 16
Konsistenz von Daten, 35

L

Logging, 261

M

Man-In-The-Middle Attack, 44
Modbus TCP, 72
Modbus-Protokoll (RTU), 130

N

NTP, 21, 261

O

Offene Kommunikation

E-Mail einrichten, 93

FTP einrichten, 94

TCP, ISO-on-TCP, UDP, einrichten, 80

Offene Kommunikation

Verbindungsparametrierung, 80

OPC UA

DB-Variablen, 168

Einführung, 136, 137, 139

Endpunkte, 147

Identifizierung, 141

Namespace, 141

OpenSSL, 162

PLC-Variablen, 168

Schichtmodell, 166

Secure Channel, 165

Security-Einstellungen, 147

Sichere Verbindung, 165

Sicherheitsmechanismen, 155

Signieren und Verschlüsseln, 157

X.509-Zertifikate, 159

Zertifikatsgenerator, 160

OPC UA-Client

Grundlagen, 150

Zertifikat, 185

OPC UA-Server

Abtastintervall, 179

Adressbereich, 143

Adressierung, 175

Applikationsname, 174

Authentifizierung, 190

Grundlagen, 139

Inbetriebnahme, 173

Leistungssteigerung, 171

Performance, 171

Runtime-Lizenzen, 193

Schreib- und Leserechte, 168

Security-Einstellungen, 183

Sendeintervall, 179

Server-Zertifikat anpassen, 188

Server-Zertifikat erzeugen, 180

Subscription, 177

TCP-Port, 177, 178

XML-Exportdatei, 172

Open User Communication

Anweisungen, 73

Merkmale, 70

Protokolle, 71

OpenSSL, 162

P

PCT, 240

PG-Kommunikation, 21, 65

Private Key, 39

Protokolle für Open User Communication, 71

Prozedur 3964(R), 130

Public Key, 39

Punkt-zu-Punkt-Kopplung, 21, 130

PUT, 122

R

Record Protocol, 43

RFC 5280, 39

S

S7-Kommunikation, 21, 121, 248

S7-Routing, 235

Verbindungsressourcen, 248

Schnittstellen für Kommunikation, 17

Schnittstellen von Kommunikationsmodulen

Punkt-zu-Punkt-Kopplung, 19

Schnittstellen von Kommunikationsprozessoren, 18

Secure Communication, 39

Secure Socket Layer, 43

Security, 258

Selbstsignierte Zertifikate, 44

Server-Zertifikat, 188

Sicherheitsmaßnahmen, 258

Firewall, 260

Logging, 261

NTP, 261

SNMP, 262

Signatur, 45

SNMP, 21, 262

SSL, 43

Stammzertifikat, 47

Symmetrische Verschlüsselung, 41

Syslog, 261

Systemdatentyp, 74

T

TCON, 73

TCP, 21, 71, 80
TDISCON, 73
TLS, 43
Transport Layer Security, 43
TRCV, 73
TRCV_C, 73
TSEND, 73
TSEND_C, 73

U

UDP, 21, 71, 80
Uhrzeitsynchronisation, 21
URCV, 122
USEND, 122
USS-Protokoll, 130

V

Verbindung
 Anweisungen für Open User Communication, 73
 Diagnose, 254
Verbindung einrichten, 31
 ISO-Verbindung mit CP 1543-1, 86
 über Projektierung, 85
Verbindungsressourcen
 Anweige im Webserver, 253
 Anzeige in STEP 7, 250
 belegen, 249
 Datensatz-Routing, 248
 HMI-Kommunikation, 247
 modulspezifisch, 251
 S7-Routing, 248
 stationsspezifisch, 250
 Überblick, 30, 242

W

Webserver, 21
Write, 21

X

X.509, 39

Z

Zertifikatsinhaber, 44
Zertifizierungsstellen, 44