Dokumentation
S7Connector

Stand 14.01.2014

S. Rothenbacher GmbH

Automation Industrieelektronik GmbH

Zeppelinstraße 16

D-89160 Dornstadt, Germany

Phone +49-7348-201208

Fax +49-7348-201382

web www.Rothenbacher-GmbH.de

e-mail Info@Rothenbacher-GmbH.de

Inhaltsverzeichnis

1 Allgemeines

Unser Softwarepaket S7Conneector besteht aus einer einzigen .Net Dll (Bibliothek).

Die dll ist für sämtliche Windows Plattformen entwickelt worden.Durch das Linux Mono Projekt kann die dll auch unter Linux verwendet werden.

S7-200®, S7-300®, S7-400®,S7-1200®, HMI®, WINCC®, PCS7®, STEP® und SIMATIC®
sind eingetragene Warenzeichen der Siemens AG,

1.1 Support

Haben Sie Fragen oder Probleme mit der Installation oder der Anwendung des S7Conectors, so wenden Sie sich bitte an unseren Support. Sie erreichen ihn entweder telefonisch unter (07348) 20 12 08 oder per E-Mail über support@rothenbacher-gmbh.de. Schicken Sie uns Ihre Fragen oder die Problembeschreibung mit der von ihnen verwendeten .net version und Betriebssystemversion.

1.2 Projektunterstützung

Kontaktieren Sie uns, wenn Sie einen personellen Engpass oder einfach Bedarf an kompetenter Projektunterstützung haben. Gerne realisieren wir für Sie Projekte zum Festpreis. Setzen Sie sich im Bedarfsfalle einfach mit uns in Verbindung, wir erstellen Ihnen gerne ein unverbindliches Angebot.

1.3 Lieferumfang

Der S7Connector wird als Zip Datei geliefert. In dieser Zip Datei sind alle Basisprogramme, Beispiele und die Demoversion der DLL enthalten.

S7Conector	
Dll S7Connector	.net Dll zum Einbinden in Visuel Studio 2008
AktivX S7Conector	
S7Connector_ComInterop	Dll und tbl zum Einbinden in VB 4-6 und Delphi,Office Anwendungen
.Net Demo C#	
C# Demo .Net Framework 3.5	Sourcecode Beispiel in C# und .net Framwork 3.5
.Net Demo VB	
VB Demo .Net Framework 3.5	Sourcecode Beispiel in VB und .net Framwork 3.5
VB6.0 Demo	
VB6.0 Demo	Sourcecode Beispiel in VB6.0

1.4 Einschränkung der Demo Version

In der Demoversion stehen Ihnen sämtliche Funktionen der Vollversion zur Verfügung. Es erscheint nur beim Starten ein kleiner Hinweise, dass es sich um eine Demoversion des S7Connectors handelt.

1.5 Lizenzvereinbarungen

EULA / Endbenutzerlizenzvereinbarung

Diese Endbenutzerlizenzvereinbarung (englisch:End User License Aggreement - im folgenden als EULA abgekürzt) enthält die Bedingungen und Konditionen bezüglich der Verwendung dieser SOFTWARE (wie unten definiert). Diese EULA enthält Beschränkungen Ihrer Rechte in Bezug auf diese SOFTWARE. Sie sollten diese EULA sorgfältig lesen und als wichtiges Merkmal dieser SOFTWARE behandeln.

1. Vereinbarung zwischen Ihnen und S.Rothenbacher GmbH

Diese EULA ist eine gesetzlich bindende Vereinbarung zwischen Ihnen und der Firma S.Rothenbacher GmbH. Sie beabsichtigen, gesetzlich an diese EULA in demselben Umfang gebunden zu werden, als ob Sie und die Firma S.Rothenbacher GmbH diese EULA physisch unterschreiben würden. Indem Sie diese SOFTWARE installieren, kopieren oder anderweitig nutzen, erklären Sie sich einverstanden, an die in dieser EULA enthaltenen Bedingungen und Konditionen gebunden zu sein. Wenn Sie nicht mit allen Bedingungen und Konditionen dieser EULA einverstanden sind, dürfen Sie die SOFTWARE nicht installieren oder benutzen.

2. Definition von "SOFTWARE"

Diese EULA regelt die Verwendung von Software-Produkten der Firma S.Rothenbacher GmbH. (beigefügt oder anderweitig verfügbar) durch Sie - einzeln und kollektiv als "SOFTWARE" bezeichnet. Der Begriff "SOFTWARE" beinhaltet, neben dem von der Firma S.Rothenbacher GmbH. gelieferten Umfang:

1) Alle Revisionen, UPDATES und/oder UPGRADES hierzu

2) Alle Daten, Bilder, ausführbare Dateien, Datenbanken, Datenbanksysteme, Computersoftware oder ähnliche Elemente, die normalerweise mit Computersoftware-Produkten verteilt oder verwendet werden

3) Alle damit in Verbindung stehenden Datenträgern, Dokumentationen (beinhaltet physische, elektronische und online verfügbare Dokumente) und gedruckte Materialien.

3. Copyright

Die SOFTWARE gehört S.Rothenbacher GmbH. und/oder ihren Lizenzgebern und wird von Copyrightgesetzen und internationalen Verträgen geschützt. Sie dürfen den Copyright-Hinweis aus keiner Kopie der SOFTWARE entfernen.

4. Einräumung einer Lizenz

Die SOFTWARE wird Ihnen nicht verkauft. Stattdessen wird die SOFTWARE auf einer nicht exklusiven Grundlage an Sie - und nur an Sie - zum Gebrauch unter den Bestimmungen dieser Vereinbarung lizenziert. S.Rothenbacher GmbH. behält alle Titel und Eigentumsrechte an der SOFTWARE sowie alle Rechte, die Ihnen nicht ausdrücklich gewährt werden. Solange Sie keine Lizenz für die SOFTWARE erhalten und installiert haben, läuft die SOFTWARE im Demomodus und ist in ihrer Funktionalität eingeschränkt oder funktioniert eines Tages nach der Installation nicht mehr. Für Details zum Demomodus lesen Sie bitte die der SOFTWARE beigefügten Dokumentation.

5. Verwendung nur an einem Einzelarbeitsplatz

Die SOFTWARE darf nur an einem einzigen Arbeitsplatz genutzt werden. Ein Arbeitsplatz ist definiert durch die Kombination eines physischen oder virtuellen Computers (oder einer Session auf einem Terminal-Server) und einer Person. Sie dürfen die SOFTWARE auf jedem Computer eines Arbeitsplatzes installieren (z.B. Arbeitsstation und Notebook), wenn sichergestellt ist, dass die SOFTWARE zu keiner Zeit von mehr als einer Person verwendet wird.

6. Eine Archivierungskopie

Sie dürfen nur eine einzige Sicherungskopie anfertigen, die ausschließlich zu Archivierungszwecken genutzt werden darf. Diese darf nicht an Dritte weiter gegeben werden.

7. Dekompilierung, Entassemblierung oder Zurückentwicklung

Sie erkennen an, dass die SOFTWARE Betriebsgeheimnisse und andere Eigentumsinformationen der Firma S.Rothenbacher GmbH. und/oder ihren Lizenzgebern enthält. Sie dürfen die SOFTWARE nicht dekompilieren, entassemblieren oder auf irgendeine Art und Weise zurückentwickeln (Reverse Engineering) - noch sich mit anderen Aktivitäten beschäftigen, um die zugrunde liegenden Informationen zu erhalten, die für den Benutzer während der normalen Verwendung der SOFTWARE nicht sichtbar sind.

8. Aktualisierungen (Updates und Upgrades)

Diese Lizenz räumt Ihnen kein Recht auf irgendwelche Erweiterungen, Fehlerbeseitigungen, Programmkorrekturen oder Aktualisierungen der SOFTWARE ein - noch irgendwelche Support-Dienstleistungen. UPDATES (Aktualisierungen) sind definiert als neue Versionen der SOFTWARE, in denen sich die Hauptversionsnummer nicht geändert hat (die Hauptversionsnummer ist die erste Zahl der Versionskennzeichnung des Produktes. Beispiel: Die Hauptversionsnummer von "1.2.3.4" ist 1). Wird die Hauptversionsnummer geändert, dann wird diese neue Version der SOFTWARE als UPGRADE definiert. Verfügbare UPDATES werden kostenlos von der Firma S.Rothenbacher GmbH. über die jeweilige Produkt-Website oder auf der Hauptwebsite (www.rothenbacher-gmbh.de.de) zum Download zur Verfügung gestellt (Kosten für Ihre Internet-Verbindung und/oder Kosten des Transfers selber sowie jegliche Kosten, die in Verbindung mit dem Erhalt des UPDATES stehen, müssen von Ihnen übernommen werden).

9. Beendigung

Die Ihnen erteilte Lizenz ist bis zur Beendigung gültig. Die Beendigung kann zu jeder Zeit durch die Rückgabe der SOFTWARE (inkl. aller Kopien davon) an die Firma S.Rothenbacher GmbH. stattfinden. Außerdem wird die Gültigkeit Ihrer Lizenz automatisch beendet (auch ohne einen Hinweis von der Firma S.Rothenbacher GmbH.), wenn Sie einer Bedingung oder Kondition dieser Vereinbarung zuwiderhandeln. Sie stimmen bei einer solchen Beendigung zu, sämtliche Bestandteile der SOFTWARE (inkl. Kopien davon) an die Firma S.Rothenbacher GmbH. zurückzugeben. Bei Beendigung kann die Firma S.Rothenbacher GmbH. die ihr per Gesetz zustehenden Rechte durchsetzen. Die Bedingungen und Konditionen dieser Vereinbarung, die die Eigentumsrechte der Firma S.Rothenbacher GmbH. schützen, bleiben auch nach Beendigung in Kraft.

10. Haftungsausschluss

Die Firma S.Rothenbacher GmbH. garantiert nicht, dass die in der SOFTWARE enthaltenen Funktionen Ihre Anforderungen erfüllt und/oder dass der Betrieb der SOFTWARE ununterbrochen, fehlerfrei oder frei von arglistigem Code ist ("arglistiger Code" bezeichnet jeglichen Programm-Code, der entwickelt wurde, um andere Computer-Programme und/oder Computer-Daten zu kontaminieren, Computerbetriebsmittel zu verbrauchen, Daten zu ändern / zu löschen / aufzuzeichnen oder zu übermitteln - oder in irgendeiner anderen Art den Normalbetrieb von Computern, Rechnersystemen oder Computernetzwerken zu stören, einschließlich Viren, trojanischen Pferden, Droppern, Würmern, Logik-Bomben und Ähnliches).

