



## Ethernet xx7 Serie

Controls Division

<b>0</b>	<b>Inhaltsverzeichnis</b>	
0.1	Dokument-Historie .....	0-4
0.2	Warenzeichen .....	0-4
<b>1</b>	<b>Einleitung</b>	
1.1	Kommunikationsarten .....	1-1
1.2	Die Ethernet Kommunikationsdienste der PCD-Familien .....	1-3
1.3	Betrieb über projektierte oder programmierte Datenbasis .....	1-3
1.4	Zusatzinformationen .....	1-4
<b>2</b>	<b>Hardwarebeschreibung und Konfiguration</b>	
2.1	Ethernet TCP/IP-Modul PCD7.F655 .....	2-1
2.1.1	Layout PCD7.F655 .....	2-1
2.1.2	LED-Funktionen .....	2-1
2.1.3	PGND-Verbindung .....	2-2
2.1.4	Verkabelung .....	2-2
2.2	PCD2.M5547 .....	2-3
2.2.1	Layout PCD2.M5547 .....	2-3
2.3	PCD3 Ethernet.....	2-4
2.3.2	Layout PCD3.M5547.....	2-4
2.3.3	Layout PCD3.M6347.....	2-4
2.3.4	Layout PCD3.M6547.....	2-5
2.3.5	LED-Funktion .....	2-5
2.4	Konfigurieren der Ethernet Hardware .....	2-6
2.4.1	Allgemeine Hinweise.....	2-6
2.4.2	Ethernet-Subnetz anlegen .....	2-6
2.4.3	Ethernet-CP in die Hardwarekonfiguration eintragen .....	2-7
2.4.4	Weitere CP- Eigenschaften einstellen .....	2-10
<b>3</b>	<b>Kommunikationsverbindungen projektieren/programmieren</b>	
3.1	Allgemeines .....	3-1
3.1.1	Datenvolumen und Mengengerüst.....	3-1
3.1.2	Neue Verbindung erzeugen .....	3-2
3.2	ISO-on-TCP Verbindungseigenschaften projektieren .....	3-5
3.2.1	Den lokalen Verbindungsendpunkt festlegen.....	3-6
3.2.2	ISO-on-TCP Adressen festlegen.....	3-8
3.2.3	ISO-on-TCP Optionen.....	3-10
3.3	TCP Verbindungseigenschaften projektieren.....	3-11
3.3.1	Den lokalen Verbindungsendpunkt festlegen.....	3-12
3.3.2	TCP Adressen festlegen .....	3-13
3.3.3	TCP- Optionen .....	3-16
3.4	UDP Verbindungseigenschaften projektieren .....	3-17
3.4.1	Den lokalen Verbindungsendpunkt festlegen.....	3-18
3.4.2	UDP Adressen festlegen .....	3-19
3.4.3	UDP mit Broadcast und Multicast .....	3-21
3.4.4	Freie UDP-Verbindung .....	3-23
3.5	Betriebsart FETCH/WRITE .....	3-24
3.6	ISO-on-TCP Verbindung programmieren.....	3-25
3.7	TCP oder UDP Verbindung programmieren .....	3-26

<b>4</b>	<b>SEND/RECEIVE- Schnittstelle im Anwenderprogramm</b>	
4.1	Arbeitsweise der SEND/RECEIVE-Schnittstelle in der CPU .....	4-1
4.2	SEND/RECEIVE-Schnittstelle programmieren .....	4-2
4.3	Allgemeine Hinweise zu den FCs / FBs .....	4-3
4.4	FC-Aufrufe parametrieren .....	4-3
4.4.1	Parameter zur CP- und Verbindungszuordnung (Eingangsparameter) .....	4-3
4.4.2	Parameter zur Angabe eines CPU-Datenbereichs (Eingangsparameter).....	4-3
4.4.3	Statusinformationen (Ausgangsparameter) .....	4-3
4.5	FCs für die SEND/RECEIVE-Schnittstelle .....	4-4
4.5.1	Datenbereich für UDP Verbindungen.....	4-4
4.5.2	FC5 AG_SEND / FC50 AG_LSEND .....	4-5
4.5.3	FC6 AG_RECV / FC60 AG_LRECV .....	4-10
<b>5</b>	<b>IP-Konfiguration Programmieren</b>	
5.1	Übersicht.....	5-1
5.2	Konfigurations-Datenbaustein.....	5-2
5.2.1	Parameterblock für Systemdaten (IP- Konfiguration) .....	5-3
5.2.2	Parameterblöcke für Verbindungstypen.....	5-6
5.3	FB55 IP_CONFIG für programmierte Kommunikationskonfiguration .....	5-7
5.4	DHCP / DNS Konfiguration .....	5-10
5.4.1	DNSQueryByName.....	5-10
5.4.2	DNSQueryByAddr.....	5-11
5.5	Anpassen IP- Konfiguration über WEB- CGI .....	5-12
5.5.1	Lesen der fixen IP- Konfiguration / Status .....	5-13
5.5.2	Lesen der DNS – Konfiguration / Status Information .....	5-14
5.5.3	Lesen der DHCP Status Information.....	5-14
5.5.4	Setzen der IP- Konfiguration.....	5-15
5.5.5	Setzen der DNS – Konfiguration.....	5-15
5.5.6	Setzen der DHCP – Konfiguration .....	5-16
5.6	Konfigurationsfile / CDB.....	5-17
5.6.1	Konfigurationsfile .....	5-17
5.6.2	CDB (Configuration Data Block) .....	5-17
5.6.3	Konfigurationsparameter.....	5-17
<b>6</b>	<b>Weitere Kommunikationsdienste</b>	
6.1	PG/OP Kommunikation .....	6-1
6.1.1	Fernprogrammierung über Industrial Ethernet.....	6-1
6.2	S7 Kommunikation (Server).....	6-2
6.3	Dateiverwaltung und Dateizugriff über FTP .....	6-2
6.3.1	FTP im Produktionsumfeld.....	6-2
6.3.2	Kommunikation einer SPS über FTP .....	6-3
6.4	HTTP-Direkt / Webserver.....	6-4
6.5	Virtuelle Private Netzwerke (VPN).....	6-7
6.6	SBC S-Bus-Kommunikation .....	6-9
6.6.1	Unterstützte Telegramme .....	6-9
6.6.2	Media Mapping .....	6-10