Diese SOFTWARE wird WIE SIE IST geliefert – ohne irgendwelche Garantien. S.Rothenbacher GmbH ist nicht verpflichtet, UPDATES, UPGRADES oder technischen Support für diese SOFTWARE bereitzustellen. Die Firma S.Rothenbacher GmbH übernimmt außerdem keine Haftung für die Genauigkeit sämtlicher von S.Rothenbacher GmbH oder von Dritten zur Verfügung gestellten Informationen oder für irgendwelche Schäden, die direkt oder indirekt durch Aktionen oder Unterlassungen aufgrund dieser Informationen entstehen.

Sie übernehmen die volle Verantwortung für die Auswahl der SOFTWARE, um Ihre zukünftigen Ergebnisse zu erreichen, sowie für die Installation, Benutzung und die Ergebnisse, die Sie von der SOFTWARE erhalten. Weiterhin übernehmen Sie das volle Risiko in Bezug auf Qualität und Leistung der SOFTWARE.

Sollte sich die SOFTWARE als fehlerhaft erweisen, übernehmen Sie (und nicht die Firma S.Rothenbacher GmbH oder ihre Distributoren oder Händler) die gesamten Kosten für alle notwendigen Service-, Reparatur- und/oder Korrektur-Leistungen.

In keinem Fall haftet die Firma S.Rothenbacher GmbH oder ihre Lizenzgeber für direkte, indirekte, beiläufige oder besondere Schäden noch für Folgeschäden oder irgendwelche Verluste, entgangene Gewinne, entgangene Einnahmen, entgangene Einsparungen oder für entstandene Datenverluste, die durch oder in Verbindung mit dieser SOFTWARE oder dieser Vereinbarung entstehen, selbst wenn die Firma S.Rothenbacher GmbH oder ihre Lizenzgeber über die Möglichkeit solcher Schäden unterrichtet wurde. In jedem Fall ist die Haftung auf den für die SOFTWARE gezahlten Betrag beschränkt, unabhängig von der Art des Schadenfalls.

11. Schlussklausel

Diese Vereinbarung bindet Sie wie auch Ihre Angestellten, Arbeitgeber, Auftragnehmer und Agenten sowie alle Nachfolger und Bevollmächtigten. Diese Vereinbarung ist die gesamte Vereinbarung zwischen uns und hat Vorrang vor allen anderen Absprachen und Vereinbarungen. Falls eine Bedingung oder Kondition dieser Vereinbarung ungültig ist oder wird, bleiben alle anderen Bedingungen und Konditionen dieser Vereinbarung davon unberührt und behalten ihre Gültigkeit. In einem solchen Fall verpflichten sich beide Parteien, die ungültig Bedingung und/oder Kondition durch eine gültige Bedingung und/oder Kondition zu ersetzten, die in ihrer rechtlichen, wirtschaftlichen und technischen Bedeutung möglichst gleichkommt. Gerichtsstand ist Deutschland.

1.6 Kompatibilität Feldbus

1.6.1 TCP/IP

	Windows 32	Windows CE	Linux
CP341	ja	ja	ja
CP341-1	ja	ja	ja
CP241-1	ja	ja	ja
CP441-1	ja	ja	ja
VIPA Ethernet	ja	ja	ja
Netlink PRO	ja	ja	ja

1.6.2 MPI(RS232)

	Windows 32	Windows CE	Linux
PC Adapter (RS232)	ja	ja	ja
PI Adapter (RS232)	ja	ja	ja
Helmholz SSW7 (RS232)	ja	ja	ja
VIPA Green Cabel	ja	ja	ja

1.6.3 PPI(RS232)

	Windows 32	Windows CE	Linux
PPI Adapter (RS232)	ja	ja	ja

1.6.4 Online PG/PC

	Windows 32	Windows CE	Linux
PC Adapter USB (MPI) und Kompatible	ja	nein	nein
PC Adapter (RS232) (MPI) und Kompatible	ja	nein	nein
PC Adapter USB (DP)	ja	nein	nein
PC Adapter (RS232) (DP)	ja	nein	nein
CP5411 (MPI)	ja	nein	nein
CP5411 (DP)	ja	nein	nein
Netlink und Netlink PRO	ja	nein	nein

1.7 Kompatibilität Betriebssystem

1.7.1 Windows X86 und X64 /ARM / MIPS

		32 Bit	64 Bit
Windows 95	.Net Framwork / Mono Framwork	-	-
Windows Me	.Net Framwork / Mono Framwork	-	-
Windows 98	.Net Framwork / Mono Framwork	X	-
Windows NT Workstation 4.0 (alle Versionen - Service Pack 6a erforderlich)	.Net Framwork / Mono Framwork	X	-
Windows 2000 Professional	.Net Framwork / Mono Framwork	X	-
Windows XP Home Edition	.Net Framwork / Mono Framwork	X	X
Windows XP Professional	.Net Framwork / Mono Framwork	X	X
Windows Vista	.Net Framwork / Mono Framwork	X	X
Windows 7	.Net Framwork / Mono Framwork	X	X
Windows 8	.Net Framwork / Mono Framwork	X	X
Windows 2003 Server	.Net Framwork / Mono Framwork	X	X
Windows 2008 Server	.Net Framwork / Mono Framwork	X	X
Windows CE .NET 4.0 – 4.2	.NET Compact Framework	X	-
Windows CE 5.0	.NET Compact Framework	X	-
Windows Embedded CE 6.0	.NET Compact Framework	X	-

1.7.2 Linux / MAC / ARM

		32 Bit	64 Bit
Ubuntu	Mono Framwork	X	X
Red Hat	Mono Framwork	X	X
OpenSuse	Mono Framwork	X	X
Debian	Mono Framwork	X	X
LinuxMint	Mono Framwork	X	X
Knoppix	Mono Framwork	X	X
Raspberry PI - Raspbian	Mono Framwork	X	-
CubieBoard 2	Mono Framwork	X	-
Apple OSX	Mono Framwork	X	X

Und sicher viele mehr ….

2 Grundlegendes zur Anwendung

2.1 Der Namespace

Alle im folgenden verwendete Funktionen sowie Variablen des S7-Connectors befinden sich im Namespace: „Ro.Communication.Protocol”.

Die Dokumentation bezieht sich darauf das der Namespace dem Projekt bekannt gegeben wurde.
Beispiel für C#

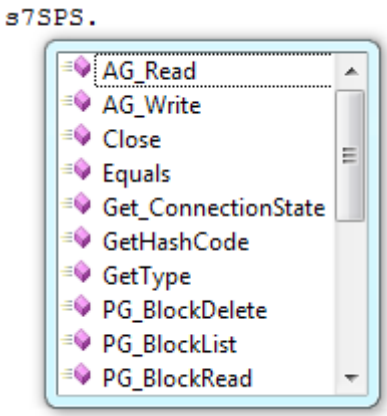
```
using Ro.Communication.Protocol;
```

2.2 Die Basisklasse

Die Basisklasse „S7.Base“ stellt den Ausgangspunkt der Projektierung dar.
Sie beinhaltet alle Funktionsaufrufe um z.B. Daten aus einem DB zu lesen bzw. zu schreiben.
Hierfür wird eine Instanz der Basisklasse benötigt.
Beispiel für C#

```
private S7.Base s7SPS;
```

Der Zugriff auf die Funktionen erfolgt nun über diese Instanz:



2.3 Die Überladung der Basisklasse

Diese Basisklasse wird durch den Verbindungsaufbau über eine der abgeleiteten Klassen „überschrieben“ und nutzt dadurch die physikalischen und logischen Verbindungsparameter, die bei der Verbindung angegeben werden.

Mögliche Verbindungsklassen zur Überladung sind:

- S7.TCPIP
- S7.MPI
- S7.PPI
- S7.ONLINE

Eine genaue Anleitung zu den einzelnen Verbindungsarten finden sie unter „3 Verbindungsaufbau“

3 Verbindungsaufbau

Bevor Sie auf Daten der SPS Steuerung zugreifen können muss die Kommunikation initialisiert werden.Stellen Sie die gewünschte Kommunikation über den Aufruf der Verbindungsaufbau-funktionen(3.3,3.4,3.5,3.6) ein. Bei Zugriff über die S7 Online PG/PC Schnittstelle konfigurieren Sie über „Systemsteuerung/PG/PC Schnittstelle einstellen“ dort können Sie alle Verbindungsparameter einstellen. Ist der Verbindungsaufbau erfolgreich verlaufen können Sie mit den Funktionen (4.1,4.3)Daten aus der Steuerung lesen.Die Fehlerbehandlung der einzelne Funktionen können Sie über die try{ }Code... catch (Exception E) abfangen. Eine Liste der möglichen Fehler und ihrer Ursachen entnehmen Sie dem Kapitel (6).

3.1 Grundlegendes zum S7 Protokoll

Das S7 Protocol und seine Ethernet-Implementierung basiert auf ISO TCP (RFC1006), das Design, ist nachrichtenorientiert Jeder Nachricht namens PDU (Protocol Data Unit),ist in seiner maximale Länge beschrenkt Die PDU Size wird beim Verbindungsaufbau ausgetauscht.Jeder Lese oder Schreib Auftrag kann diese PDU seize nicht überschreiten (Der S7Connetor kann beim Lessen „AG_Read diese grenze überschreiten aber nur für den Anwender in Wirklichkeit spiltet der S7Connector auftrage die nicht in die PDU Size passen)
Ein S7-Protokoll besteht immer aus Einem-Header,Einem Satz von Parametern. Dann folgen optionale Daten wie Dazugehörige Parameter-Daten. oder einem Datenblock.

3.2 S7 Protokoll Kompatibilität

	S7 200/LOGO	S7 300/400	WinAC	S7 1200	S7 1500	Drive
AG_Read	X	X	X	X	X	X
AG_Write	X	X	X	X	X	X
PG_RunMode	-	X	X	-	-	X
PG_Start	-	X	X	-	-	X
PG_Stop	-	X	X	-	-	X
PG_GeneralReset	-	X	X	-	-	X
PG_ClockRead	-	X	X	-	-	X
PG_ClockWrite	-	X	X	-	-	X
PG_BlockList	-	X	X	-	-	X
PG_BlockDelete	-	X	X	-	-	X
PG_BlockRead	-	X	X	-	-	X
PG_BlockWrite	-	X	X	-	-	X

3.3 S7 TCP-IP

Die Überladung durch „S7.TCPIP“ referenziert die Instanz der Basisklasse „S7.Base“ mit einer S7-Kommunikation über TCP / IP (RFC1006).
RFC 1006 (ISO Transportdienst über TCP) ist eine Protokoll-Erweiterung für das TCP-Protokoll. Hierbei werden zusätzlich zu den TCP Daten weitere Informationen zwischen den Teilnehmern übertragen, um bestimmte Dienste für den Anwender erbringen zu können (ISO-Dienste als Erweiterung zu TCP).

	Beschreibung
Position	Der Konstruktor „S7.TCPIP“ wurde in 2 Varianten überlagert. TCPIP(string IP, int Port, int TimeoutInMS, S7TypeSeries TS) TCPIP(string IP, int Port, int TimeoutInMS, string TSAP_E, string TSAP_F)
Codeauszug C#	try { s7SPS = new S7.TCPIP(“192.168.2.100”, 102,1000, S7TypeSeries.S7300); } catch (Exception e) { // Exception Handling }
Beschreibung	Die Funktion dient zum Verbindungsaufbau über TCP/IP. Als Rückgabe erhalten Sie eine Instanz auf den S7connector (Beispiel s7SPS). Die Funktion enthält eine Überlagerung mit der Sie die Möglichkeit haben die Tsap Parameter selbst einzustellen. Für die Standard Fälle sollte aber die Funktion mit dem enum Datentyp S7TypeSeries ausreichend sein. Weitere enum Type sind: S7Connector.TypeSeries.S71200;S7Connector.TypeSeries.S7200; S7Connector.TypeSeries.S7300;S7Connector.TypeSeries.S7400;
Parameter	– IP : TCP-IP Adresse der SPS z.b. "192.168.2.100" – Port : TCP-IP Port der SPS (standard: 102) – TimeoutInMS : Zeitspanne in Millisekunden in der die SPS eine Anfrage beantworten muss, bevor eine "Exception" ausgelöst wird.  Betrifft nicht die Netzwerkverbindung da diese Timeout-Zeit durch das System(Windows/Linux) gesteuert wird! – TS : Enumerierung der Standard-S7Typen(S7200, S7300, S7400, S71200,S71500) – TSAP_E : Siehe 3.3.1 Manueller TSAP – TSAP_F : Siehe 3.3.1 Manueller TSAP

3.3.1 Manueller TSAP

Eigener TSAP
Der eigene TSAP legt die Verbindungsadresse im eigenen System fest, über die Daten ausgetauscht werden sollen.

Fremder TSAP
Der fremde TSAP bestimmt die Verbindungsadresse des anderen Systems. Um die Verbindung aufbauen zu können muss der “eigene TSAP” dem “fremden TSAP” des anderen Systems entsprechen.
Er besteht aus genau zwei Gruppen (Bytes). Jede Gruppe wird aus zwei hexadezimalen Zeichen gebildet, wobei die beiden Gruppen Punkt voneinander getrennt werden.

Erste Gruppe: enthält Gerätekennungen für die in der S7 Ressourcen bereit gestellt sind.
01 PG oder PC
02 OS (Bedien- bzw. Beobachtungsgerät)
03 z.B. SIMATIC S7-SPS

Zweite Gruppe: enthält die Adressen dieser Komponenten.
Bit (7 ..5) Rack Nr. x 2
Bit (4 ..0) Steckplatz der CPU
Standardbeispiele für die S7 Verbindung
Standard in den meisten Fällen sind die TSAPs folgendermaßen einzustellen:

S7 200: TSAP_E = 10.0 TSAP_F = 10.0 (Wird im Ethernet Assistent von Microwin eingestellt)
S7 300: TSAP_E = 1.0 TSAP_F = 3.2
S7 400: TSAP_E = 1.0 TSAP_F = 3.3
S7 1200: TSAP_E = 10.0 TSAP_F = 3.1
S7 1500: TSAP_E = 10.0 TSAP_F = 3.1
WinLC: TSAP_E = 1.0 TSAP_F = 1.2

Achtung Variablenbytes einer S7 200 schreiben und lesen ist über denn DB 1 gelöst. Diese Vbyte Adressen sind Analog zum DB1 z.b. VB0 entspricht DB1.DBB0

Achtung die Anzahl der Verbindungen ist in der Simatic CPU begrenzt. Das bedeutet nicht das unendlich viele Kommunikationen zu einer CPU aufgebaut werden können. Die verfügbaren Ressourcen ihrer CPU entnehmen Sie dem Simatic CPU Handbuch.

3.4 S7 MPI (RS232) Verbindungsaufbau

Die Überladung durch „S7.MPI“ referenziert die Instanz der Basisklasse „S7.Base“ mit einer S7-Kommunikation über MPI. Jede S7-Station besitzt eine MPI-Schnittstelle (Multi Point Interface). Über diese Schnittstelle werden die am Automatisierungsprojekt beteiligten S7-Stationen mit dem Programmiergerät vernetzt.Bei MPI handelt es sich um Schnittstellen mit RS485-Physik, über die ein Siemens-internes Protokoll abgewickelt wird. Es können bis zu 32 Teilnehmer über eine Zweidrahtleitung verbunden werden. Jeder Teilnehmer bekommt eine MPI-Adresse zugewiesen, über die er ansprechbar ist.

	Beschreibung
Position	MPI(string ComPort, int TimeoutInMS, int BaudRate, int DataBits, System.IO.Ports.StopBits StopBits, System.IO.Ports.Parity Parity, byte mpiAddressPG, byte mpiAddressAG)
Codeauszug C#	Beispiel für C# `try` `{` `  s7SPS = new S7.MPI("COM1", 1000, 38400, 8, 1, System.IO.Ports.Parity.Odd, 0, 2);` `}` `catch (Exception e)` `{` `  // Exception Handling` `}`
Beschreibung	Die Funktion dient zum Verbindungsaufbau über MPI. Als Rückgabe erhalten Sie eine Instanz auf den S7connector (Beispiel s7SPS).Diese Verbindung kann nur seriell (ausgenommen USB PC Adapter die einen Vcom (Virtueller com Port) mitbringen) kommunizieren. Für Verbindungen mit einem MPI PC Adapter für USB nutzen Sie die Verbindungsfunktion (3.6)
Parameter	– ComPort : Adresse der seriellen Schnittstelle des Host PCs z.B. “COM1” – TimeoutInMS :Zeitspanne in Millisekunden in der die SPS eine Anfrage beantworten muss, bevor eine "Exception" ausgelöst wird. ⚠ Betrifft nicht die Netzwerkverbindung da diese Timeout-Zeit durch das System(Windows/Linux) gesteuert wird! – BaudRate : Geschwindigkeit der seriellen Datenübertragung z.B. 38400 – DataBits : Anzahl Datenbits der seriellen Datenübertragung z.B. 8 – StopBits : Anzahl Stopbits der seriellen Datenübertragung – Parity : Parität der seriellen Datenübertragung – mpiAddressPG : MPI Adresse des Host PCs z.B. 0 – mpiAddressAG : MPI Adresse der SPS z.B. 2

3.5 S7 PPI (RS232) Verbindungsaufbau

Die Überladung durch „S7.PPI“ referenziert die Instanz der Basisklasse „S7.Base“ mit einer S7-Kommunikation über PPI. Beim Point to Point Interface (PPI) handelt es sich um Schnittstellen mit RS485-Physik, über die ein Siemens-internes Protokoll abgewickelt wird.

	Beschreibung
Position	PPI(string ComPort, int TimeoutInMS, int BaudRate, int DataBits, System.IO.Ports.StopBits StopBits, System.IO.Ports.Parity Parity, byte ppiAddressPG, byte ppiAddressAG)
Codeauszug C#	try { s7SPS = new S7.PPI("COM1", 1000, 38400, 8, System.IO.Ports.StopBits.One, System.IO.Ports.Parity.None,0, 2); } catch (Exception e) { // Exception Handling }
Beschreibung	Die Funktion dient zum Verbindungsaufbau über PPI . Als Rückgabe erhalten Sie ein e Instanz auf den S7connector (Beispiel s7SPS).Diese Verbindung kann nur seriell kommunizieren. Für Verbindungen mit einem PPI PC Adapter für USB nutzen Sie die Verbindungsfunktion (3.6) .Diese Verbindungsart wird ausschließlich bei der Simatic S7200 Serie verwendet.
Parameter	– ComPort : Adresse der seriellen Schnittstelle des Host PCs z.B. “COM1” – TimeoutInMS :Zeitspanne in Millisekunden in der die SPS eine Anfrage beantworten muss, bevor eine "Exception" ausgelöst wird. ⚠ Betrifft nicht die Netzwerkverbindung da diese Timeout-Zeit durch das System(Windows/Linux) gesteuert wird! – BaudRate : Geschwindigkeit der seriellen Datenübertragung z.B. 38400 – DataBits : Anzahl Datenbits der seriellen Datenübertragung z.B. 8 – StopBits : Anzahl Stopbits der seriellen Datenübertragung z.B 1 – Parity : Parität der seriellen Datenübertragung – ppiAddressPG : PPI Adresse des Host PCs z.B. 0 – ppiAddressAG : PPI Adresse der SPS z.B. 2

⚠

Achtung Variablenbytes einer S7 200 schreiben und lesen ist über den DB 1 gelöst. Diese Vbyte Adressen sind analog zum DB1 z.b. VB0 entspricht DB1.DBB0

3.6 S7 Online PG/PC Schnittstelle Verbindungsaufbau

Die Überladung durch „S7.ONLINE“ referenziert die Instanz der Basisklasse „S7.Base“ mit einer S7-Kommunikation über die bei Siemens in „PG/PC Schnittstelle“.