<b>7</b>	<b>Email</b>	
7.1	Einleitung .....	7-1
7.2	System Funktion SFB SendMail .....	7-3
7.3	Konfigurationen.....	7-4
7.3.1	MTA Konfiguration.....	7-4
7.3.2	Authentication .....	7-4
7.4	Mail Inhalt.....	7-5
<b>8</b>	<b>SBC Modbus</b>	
8.1	Einleitung .....	8-1
8.1.1	SBC Modbus Server .....	8-1
8.1.2	SBC Modbus Client.....	8-1
8.1.3	SBC Modbus Limitierungen .....	8-2
8.2	Modbus Medien und xx7 Medien .....	8-2
8.3	System Funktionen für Modbus Client .....	8-4
8.3.1	OpenIPChannel (FB490) .....	8-4
8.3.2	SendReadReq (FB492) .....	8-5
8.3.3	SendWriteReq (FB493).....	8-7
8.4	System Funktionen für Modbus Server.....	8-10
8.4.1	InitServer (FB500).....	8-10
8.4.2	InitUID (FB502).....	8-10
8.4.3	InitMap (FB503) .....	8-11
8.5	System Funktionen für Modbus Diagnose .....	8-13
8.5.1	Diagnose Register .....	8-13
8.5.2	GetUID_Diag (FB504).....	8-14
8.5.3	GetServer_Diag (FB505) .....	8-14
8.6	Fehler Meldungen .....	8-15
8.6.1	Fehler Rückgabe von FB Aufrufen.....	8-15
8.6.2	Modbus Shell Fehler Meldungen .....	8-16
8.6.3	Modbus Treiber Fehler Meldungen .....	8-17
8.6.4	Ausnahme Fehler Meldungen (vom Server gesendet) .....	8-17
<b>A</b>	<b>Anhang</b>	
A.1	Icons .....	A-1
A.2	Adresse.....	A-2

<b>0.1 Dokument-Historie</b>			
Dokument-Nr.	Ausgabe	Publikation	Bemerkungen
26/791	pD1	2005-04-30	Publizierte Ausgabe
26/791	pD2	2008-07-21	Adressen und Logo angepasst
26/791	pDE3	2009-05-20	- Kapitel E-Mail neu in separatem Dokument. - PCD7.F650 ersetzt durch PCD7.F655 - neue CPUs ergänzt
26/791	DE04	2009-10-27	- Änderungen in Kap. 6 und 7 - Kap. 8 Modbus neu
	DE05	2014-03-24	Logoänderung

## 0.2 Warenzeichen

Saia PCD® ist ein eingetragenes Warenzeichen der Saia-Burgess Controls AG

STEP®, SIMATIC®, S7-300®, S7-400®, und Siemens® sind eingetragene Warenzeichen der Siemens AG

Technische Anpassungen und Änderungen unterliegen dem Stand der Technik

Saia-Burgess Controls AG

© Alle Rechte vorbehalten

# 1 Einleitung

Die Steuerungen der PCD Serie xx7 bieten eine Reihe von Kommunikationsdiensten über Ethernet. Das Spektrum erstreckt sich hierbei von allgemein gültigen Standardprotokollen, die bei der S5-kompatiblen Kommunikation zu finden sind, bis hin zur komfortablen S7- Kommunikation. Dabei kommen sowohl Protokolle wie TCP/IP und UDP als auch spezielle Protokolle zum Einsatz.



Die Projektierung und Programmierung der Ethernet Kommunikationsdiensten erfolgt mit Hilfe der Step<sup>®</sup>7 Tools von Siemens.