S7 Online unterstützt MPI USB und RS232 und Profibus DP kompatible Kommunikationen.(Auch Netx und Netlink)

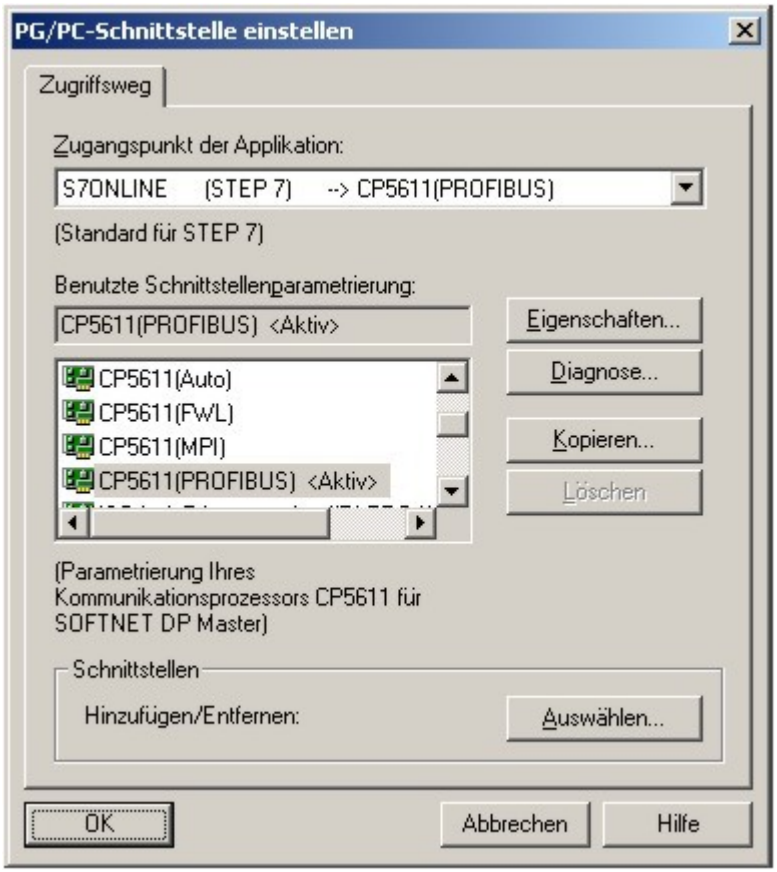
	Beschreibung
Position	ONLINE(string CP_Syntax, int TimeoutInMS, byte mpiAddressAG)
Codeauszug C#	`try` `{` `  SPS = new S7.ONLINE("S7ONLINE", 1000, 2);` `}` `catch (Exception e)` `{` `  // Exception Handling` `}`
Beschreibung	Die Funktion dient zum Verbindungsaufbau über die PG/PC Schnittstelle. Als Rückgabe erhalten Sie eine Instanz auf den S7connector (Beispiel s7SPS).Diese Verbindung nutzt die SIEMENS eigenen Kommunikation welche auch von Step7 verwendet wird. Vorteil dieser Verbindungsart ist , dass auch mehrere MPI Verbindungen von einem Rechner aus möglich sind z.B. Step7 und S7conector sind gleichzeitig online. Des weiteren werden auch USB Adapter sowohl als auch Profibus CP unterstützt.
Parameter	– CP_Syntax : Name des in „PG/PC Schnittstelle einstellen“ angelegten Zugangspunkts der Applikation z.B. “S7ONLINE” – TimeoutInMS : Zeitspanne in Millisekunden in der die SPS eine Anfrage beantworten muss, bevor eine "Exception" ausgelöst wird. – mpiAddressAG : MPI Adresse der SPS z.B. 2

3.6.1 Einstellen der PG/PC-Schnittstelle für die Kommunikation

Wenn Sie eine Kommunikation über Simatic Net aufbauen möchten, sollten Sie auch die PG/PC-Schnittstelle einstellen. Gehen Sie dazu folgendermaßen vor:

1. Wählen Sie dazu im Windows Betriebssystem „Start > Systemsteuerung > PG/PC-Schnittstelle einstellen

2. Wählen Sie den CP, der in Ihrem PC System installiert ist und über den Sie kommunizieren möchten.



3. Bestätigen Sie ihre Einstellungen.

Hinweis: (für PROFIBUS-Systeme benötigen Sie z. B. die Siemens-Karte CP5611 oder 5511 oder PC-Adapter (USB oder RS232 6ES7972-0CB20-0XA0) oder einen entsprechende andere Karte).

4 AG Funktionen

4.1 AG_Read and (MultibleRead)

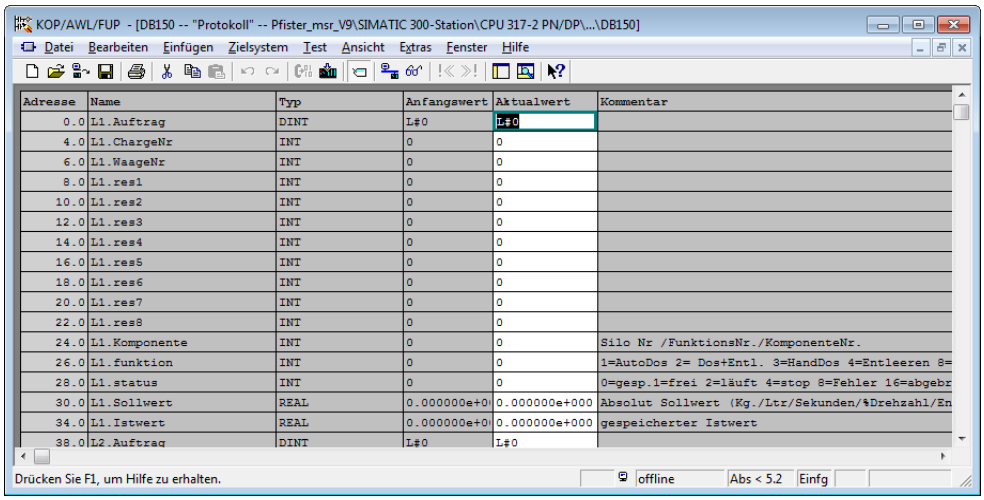
Die Funktion „AG_Read“ liest Daten von einer verbundenen SPS.
Die Funktion „AG_Read“ hat einen überlagerung die es ermöglicht Daten aus der CPU im sogenaten MultibelRead verfahren zu lessen,Dabei werden die Adressen als Arry angegeben.

	Beschreibung
Position	public virtual object AG_Read(Int16 DBNr, Int32 StartAdr, Int16 Count, S7DataTypes RetType, S7MemmoryTypes MemArea) public virtual object AG_Read(Int16 DBNr[], Int32 StartAdr[], Int16 Count[], S7DataTypes RetType[], S7MemmoryTypes MemArea[])
Codeauszug C#	try { short DBNr = Convert.ToInt16(TB_DBNr_Read.Text); short Count = Convert.ToInt16(TB_Count_Read.Text); int Start = Convert.ToInt32(TB_StartAddress_Read.Text); object Result = SPS.AG_Read(DBNr, Start, Count,Read_Selected_DataType, Read_Selected_MemmoryType); LB_Result.Items.Clear(); for (int i = 0; i < (Result as Array).Length; i++) LB_Result.Items.Add("VAL" + i.ToString("0000") + " = " + (Result as Array).GetValue(i).ToString()); } catch (Exception E) { // Exception Handling } }
Beschreibung	(Datablock, Flag, Input, Output, Counter,Timer, Periphery, InstanceDatablock, LocalData, Localdataofpreviousblock) (CHAR, BYTE, WORD, INT, REAL, DWORD, BOOL, S5TIME,TIMEIEC32 ,COUNTER)
Parameter	– DBNr : DB Adresse, von dem gelesen werden soll. Beim Lesen von anderen Speicherbereichen (Merkerbereich, E/A Bereich usw) muss die Adresse immer "0" sein. – StartAdr : Byteadresse, wo der Leseauftrag startet. – Count : Anzahl Daten, die gelesen werden sollen. – RetType : Datenart, die gelesen werden soll.4.5 – MemArea : Speicherbereich, von dem gelesen werden soll.4.4

Achtung Normalerweise ist das beachten der PDU Size nicht nötig da der S7Connector einen zugrossen Auftrag selbst Splitet und das für den Normalen Programmiere keinen Auswirkung hat und auch normalerweise nicht bemerkt wird.
Ausnahme hierbei ist die verwendung der "AG_ReadMultibelRead " überlagerung die gesamt Auftrags grösse darf die PDU Size in diesem Fall nicht überschreiten.Sihe dazu auch:3.1)

4.2 AG_Read Arbeiten mit Strukturen

In den meisten Fällen ist es in der Praxis so das bei zugriffen auf DB einr Simatic S7 die Daten nicht Kositen sind das bedeutet das z.b. 16 Bool werte dann 3 Word Werte 1 Int Wert ,1 Bool , 1 Real wert



Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	L1.Auftrag	DINT	L#0	L#0	
4.0	L1.ChargeNr	INT	0	0	
6.0	L1.WaageNr	INT	0	0	
8.0	L1.res1	INT	0	0	
10.0	L1.res2	INT	0	0	
12.0	L1.res3	INT	0	0	
14.0	L1.res4	INT	0	0	
16.0	L1.res5	INT	0	0	
18.0	L1.res6	INT	0	0	
20.0	L1.res7	INT	0	0	
22.0	L1.res8	INT	0	0	
24.0	L1.Komponente	INT	0	0	Silo Nr /FunktionsNr./KomponenteNr.
26.0	L1.funktion	INT	0	0	1=AutoDos 2= Dos+Entl. 3=HandDos 4=Entleeren 8=
28.0	L1.status	INT	0	0	0=gesp.1=frei 2=läuft 4=stop 8=Fehler 16=abgebr
30.0	L1.Sollwert	REAL	0.000000e+0	0.000000e+000	Absolut Sollwert (Kg./Ltr/Sekunden/%Drehzahl/En
34.0	L1.Istwert	REAL	0.000000e+0	0.000000e+000	gespeicherter Istwert
38.0	L2.Auftrag	DINT	L#0	L#0	

Wenn sie diese vielen Unterschiedlichen werte jeweis mit einem Lesse Auftrag aus der SPS Lesen so ist diese nicht besonders Perfomant.Besser ist es denn gesamte datenbereich mit einem einzigen Leseauftrag zu lessen und die Daten dann zu interpretieren z.b.

Beschreibung

```
byte[] Result = (byte[])SPS.AG_Read(150, 0, 76, S7Connector.DataTypes.BYTE,
S7Connector.MemmoryTypes.Datablock);
LB_Result.Items.Clear();
UInt32[] l_auftag = (UInt32[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(0,
4).ToArray(), S7Connector.DataTypes.DWORD);
UInt16[] i_charge = (UInt16[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(4,
2).ToArray(), S7Connector.DataTypes.WORD);
UInt16[] i_csoll = (UInt16[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(8,
2).ToArray(), S7Connector.DataTypes.WORD);
UInt16[] i_waage = (UInt16[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(6,
2).ToArray(), S7Connector.DataTypes.WORD);
Int16[] i_komponente = (Int16[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(24,
2).ToArray(), S7Connector.DataTypes.INT);
UInt16[] i_funktion = (UInt16[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(26,
2).ToArray(), S7Connector.DataTypes.WORD);
UInt16[] i_status = (UInt16[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(28,
2).ToArray(), S7Connector.DataTypes.WORD);
float[] r_soll = (float[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(30,
4).ToArray(), S7Connector.DataTypes.REAL);
float[] r_ist = (float[])S7Connector.Base.DataToRetType(Result.ToList().GetRange(34,
4).ToArray(), S7Connector.DataTypes.REAL);
LB_Result.Items.Add("l_auftag = " + Convert.ToString(l_auftag[0]));
LB_Result.Items.Add("i_charge = " + Convert.ToString(i_charge[0]));
LB_Result.Items.Add("i_waage = " + Convert.ToString(i_waage[0]));
LB_Result.Items.Add("i_komponente = " + Convert.ToString(i_komponente[0]));
LB_Result.Items.Add("i_funktion = " + Convert.ToString(i_funktion[0]));
LB_Result.Items.Add("i_status = " + Convert.ToString(i_status[0]));
LB_Result.Items.Add("r_soll = " + Convert.ToString(r_soll[0]));
LB_Result.Items.Add("r_ist = " + Convert.ToString(r_ist[0]));
LB_Result.Items.Add("i_csoll = " + Convert.ToString(i_csoll[0]));
```

4.3 AG_Write

Die Funktion „AG_Write“ schreibt Daten in eine verbundene SPS.

	Beschreibung
Position	public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, string Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, byte[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, UInt16[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, Int16[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, UInt32[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, float[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, S5Time[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, TimeIEC32[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, Counter[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdrInBits, MemmoryTypes MemArea, bool Value)
Codeauszug C#	try { short DBNr = Convert.ToInt16(TB_DBNr_Read.Text); short Count = Convert.ToInt16(TB_Count_Read.Text); int Start = Convert.ToInt32(TB_StartAddress_Read.Text); switch (CB_DataType_Read.Text) { case "BYTE": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new byte[] { Convert.ToByte(TB_Value.Text) }); break; case "WORD": case "DATE": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new UInt16[] { Convert.ToUInt16(TB_Value.Text) }); break; case "INT": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new Int16[] { Convert.ToInt16(TB_Value.Text) }); break; case "REAL": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new float[] { float.Parse(TB_Value.Text) }); break; case "DWORD": case "TIME_OF_DAY": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new UInt32[] { Convert.ToUInt32(TB_Value.Text) }); break;

	`` case "S5TIME": S7Connector.S5Time[] s5Times = new S7Connector.S5Time[2]; s5Times[0] = new S7Connector.S5Time(new TimeSpan(0, 0, 0, 10, 0)); s5Times[1] = new S7Connector.S5Time(new TimeSpan(0, 0, 0, 11, 0)); SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, s5Times); break; case "TIMEIEC32": S7Connector.TimeIEC32[] s7Times = new S7Connector.TimeIEC32[1]; s7Times[0] = new S7Connector.TimeIEC32(new TimeSpan(0, 0, 2, 1, 100)); SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, s7Times); break; case "COUNTER": S7Connector.Counter[] s7Conter= new S7Connector.Counter[1]; s7Conter[0] = new S7Connector.Counter(Convert.ToInt16(TB_Value.Text)); SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, s7Conter); break; case "BOOL": bool Val = false; if (TB_Value.Text.ToLower() == "true") Val = true; SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, Val); break; } ```
Beschreibung	
Parameter	– DBNr : DB Adresse, die beschrieben werden soll. Beim Beschreiben von anderen Speicherbereichen (Merkerbereich, E/A Bereich) muss die Adresse immer "0" sein. – StartAdr : Byteadresse, wo der Schreibauftrag startet. – MemArea : Speicherbereich, von dem gelesen werden soll.4.4 – Value(s) : Werte, die geschrieben werden sollen. Beim Schreiben von Boolschen Werten, wird nur ein Wert berücksichtigt. Der Datentyp wird durch die Überlagerung des Parameters automatisch erkannt.

4.4 Speicherbereich

Beim Laden in die CPU werden die Bausteine des S7-Programms in einen Ladespeicher transferiert. Das Betriebssystem generiert daraus ein ausführbares Programm im Arbeitsspeicher (RAM) der CPU. Daneben gibt es den Systemspeicher (RAM), der in bestimmte Bereiche unterteilt ist:

MemArea	S7-Notation		
Datablock Datenbausteine	DB	Datenbausteine werden im Step7 erstellt in Größe und Datenstruktur. Sie können entweder als (Globale DB) oder sie sind einem bestimmten FB oder SFB zugeordnet (Instanz-DB).	
Input Eingänge	I E	Speicherbereich, der von der SPS jeweils vor Beginn oder am Ende des Zyklus mit den Daten der Ein/Ausgabebaugruppen beschrieben wird.	
Output Ausgänge	Q A		
Flag Merker	F M	Anwender - Speicherbereich	
Timer	T		
Counter, Zähler	C Z		
LocalData Lokaldaten	L	Speicherbereich für temporäre Daten eines Codebausteins. Der Aufbau des jeweiligen Lokaldaten-Stacks ergibt sich aus der zugehörigen Deklarationstabelle	
Periphery Eingangsperipherie	PE	Speicherbereich, der direkt mit den Daten der Ein-/Ausgabebaugruppen in Verbindung steht. Durch Zugriff auf Daten des Peripheriebereichs können die Ein/Ausgabebaugruppen unabhängig vom OB1-Zyklus erreicht werden.	
Periphery Ausgangsperipherie	PA		

Auf alle Speicherbereiche kann durch den S7Conector lesend und schreibend zugegriffen werden. Die folgende Tabelle zeigt die Speicheraufteilung. Die Menge der verfügbaren Operanten (d.h. die Größe des Systemspeichers) ist abhängig von der jeweiligen CPU.

4.5 Datentypen

Jeder elementare Datentyp verfügt über einen zugeordneten Speicherplatz mit fester Länge. Der Datentyp BOOL zum Beispiel hat nur ein Bit, ein Byte (BYTE) besteht aus 8 Bits, ein Wort (WORD) sind 2 Bytes (bzw. 16 Bits), ein Doppelwort (DWORD) hat 4 Bytes (bzw. 32 Bits). Die folgende Tabelle zeigt alle vorhandenen elementaren Datentypen:

S7 Datentype	C# Datentype	Beschreibung	Bits	Wertebereichs
BOOL	bool	Einzelnes Bit	1	TRUE FALSE
BYTE	byte	Festpunktzahl	8	0 bis 255
WORD	uint16	Festpunktzahl	16	0 bis 65535
DWORD	uint32	Festpunktzahl	32	
INT	int16	Festpunktzahl	16	-32768 bis 32767
DINT	int32	Festpunktzahl	32	-2147483648 bis 2147483647
REAL	float	Gleitpunktzahl	32	1.0 1.238 1.2E-2 -5E12 100.0 -1000.0 -123E-18
S5TIME	S5Time	Zeitwert Simatic 16	16	S5T#0ms bis S5TIME#2h46m30s
TIME	TimeIEC32	Zeitwert IEC 32	32	-24d20h31m23s647ms bis 24d20h31m23s647ms
CHAR	string	ASCII-Zeichen	8	
DATE			16	
TIME_OF_DAY			32	

5 PG Funktionen

Die PG Funktion (Programmiergerätfunktionen) stehen nur bei Verbindungen zu S7 300 & S7 400 zur Verfügung. Bei einer Conection zu einer S7 200 oder S 1200, S7 1500 wird einen Exception ausgelöst.

5.1 PG_RunMode

	Beschreibung
Position	public virtual bool PG_RunMode();
Codeauszug C#	TB_Runmode.Text = ((SPS.PG_RunMode())?"Run":"Stop");
Beschreibung	Die Funktion „PG_RunMode“ gibt den momentanen Betriebszustand der Baugruppe zurück. Ist der Rückgabewert der Funktion „TRUE“, so befindet sich die SPS in der Betriebsart „RUN“. Andernfalls ist der Zustand der SPS „STOP“.
Parameter	

5.2 PG_Start

	Beschreibung
Position	public virtual bool PG_Start()
Codeauszug C#	SPS.PG_Start();
Beschreibung	Die Funktion „PG_Start“ sendet an die verbundene SPS einen Startbefehl. Wenn die Stellung des Schlüsselschalters „START“ ist und kein unbehandelter Fehler ansteht, wechselt die SPS in die Betriebsart „RUN“.Im Programm angesteuerte Ausgänge sind geschaltet.
Parameter	

5.3 PG_Stop

	Beschreibung
Position	public virtual bool PG_Stop()
Codeauszug C#	SPS.PG_Stop();
Beschreibung	Die Funktion „PG_Stop“ sendet an die verbundene SPS einen Stoppbefehl. Die SPS wechselt in die Betriebsart „STOP“.Der Programmbetrieb ist unterbrochen. Alle Ausgänge sind spannungsfrei geschaltet oder werden auf Default-Werte gesetzt.
Parameter	

5.4 PG_GeneralReset

	Beschreibung
Position	public virtual bool PG_GeneralReset()
Codeauszug C#	if(MessageBox.Show("Are You Sure?","General Reset",MessageBoxButtons.YesNo) == DialogResult.Yes) SPS.PG_GeneralReset();
Beschreibung	Die Funktion „PG_General_Reset“ führt die Funktion „Urlöschen“ auf der SPS aus. Nur in Betriebsart STOPP möglich.Das Urlöschen einer Steuerung hat folgende Auswirkungen auf die CPU: 1. Das Anwenderprogramm wird komplett gelöscht 2. Merkerzeiten- und Zähler werden gelöscht 3. Die Parameter der CPU werden auf die Grundeinstellungen zurückgesetzt. 4. Die MPI-Parameter bleiben erhalten, damit die CPU noch ansprechbar bleibt. 5. Diagnosepuffer, Echtzeituhr und Betriebsstundenzähler bleiben erhalten. 6. Besitzt die CPU einen Schacht für eine Memory-Card und ist ein Flash-Eprom mit darauf gespeichertem Programm gesteckt, so bootet die Steuerung mit diesen Daten.
Parameter	

5.5 PG_Compress

	Beschreibung
Position	public virtual bool PG_Compress()
Codeauszug C#	SPS.PG_Compress();
Beschreibung	Die Funktion „PG_Compress“ führt den Befehl „Komprimieren“ auf der verbundenen SPS aus. Komprimieren ordnet die Bausteine im Arbeitsspeicher lückenlos hintereinander an und entfernt alte, ungültige Bausteine. Dazu sollte die CPU im Betriebszustand STOPP stehen.
Parameter	

5.6 PG_ClockRead

	Beschreibung
Position	public virtual DateTime PG_ClockRead()
Codeauszug C#	DateTime spstime = SPS.PG_ClockRead(); DTP_Clock.Value = spstime;
Beschreibung	Die Funktion „PG_ClockRead“ liest die Systemzeit der verbundenen SPS aus.
Parameter	

5.7 PG_ClockWrite

	Beschreibung
Position	public virtual bool PG_ClockWrite(DateTime NewTime)
Codeauszug C#	SPS.PG_ClockWrite(DTP_Clock.Value);
Beschreibung	Die Funktion „PG_ClockWrite“ setzt die Systemzeit der verbundenen SPS.
Parameter	– NewTime : Zu schreibende Systemzeit.