## 1.1 Kommunikationsarten

Die Saia PCDs der xx7 Serie unterstützen je nach PCD-Typ folgende Kommunikationsarten:



mögliche Kommunikationsarten	Schnittstellen/ Dienste/ Protokolle
PG/OP- Kommunikation S7 Kommunikation	über das Protokoll - TCP/IP (RFC 1006) (nur Server)
S5-kompatible -Kommunikation	mit SEND/RECEIVE Schnittstelle über die Protokolle - ISO-on-TCP (TCP/IP mit RFC1006) - TCP - UDP - E-mail
HTML- Prozesskontrolle über Web Browser	über die Protokolle - HTTP /IP-Protokoll
Dateiverwaltung und Dateizugriff über FTP	über die Protokolle - FTP /IP-Protokoll (nur Server)
SBC-kompatible Kommunikation	über Protokolle S-Bus
Modbus	über Modbus Protokolle (sowohl Master wie Slave)
IP-Dienste DHCP, DNS	Protokolle

Die Kommunikationsarten sind abhängig von der verwendeten PCD Steuerung.

- **PG/OP**

Die PG/OP-Kommunikation dient zum Laden von Programmen und Konfigurationsdaten, zum Durchführen von Test- und Diagnosefunktionen sowie zum Bedienen und Beobachten einer Anlage über OPs / Visualisierungssysteme.

### • **S7- Kommunikation**

Die S7 Kommunikation bildet eine einfache und effiziente Schnittstelle zwischen PCD- Steuerungen und SIMATIC S7-Stationen oder zu PG/PC. Die PCD-Steuerungen können nur als S7-Server betrieben werden.

### • **S5- kompatible Kommunikation mit SEND/RECEIVE Schnittstelle**

Die SEND/RECEIVE-Schnittstelle ermöglicht je nach PCD-Typ die programmgesteuerte Kommunikation über eine projektierte Verbindung von PCD xx7 zu PCD xx7, zu PCD Classic, zu SIMATIC S7/S5, zu PC/PG und zu beliebigen Fremdstationen.

Je nach PCD-Typ stehen an der SEND/RECEIVE-Schnittstelle folgende Kommunikationsdienste zur Verfügung:

- TCP/IP für die netzwerkübergreifende Kommunikation mit
  - ISO-on-TCP-Verbindungen (RFC 1006),
  - TCP-Verbindungen und
  - UDP-Datagrammdienst (einschliesslich Broadcast).
- E-Mail versenden  
Die Steuerung wird in die Lage versetzt, abhängig von Prozessereignissen Nachrichten zu versenden.

### • **S5- kompatible Kommunikation mit FETCH/WRITE-Diensten (Server)**

Die FETCH/WRITE-Dienste (Server) ermöglichen den direkten Zugriff auf System-speicherbereiche in der PCD xx7 von SIMATIC S5, SIMATIC PC-Stationen oder von Fremdgeräten.

Für FETCH/WRITE-Zugriffe stehen folgende Kommunikationsdienste zur Verfügung:

- TCP-Verbindungen
- ISO-on-TCP-Verbindungen

### • **HTTP-Direct**

Mit den PCD3.Mxxx7 und PCD2.M5xx7 Systemen kann über Ethernet direkt auf den integrierten Web Server zugegriffen werden.

### • **Dateizugriff über FTP**

Bei den PCD3.Mxxx7 und PCD2.M5xx7 Systemen stehen zusätzliche Funktionen für FTP-Dienste zur Verfügung.

Die PCD Station kann im FTP Server Betrieb angesprochen werden. Eine andere Station, beispielsweise ein PC, übermittelt schreibend oder lesend Datenbausteine in die PCD oder Dateien in das Filesystem auf der PCD.

### • **SBC Kommunikation**

Über das S-Bus Protokoll können von einer Classic PCD Medien gelesen und geschrieben werden. Die xx7 Steuerungen können nur als S-Bus Server betrieben werden.

### • **Modbus**

Die PCD können sowohl als Modbus Master wie als Modbus - Slave betrieben werden. Die PCD stellt verschiedene System Funktionen für die Modbus- Kommunikati-

on zur Verfügung.

• **IP-Dienste**

Die PCD Systeme unterstützen die IP-Protokoll Dienste DHCP (Dynamic Host Configuration Protocol, und DNS (Domain Name System). Über Systemfunktionen können diese Dienste konfiguriert werden.



**1.2 Die Ethernet Kommunikationsdienste der PCD-Familien**

Je nach Ethernet-Karte unterstützen die xx7 PCDs folgende Kommunikationsmöglichkeiten:

PCD System		unterstützte Funktionen								
		PG/OP	S7-Server	S5-kompatibel	HTTP-Direkt	FTP-Server	E-mail	IP-Dienste	Modbus	S-Bus
	PCD1.M137 PCD2.M157 PCD2.M177		•	• <sup>1)</sup>						
	PCD2.M487	•	•	•						
	PCD2.M5xx7	•	•	•	•	•	•	•	•	•
	PCD3.Mxxx7	•	•	•	•	•	•	•	•	•

<sup>1)</sup> ohne S5-FETCH/WRITE

**1.3 Betrieb über projektierte oder programmierte Datenbasis**

**Projektierung**

Für den Anschluss und die Projektierung der Ethernet-Karten ist die Projektiersoftware STEP<sup>®</sup>7 und die Option SIMATIC NET NCM S7 erforderlich.

NCM S7 wird als STEP<sup>®</sup>7 Option automatisch installiert und ist damit in STEP<sup>®</sup>7 integriert.