5.8 PG_BlockList

	Beschreibung
Position	public virtual int[] PG_BlockList(S7BlockTypes BlockType)
Codeauszug C#	LB_Blocks.Items.Clear(); int[] Blocklist; Blocklist = SPS.PG_BlockList(S7Connector.BlockTypes.FC); foreach(int I in Blocklist) LB_Blocks.Items.Add("FC" + I.ToString()); Blocklist = SPS.PG_BlockList(S7Connector.BlockTypes.DB); foreach (int I in Blocklist) LB_Blocks.Items.Add("DB" + I.ToString()); Blocklist = SPS.PG_BlockList(S7Connector.BlockTypes.FB); foreach (int I in Blocklist) LB_Blocks.Items.Add("FB" + I.ToString()); Blocklist = SPS.PG_BlockList(S7Connector.BlockTypes.OB); foreach (int I in Blocklist) LB_Blocks.Items.Add("OB" + I.ToString());
Beschreibung	Die Funktion „PG_BlockList“ liest den Index der auf der verbundenen SPS Bausteine (DB, FC, SFC, FB,OB).
Parameter	– BlockType : Datentyp, dessen Index gelesen werden soll.

5.9 PG_BlockDelete

	Beschreibung
Position	public virtual byte[] PG_BlockDelete(S7BlockTypes BlockType, int BlockNo)
Codeauszug C#	try{ byte[] block_result; string test; string fdb; string fnr; int blocknr; S7Connector.BlockTypes blocktype; test = LB_Blocks.Text; fdb = test.Substring(0, 2); fnr = test.Substring(2); blocknr = Convert.ToInt32(fnr); blocktype = S7Connector.BlockTypes.FC; if (MessageBox.Show("Are You Sure?", "Block Delete", MessageBoxButtons.YesNo) == DialogResult.Yes){ block_result = SPS.PG_BlockDelete(blocktype, blocknr); LB_Blocks.Items.Clear(); } } catch (Exception E){ LB_Errors.Items.Add(E.ToString()); } }
Beschreibung	Die Funktion „PG_BlockDelete“ löscht einen Baustein (DB, FC, SFC, FB) auf der verbundenen SPS.
Parameter	– BlockType : Datentyp des zu löschenden Bausteins. – BlockNo : Adresse des zu löschenden Bausteins.

5.10 PG_BlockRead

	Beschreibung
Position	public virtual byte[] PG_BlockRead(S7BlockTypes BlockType, int BlockNo)
Codeauszug C#	try{ byte[] block_result; string test; string fdb; string fnr; int blocknr; S7Connector.BlockTypes blocktype; test = LB_Blocks.Text; fdb = test.Substring(0, 2); fnr = test.Substring(2); blocknr = Convert.ToInt32(fnr); blocktype = S7Connector.BlockTypes.DB;break; block_result = SPS.PG_BlockRead(blocktype, blocknr); LB_Blocks.Items.Clear(); for (int i = 0; i < (block_result as Array).Length; i++) LB_Blocks.Items.Add(block_result[i]); } catch (Exception E) { LB_Errors.Items.Add(E.ToString()); } }
Beschreibung	Die Funktion „PG_BlockRead“ liest einen Baustein (DB, FC, SFC, FB) von der verbundenen SPS. Rückgabe erfolgt im MCS7 Byte Code.
Parameter	– BlockType : Datentyp des zu lesenden Bausteins. – BlockNo : Adresse des zu lesenden Bausteins.

5.11 PG_BlockWrite

	Beschreibung
Position	public virtual byte[] PG_BlockWrite(byte[] MCS7Code)
Codeauszug C#	```try { byte[] block_result; byte[] mcs7code = new byte[] { 112, 112, 1, 1, 2, 12, 0, 1, 0, 0, 0, 92, 0, 0, 0, 0, 3, 144, 215, 128, 37, 65, 3, 143, 195, 174, 37, 65, 0, 8, 0, 6, 0, 0, 0, 6, 192, 0, 152, 200, 101, 0, 1, 1, 0, 0, 0, 0, 0, 5, 1, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 238, 149, 0, 0, 0, 0, 0, 0, 0, 0 }; if (MessageBox.Show("Are You Sure? Write FC1 : E0.0 = M200.0 ", "Block Delete", MessageBoxButtons.YesNo) == DialogResult.Yes) { block_result = SPS.PG_BlockWrite(mcs7code); LB_Blocks.Items.Clear(); } } catch (Exception E) { // Exception Handling }```
Beschreibung	Die Funktion „PG_BlockWrite“ schreibt einen Baustein (DB, FC, SFC, FB) auf die verbundene SPS.
Parameter	– MCS7Code : Code des zu schreibenden Baustein (im MCS7 Format).

5.12 PG_Passopen

	Beschreibung
Position	public virtual bool PG_Passopen(string password)
Codeauszug C#	try { string password = txt_passtext.Text; MessageBox.Show(Convert.ToString(SPS.PG_Passopen(password))); } catch (Exception E) { LB_Errors.Items.Add(E.ToString()); }
Beschreibung	Die Funktion „PG_Passopen“ Sendet das Verbindungspasswort für diese CPU. Die maximale Länge des zu vergebene Passwort in Step7 sind 8 Zeichen
Parameter	– password : Passwort als Ascii-Zeichen Floge

6 Fehlerbehandlung

6.1 Exception Handling

Komt es währen der Kommunikation oder durch die Kommunikation zu einem Programm Fehler so können diese mittels try{ }Code... catch (Exception E) abgefangen werden. Sollte es durch einen Fehler zu einem Verbindungsabbruch kommen oder ist der Fehler ein Verbindungsabbruch. So wird der S7Connector beim nächsten Lese oder Schreibversuch selbständig versuchen die Verbindung wieder aufzubauen.

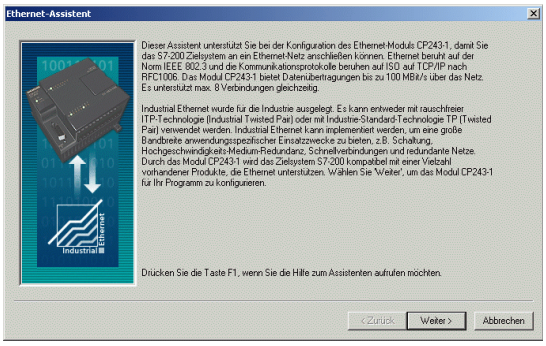
6.2 Error List

Error = "hardware fault";
Error = "object access not allowed: occurs when access to timer and counter data type is set to signed integer and not BCD";
Error = "Not context";
Error = "address out of range: occurs when requesting an address within a data block that does not exist or is out of range";
Error = "address out of range";
Error = "write data size mismatch";
Error = "object does not exist: occurs when trying to request a data block that does not exist";
Error = "communication link not available";
Error = "negative acknowledge / time out error";
Error = "data does not exist or is locked";
Error = "unknown error";
Error = "wrong interface specified";
Error = "too many interfaces";
Error = "interface already initialized";
Error = "interface already initialized with another connection";
Error = "interface not initialized; this may be due to an invalid MPI address (local or remote ID) or the PLC is not communicating on the MPI network";
Error = "can't set handle";
Error = "data segment isn't locked";
Error = "data field incorrect";
Error = "block size is too small";
Error = "block boundary exceeded";
Error = "wrong MPI baud rate selected";
Error = "highest MPI address is wrong";
Error = "address already exists";
Error = "not connected to MPI network";
Error = "-";
Error = "hardware error";
Error = "hardware error";
Error = "communication link unknown";
Error = "communication link not available";
Error = "MPI communication in progress";
Error = "MPI connection down; this may be due to an invalid MPI address (local or remote ID) or the PLC is not communicating on the MPI network";
Error = "interface Is busy";
Error = "not permitted in this mode";
Error = "hardware error";
Error = "access to object not permitted";
Error = "Not context";
Error = "address invalid. This may be due to a memory address that is not valid for the PLC";
Error = "data type not supported";
Error = "data type not consistent";
Error = "object doesn't exist. This may be due to a data block that doesn't exist in the PLC";
Error = "not enough memory on CPU";
Error = "maybe CPU already in RUN or already in STOP";