**Programmierte Konfiguration**

Es gibt Anwendungsbereiche, bei denen es vorteilhaft ist, die Konfiguration nicht über die Projektierungsschnittstelle von STEP<sup>®</sup>7 sondern über spezifische Applikationen programmgesteuert einzurichten.

Für diese Anwendungen steht ab STEP7 V5.2 SP1 ein Funktionsbaustein zur Verfügung, der die flexible Übergabe von Datenbausteinen mit Projektierungsdaten an die Ethernet-Karte ermöglicht.

## 1.4 Zusatzinformationen

### Einsatzzweck für TCP-Verbindungen

1

TCP-Verbindungen sollten in erster Linie zur Ankopplung von Fremdsystemen verwendet werden, wenn diese den Protokollzusatz RFC1006 nicht unterstützen. Für die Kommunikation zwischen Geräten der SBC xx7 Familie sollten Sie ISO-on-TCP-Verbindungen verwenden, denn diese sind komfortabler einzusetzen!

Bei TCP-Verbindungen gibt es im Protokoll keine Informationen über das Ende einer Nachricht bzw. den Anfang einer neuen Nachricht. Daher muss die Empfängerstation wissen, wieviele Bytes zu einer Nachricht gehören und demnach einen exakt dieser Länge entsprechenden ANY-Pointer beim Aufruf des FC AG\_RECV/AG\_LRECV übergeben. (Beispiel: Sollen ständig 100 Byte Daten empfangen werden, könnte der ANY-Pointer folgendermassen aussehen: P#DB100.DBX 0.0 Byte 100) Wenn Daten mit variabler Länge ausgetauscht werden sollen, empfiehlt sich folgendes Vorgehen:

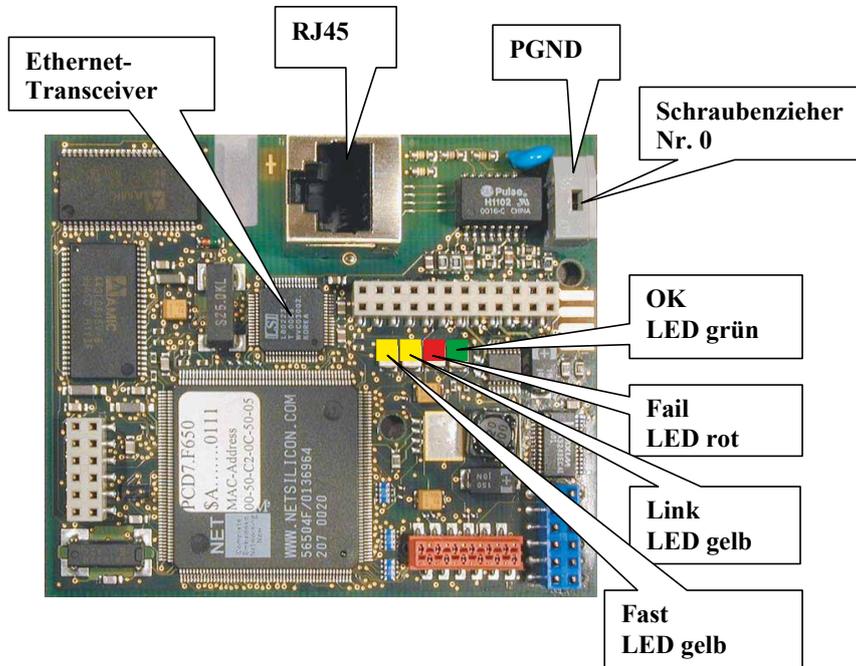
Vor den eigentlichen Nutzdaten im Telegramm wird eine Information über die Länge der Nutzdaten eingefügt. Die Empfängerstation wertet zunächst nur die Längeninformation aus. In einem weiteren Empfangsauftrag wird die entsprechende Nutzdatenumenge abgeholt, indem ein ANY-Pointer entsprechender Länge an der FC-Schnittstelle zum Abholen der eigentlichen Nutzdaten mitgegeben wird.

## 2 Hardwarebeschreibung und Konfiguration

### 2.1 Ethernet TCP/IP-Modul PCD7.F655

#### 2.1.1 Layout PCD7.F655

2



#### 2.1.2 LED-Funktionen

- Fast (gelb)** Diese LED leuchtet, wenn eine schnelle 100 Mbit/s-Verbindung erkannt wird, und ist dunkel bei 10 Mbit/s.
- Link (gelb)** Diese LED leuchtet, wenn eine Verbindung erkannt wird und blinkt bei Ethernet-Verkehr (aus bei Aktivität, ein bei keiner Aktivität).
- Fail (rot)** Diese LED blinkt, wenn ein Hardware- oder Firmware-Fehler erkannt wurde.
- OK (grün)** Diese LED zeigt den fehlerfreien Betrieb an (blinkt mit einer Frequenz von 2 Hz).



Die gelben LEDs haben während der Boot-Sequenz des Prozessors eine andere Bedeutungen als im Normalbetrieb. Unmittelbar nach einem Reset (beim Einschalten) zeigt die Fast LED eine Verbindung mit 10 Mbit/s an, während die Link LED eine 100-Mbit/s-Verbindung signalisiert.

Die Fail LED (rot) zeigt an, dass ein Fehler erkannt wurde. Die Blinkfrequenz liefert Information über den erkannten Fehler. Die OK LED kann ebenfalls blinken. Abhängig vom erkannten Fehler können die grünen und roten LEDs blinken. Wenn die grüne LED alleine blinkt, ist kein Fehler vorhanden.



Der Ethernet-Transceiver wird ziemlich heiss. Ohne Verbindung liegt seine Temperatur etwa 30° über der Umgebungstemperatur. Dies ist normal. Bitte nicht die Finger daran verbrennen!

### 2.1.3 PGND-Verbindung

Für die Einhaltung der EMV-Werte muss PGND (Schutzerde) immer mit der Chassis-Erde verbunden sein

### 2.1.4 Verkabelung

#### Kabeltyp

Dieses Modul ist für den Einsatz mit abgeschirmtem und nicht abgeschirmtem 100-Ohm-Kabel (UTP oder STP) ausgelegt. Bei elektromagnetischen Störungen ist abgeschirmtes Kabel für einen höheren Durchsatz sehr zu empfehlen. Wenn eine 100-Mbit/s-Verbindung benutzt wird, ist Kabeltyp CAT 5 die Minimalanforderung.

Für eine 10-Mbit/s-Verbindung muss mindestens der Kabeltyp CAT3 verwendet werden.



Dieses Modul benutzt ein Autonegotiationsverfahren zur Bestimmung der Übertragungsgeschwindigkeit. Um die Verbindung mit 100 Mbit/s aufzubauen, müssen beide Enden die Autonegotiation unterstützen. Ist dies nicht der Fall, wird das Modul die Verbindung mit 10 Mbit/s herstellen. Das Autonegotiationsverfahren erkennt den verwendeten Kabeltyp jedoch nicht. Das bedeutet, dass Sie die Verbindung zwar mit einem CAT3-Kabel mit 100 Mbit/s aufbauen können, aber eine solche Verbindung eventuell nicht zuverlässig arbeiten wird (dies kann zum Beispiel mit einem Überkreuzungskabel auftreten).

Es gibt keine Möglichkeit, PCD7.F65x zu zwingen, nur mit 10 Mbit/s zu arbeiten.

#### Verdrahtung

Die Verkabelung kann entweder Peer-to-Peer mit einem Überkreuzungskabel oder mit einem normalen Patchkabel via einen Hub oder einen Switch erfolgen.

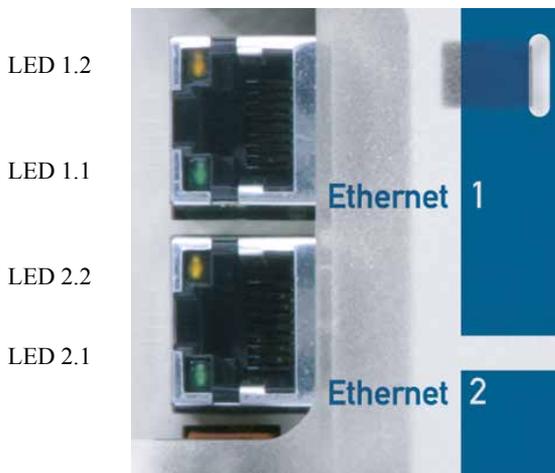
**2.2 PCD2.M5547**

Die PCD2.M5547 werden mit einem integrierten Ethernet Stack ausgeliefert. Für diese PCD2 Steuerungen ist deshalb keine externe Ethernet Karte notwendig. Die Konfiguration wird wie bei der PCD7.F655 Karte mithilfe der SIMATIC Software vorgenommen.

**2.2.1 Layout PCD2.M5547**



Detail:

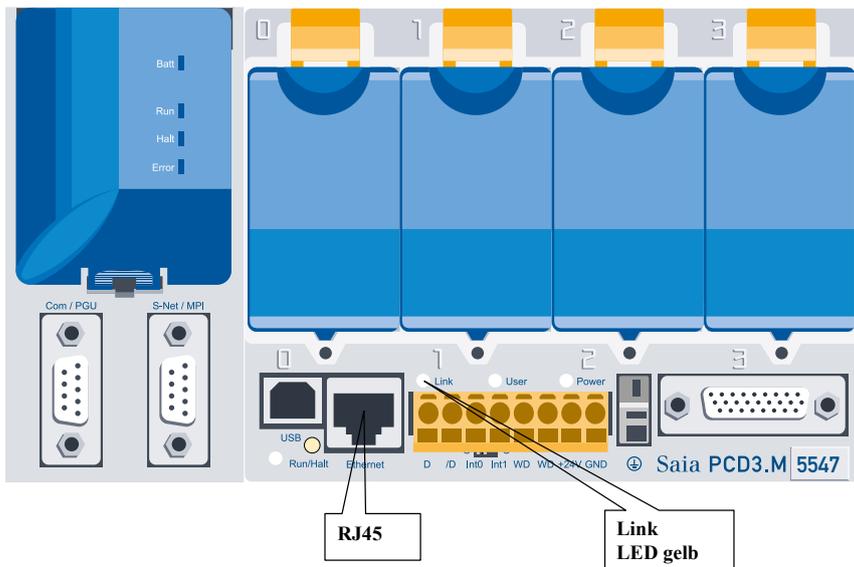


Ethernet 1 2	LED Funktion	LED Farbe	On	Off	Blinked
LED 1.1	Aktivitätsart (actif)	Grün	Keine Aktivität	Keine Aktivität	Netzwerk-Aktivität
	Verbindung (link)	Grün	Verbunden	Keine Verbindung	Aktiv
LED 1.2	Übertragungsgeschwindigkeit (data speed)	Grün	100 mb/s	10 mb/s	-
	Übertragungsgeschwindigkeit (data speed)	Orange	100 mb/s	10 mb/s	-