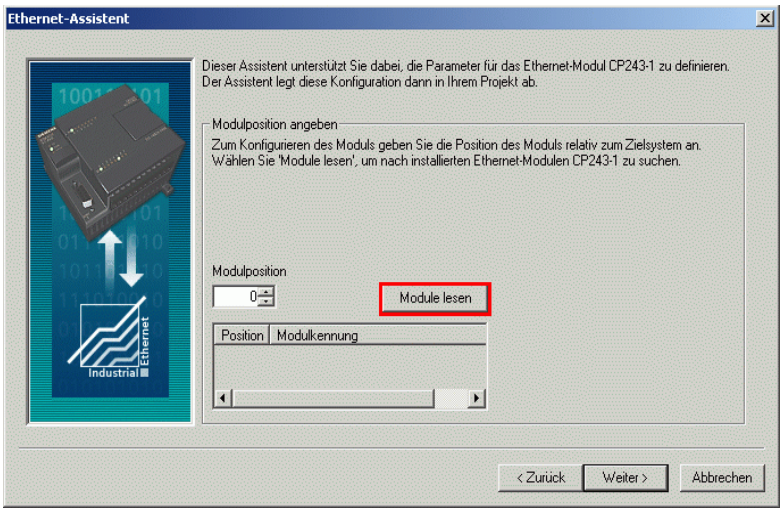
```
Error = "serious error";
Error = "wrong PDU (response data) size";
Error = "Not address";
Error = "Step7: variant of command is illegal.";
Error = "Step7: status for this command is illegal.";
Error = "Step7: function is not allowed in the current protection level.";
Error = "syntax error: block name";
Error = "syntax error: function parameter";
Error = "syntax error: block type";
Error = "no linked data block in CPU";
Error = "object already exists";
Error = "object already exists";
Error = "data block in EPROM";
Error = "block doesn't exist";
Error = "no block available";
Error = "block number too large";
Error = "coordination rules were violated";
Error = "protection level too low";
Error = "protection violation while processing F-blocks; F-blocks can only be processed after password
input";
Error = "invalid SSL ID";
Error = "invalid SSL index";
Error = "information doesn't exist";
Error = "diagnosis: DP Error";
Error = "this job does not exist";
Error = "maybe invalid BCD code or Invalid time format";
Error = "wrong ID2, cyclic job handle";
Error = "API function called with an invalid parameter";
Error = "timeout, check RS232 interface";
Error = "Lack of resources in driver or in the library // Ressourcenengpaß im Treiber oder in der Library";
Error = "Configuration error // Konfigurationsfehler";
Error = "Job not currently permitted // Auftrag zur Zeit nicht erlaubt";
Error = "Parameter error";
Error = "Device already/not yet open // Gerät bereits/noch nicht geöffnet.";
Error = "CP not reacting // CP reagiert nicht";
Error = "Error in firmware // Fehler in der Firmware";
Error = "Lack of memory for driver // Speicherengpaß im Treiber";
Error = "No message // Keine Nachricht vorhanden";
Error = "Error accessing application buffer // Fehler bei Zugriff auf Anwendungspuffer";
Error = "Timeout expired // Timeout abgelaufen";
Error = "Maximum number of logons exceeded // Die maximale Anzahl an Anmeldungen ist überschritten";
Error = "Job aborted // Der Auftrag wurde abgebrochen";
Error = "An auxiliary program could not be started // Ein Hilfsprogramm konnte nicht gestartet werden";
Error = "No authorization exists for this function // Keine Autorisierung für diese Funktion vorhanden";
Error = "Initialization not yet completed // Initialisierung noch nicht abgeschlossen";
Error = "Function not implemented // Funktion nicht implementiert";
Error = "CP name does not exist // CP-Name nicht vorhanden";
Error = "CP name does not exist // CP-Name nicht vorhanden";
Error = "CP name not configured // CP-Name nicht konfiguriert";
Error = "Channel name does not exist // Kanalname nicht vorhanden";
Error = "Channel name not configured // Kanalname nicht konfiguriert";
Error = "Error // Unbekanter Fehler "
```

7 S7 200 TCP-IP CP243-1 Anhang

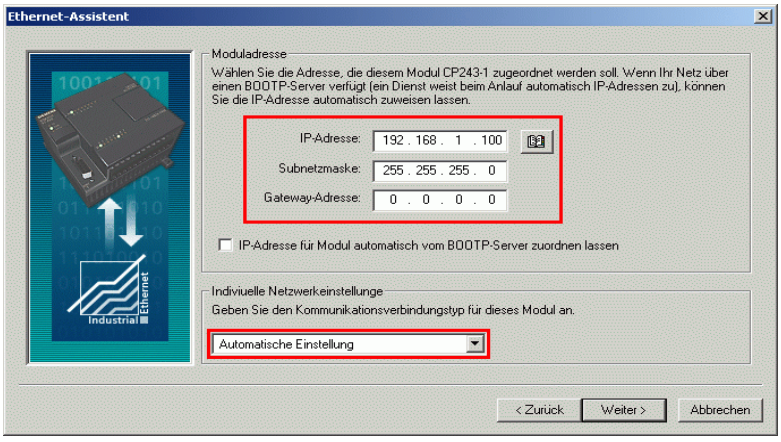
Die Einstellungen für den CP243-1 werden in STEP 7-Micro/WIN über den Ethernet-Assistenten vorgenommen. Starten Sie den Ethernet-Assistenten über "Extras > Ethernet-Assistent...".



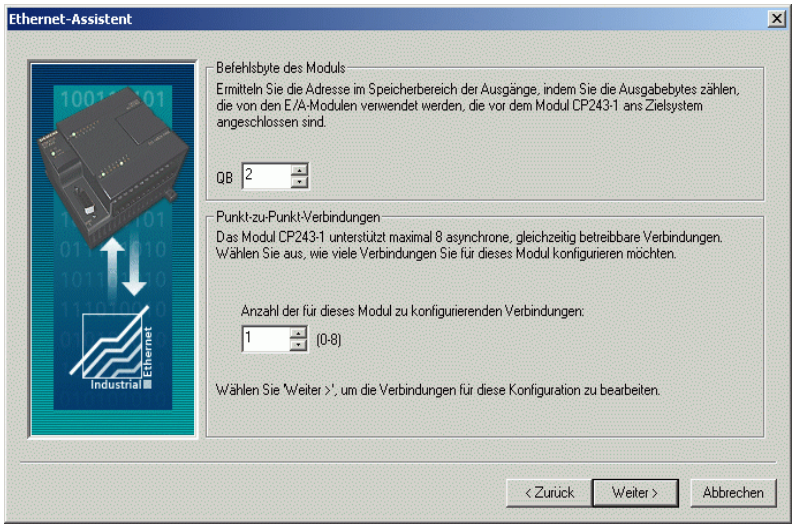
Der S7Connector kann nur eine Verbindung zu einem CP243-1 aufbauen, wenn das Modul auf der Position 0 projektiert ist. Der zugehörige TSAP muss daher in der letzten Stelle immer eine 0 enthalten (TSAP 02.00).



Legen Sie eine IP-Adresse für den CP 243-1 fest.



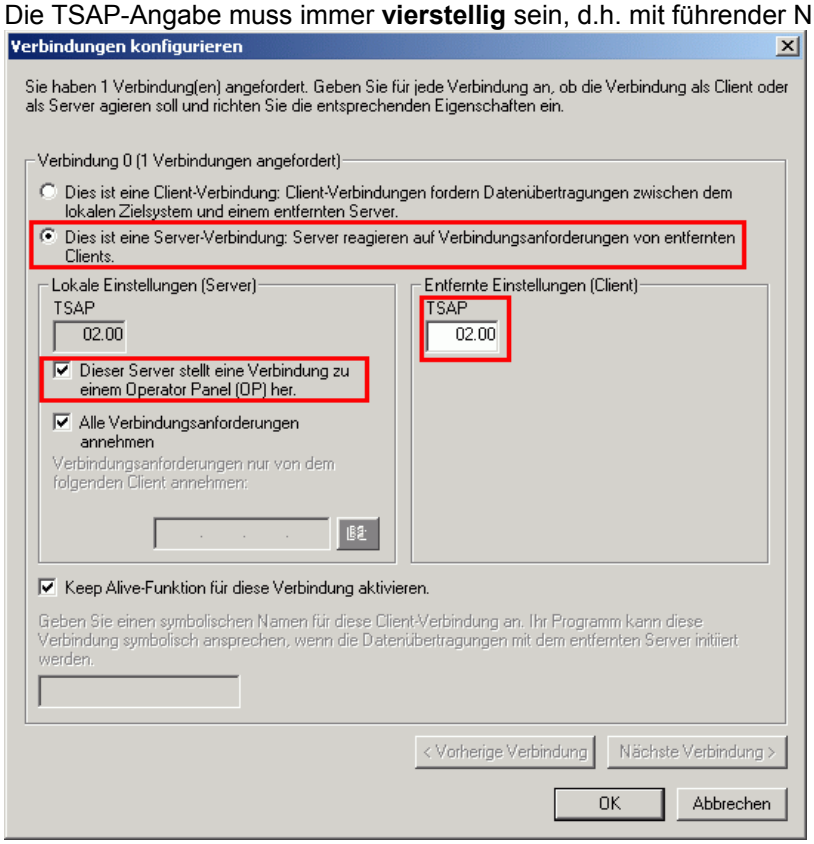
Geben Sie das Befehlsbyte des Moduls an und die Anzahl der Punkt-zu-Punkt-Verbindungen zum CP243-1. Klicken Sie auf "Weiter".



Die Konfiguration für eine Kopplung des CP243-1 mit S7Connector muss wie folgt übernommen werden. Bestätigen Sie bitte die Einträge mit **"OK"**.

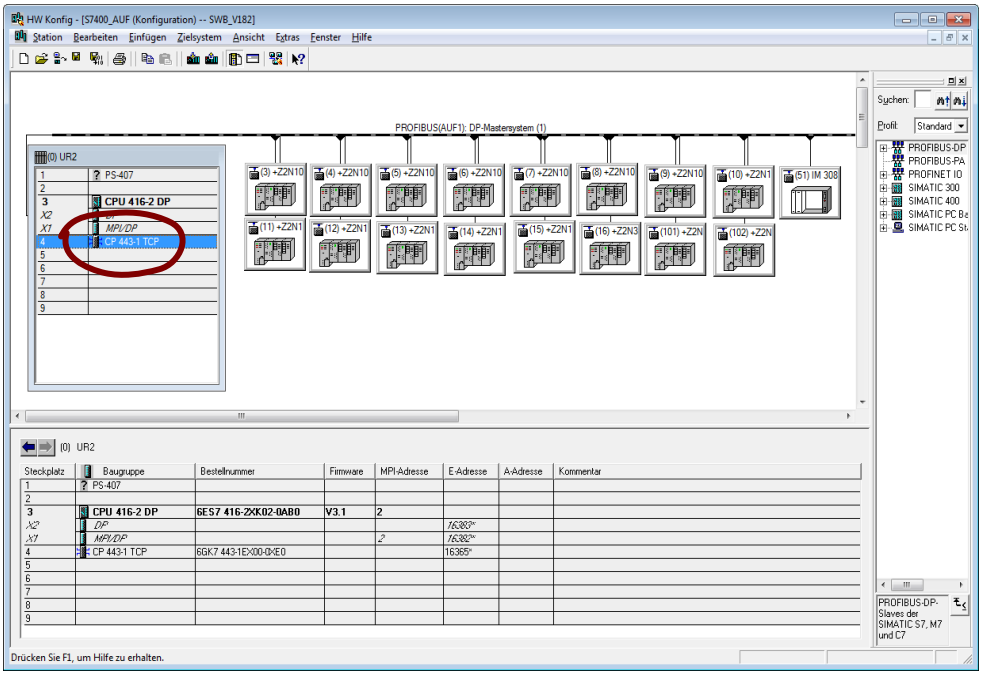
Achtung:

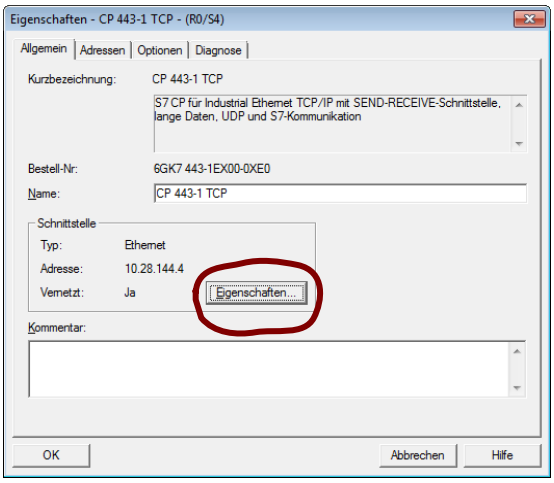
Die TSAP-Angabe muss immer **vierstellig** sein, d.h. mit führender Null (**0**2.00).