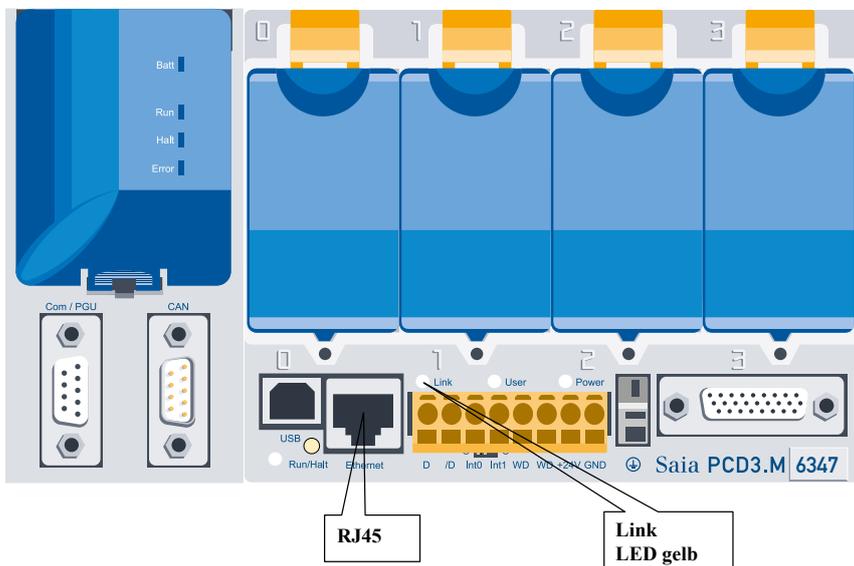
**2.3 PCD3 Ethernet**

Die PCD3.Mxxx7 Systeme werden mit einem integrierten Ethernet Stack ausgeliefert. Für diese PCD3 Steuerungen ist deshalb keine externe Ethernet Karte notwendig. Die Konfiguration wird wie bei der PCD7.F655 Karte mithilfe der SIMATIC Software vorgenommen.

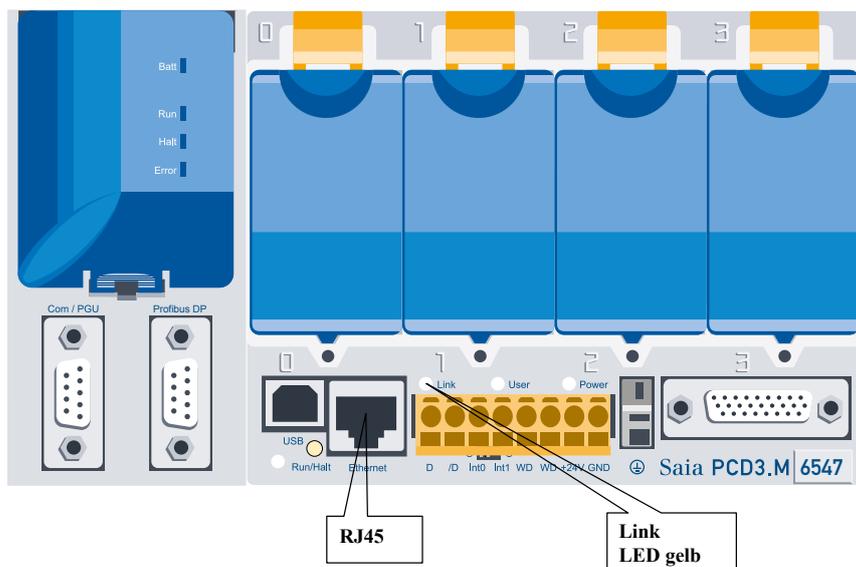
**2.3.2 Layout PCD3.M5547**



**2.3.3 Layout PCD3.M6347**



### 2.3.4 Layout PCD3.M6547



2

### 2.3.5 LED-Funktion

**Link (gelb)** Diese LED leuchtet, wenn eine Verbindung erkannt wird und blinkt bei Ethernet-Verkehr (aus bei Aktivität, ein bei keiner Aktivität).