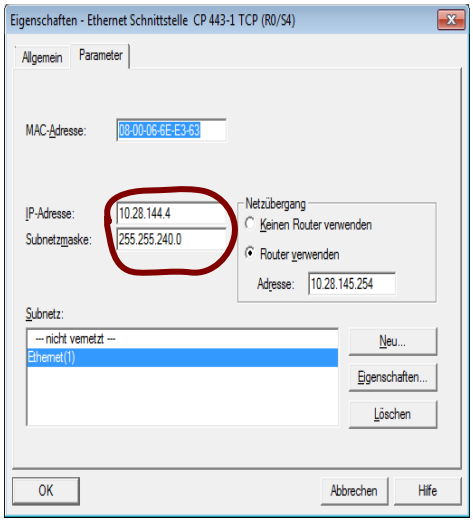


8 S7 300/400 TCP-IP Anahng

Beispiel Konfiguration einer S7 400 Komunikation über TCP/IP es sind keinen Spezielen Änderungen an der Haedwer nötig.Da der S7Connector sich verhält wie Step7 kann grundlegend davon ausgegängen werden das wenn Step 7 mit der Ziel CPU komunizieren kann. So kann das auch der S7Conector.







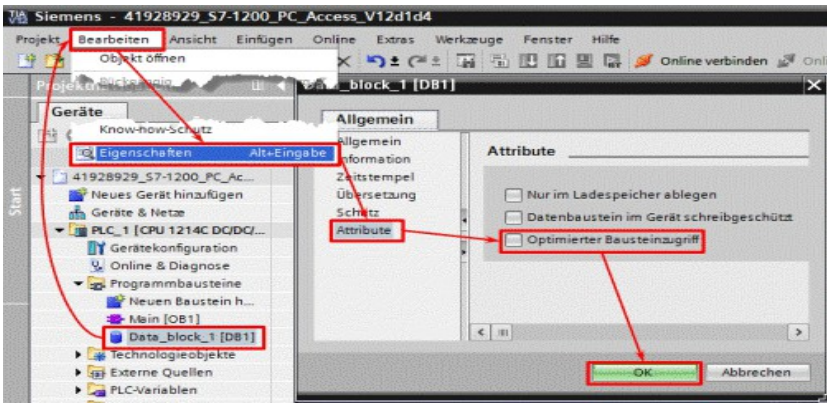
9 S7 1200/1500 TCP-IP Anahng

Der S7connector kann auf S71200/1500 CPU zugreifen, Aber er hat nur grundlegende Datenübertragungs Rechte. Alle PG-Operationen (Steuerung / Verzeichnis / etc ..) sind nicht Erlaubt. Für das Lesen/Schreiben von DB's sind Besondere Einstellungen im TIA Portal nötig

1. Es Kann nur auf globale DBs zugegriffen werden kann.
2. Die optimierte Blockzugriff muss ausgeschaltet sein.
3. Die Zugriffsebene muss "voll" sein und die "Verbindungsmechanismus" muss GET / PUT ermöglichen.

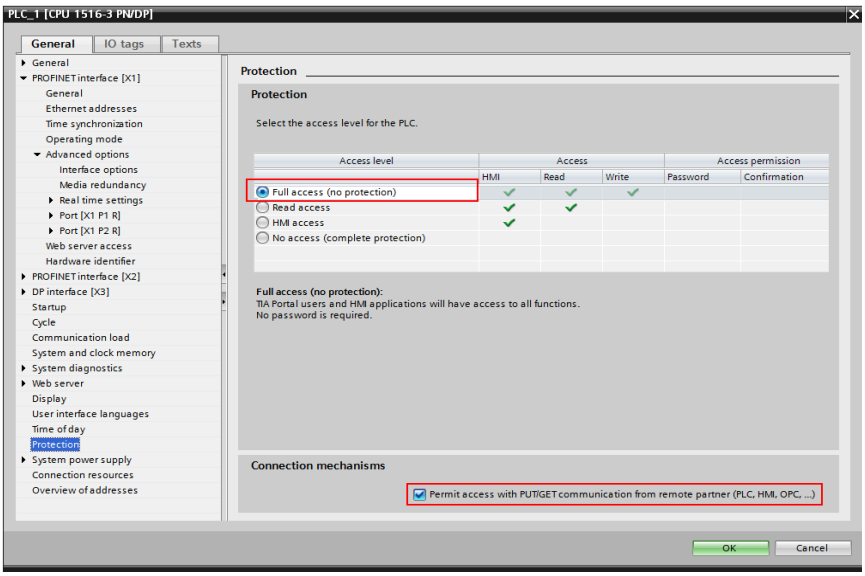
9.1 S71200/1500 DB Eigenschaften

Wählen Sie einen DB im linken Bereich unter "Programmbausteine", und drücken Sie Alt-Enter (oder im Kontextmenü wählen Sie "Eigenschaften ...")
Deaktivieren Sie die Option Optimierter Zugriff.



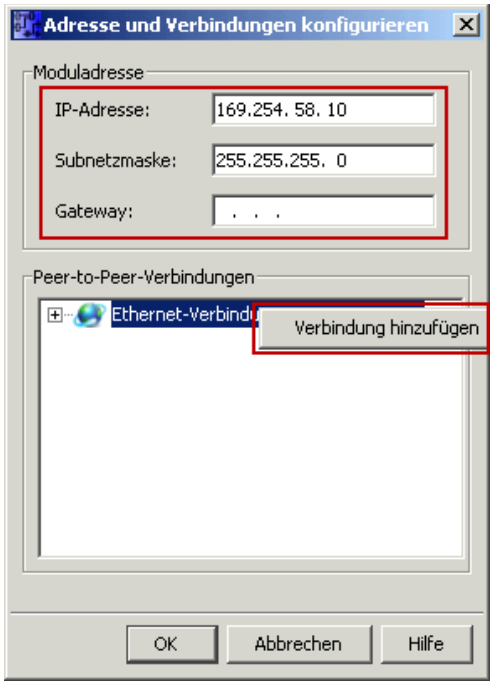
9.2 S71200/1500 Schutz

Wählen Sie das CPU-Projekt im linken Bereich und drücken Sie Alt-Enter (oder im Kontextmenü wählen Sie "Eigenschaften ...")
In der abschnidt Protection, wählen Sie "Vollzugriff" und aktivieren Sie "Zugriff erlauben mit PUT / GET ....", wie in der Abbildung.



10 S7 LOGO 0BA7 TCP-IP Anahng

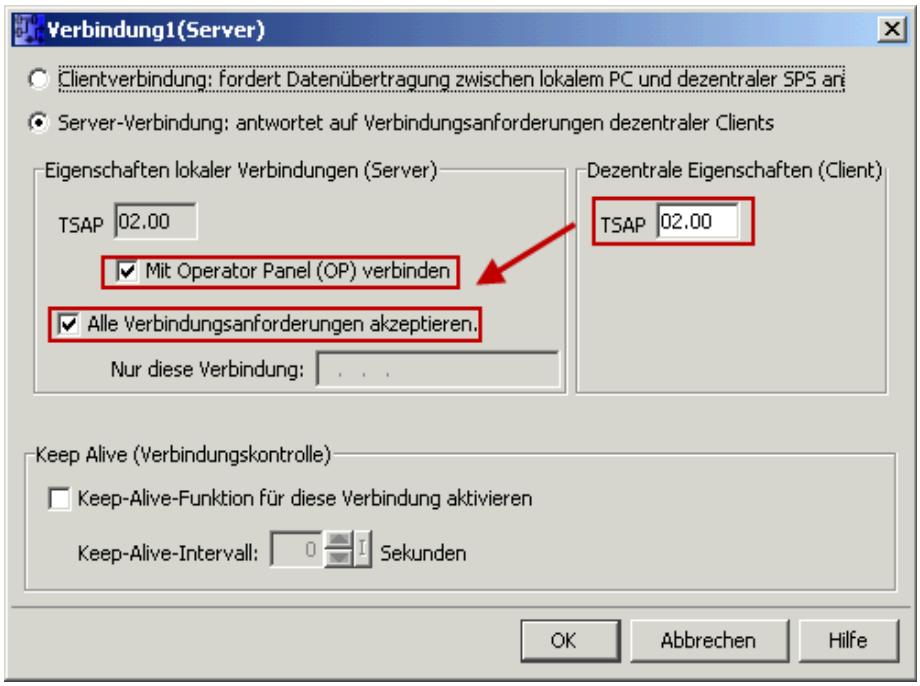
Öffnen Sie LOGO!Soft Comfort Gehen Sie in das Menü "Extras > *Ethernet-Verbindung*“



Klicken Sie mit der rechten Maustaste auf "Ethernet-Verbindungen" und auf "Verbindung hinzufügen"

Wählen Sie den Punkt "Server-Verbindung" aus, und geben Sie unter "TSAP" den Wert 02.00 ein.

Wählen Sie "Mit Operator-Panel verbinden" und "Alle Verbindungsanforderungen akzeptieren" aus.



Gehen Sie in das Menü "Extras > Übertragen > PC > LOGO!". Wählen Sie die IP-Adresse Ihrer LOGO! aus und bestätigen Sie mit "OK", damit das Programm in Ihr LOGO! 0BA7-Gerät geladen wird.