Die PCD3 benutzt ein Autonegotiationsverfahren zur Bestimmung der Übertragungsgeschwindigkeit. Um die Verbindung mit 100 Mbit/s aufzubauen, müssen beide Enden die Autonegotiation unterstützen. Ist dies nicht der Fall, wird das Modul die Verbindung mit 10 Mbit/s herstellen. Das Autonegotiationsverfahren erkennt den verwendeten Kabeltyp jedoch nicht. Das bedeutet, dass Sie die Verbindung zwar mit einem CAT3-Kabel mit 100 Mbit/s aufbauen können, aber eine solche Verbindung eventuell nicht zuverlässig arbeiten wird (dies kann zum Beispiel mit einem Überkreuzungskabel auftreten).

Es gibt keine Möglichkeit, die PCD3 zu zwingen, nur mit 10 Mbit/s zu arbeiten.

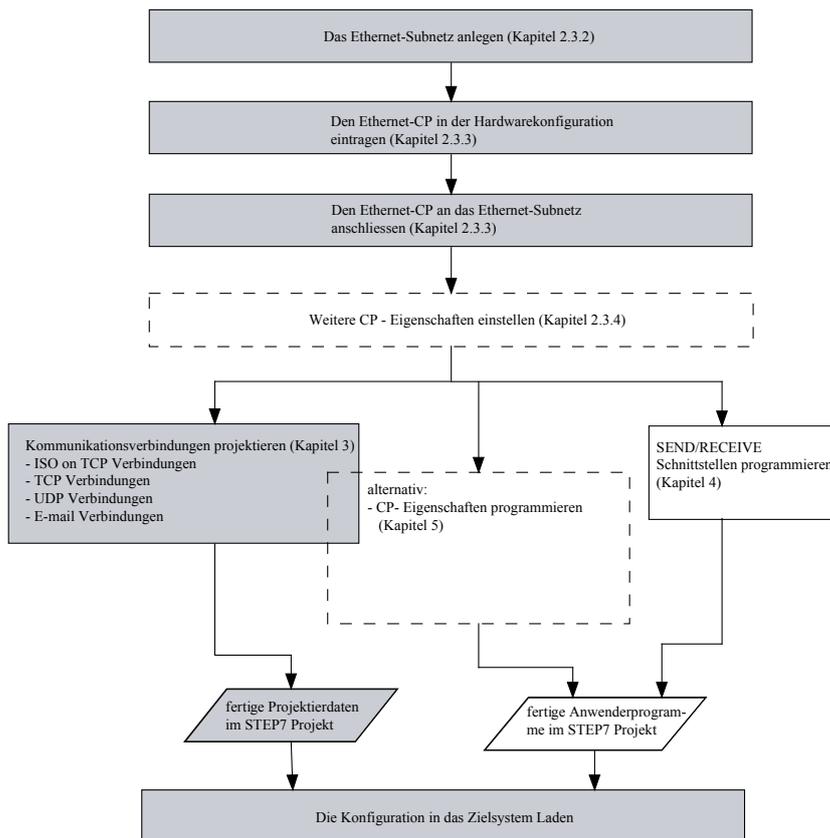
## 2.4 Konfigurieren der Ethernet Hardware

### 2.4.1 Allgemeine Hinweise

Um eine Saia PCD an Ethernet anzuschliessen, konfigurieren Sie einen Siemens CP (CP443-1) im Hardware Konfigurator des SIMATIC Managers. Der Siemens CP wird wie jede andere Baugruppe bei SIMATIC S7 in einem STEP®7-Projekt verwaltet. Mit STEP®7 wird die Hardware konfiguriert und die Anwendersoftware erstellt und verwaltet.

Damit die Saia PCD die Ethernet Konfiguration erkennt muss ein 400er CP konfiguriert werden. Eine Konfiguration mit 300er CP's wird nicht korrekt ausgewertet, somit werden die Ethernetdienste in der PCD nicht gestartet.

Alternativ können bei PCD Systemen mit integriertem Ethernet Stack, die CP-Konfiguration während der Programm ausführung geändert werden. Die PCD stellt spezielle Systemfunktionen sowie WEB-CGI Befehle zur Verfügung (Siehe Kapitel 5). Falls die PCD Systeme in einer IP- Umgebung eingebunden werden, können Sie so konfiguriert werden, dass die PCD über DHCP Ihre IP-Adresse beim Aufstarten dynamisch beziehen kann. (Siehe Kapitel 9). Um einen CP zu konfigurieren, führen Sie folgende Schritte aus (die gestrichelten Linien sind Optionen):



### 2.4.2 Ethernet-Subnetz anlegen

Es ist zweckmässig, das Subnetz vor der Konfiguration der Stationen anzulegen, da die Zuordnung der SIMATIC-Stationen dann weitgehend automatisch erfolgen kann.

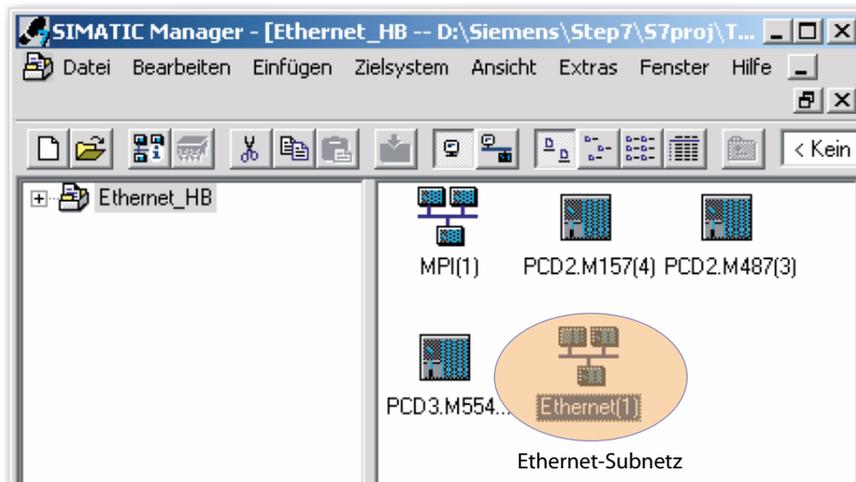
Es ist auch möglich, während der Konfiguration eines CP das Subnetz nachträglich anzulegen.

Gehen Sie bitte wie folgt vor:

1. Wählen Sie im SIMATIC Manager das Projekt an.
2. Wählen Sie **Einfügen -> Subnetz -> Industrial Ethernet**.

**Ergebnis:** Im Projekt wird ein Objekt vom Typ Netz angelegt. Damit können alle im Projekt angelegten SIMATIC-Stationen an dieses Subnetz angeschlossen werden.

2



### 2.4.3 Ethernet-CP in die Hardwarekonfiguration eintragen

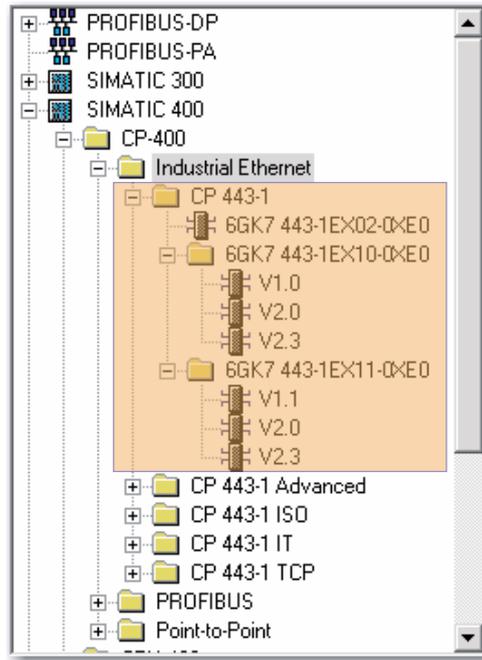
#### Vorgehensweise

Indem Sie den Ethernet-CP in das Rack einer SIMATIC-400er Station einbauen und zuordnen, stellen Sie den logischen Anschluss zwischen Ethernet-CP und Subnetz her.

1. Wählen Sie in Ihrem Projekt die Station aus, die Sie über den Ethernet-CP an Industrial Ethernet anschliessen möchten.
2. Plazieren Sie den CP443-1 in der Konfigurationstabelle wie jede andere Baugruppe, indem Sie ihn aus dem Hardware Katalog auswählen und den Steckplatz im Rack wählen.



Als Ethernet CP muss **zwingend** eine CP443-1 projektiert werden. Der Ausgabe-stand spielt keine Rolle. In nachfolgender Abbildung sind die unterstützten Ethernet CPs orange markiert.

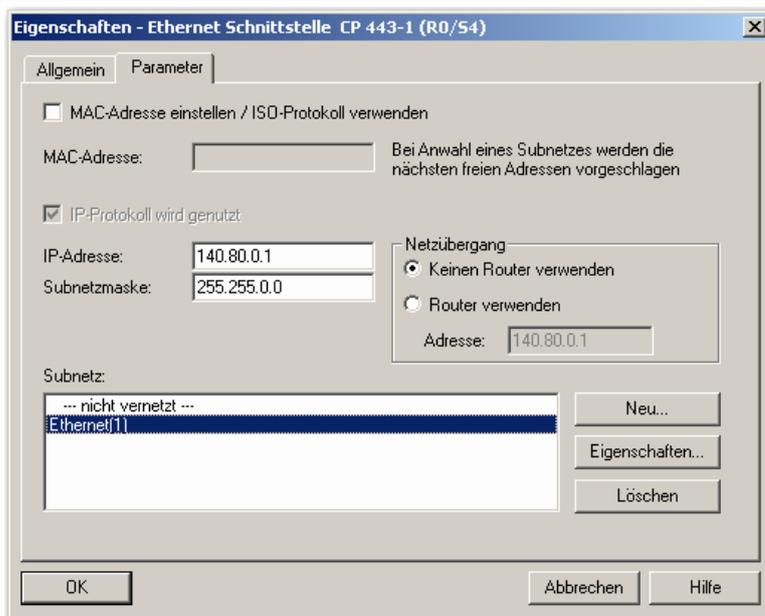


CPs werden im Hardware Katalog anhand eines kurzen Beschreibungstextes, ergänzt durch die Bestellnummer, ausgewählt. Die CPs sind aufgrund der Installation von NCM S7 im Katalog eingetragen und verfügbar.

**Ergebnis:** Der CP ist der SIMATIC-Station zugeordnet.

**Subnetz anschliessen**

Damit Sie den Netzanschluss des Ethernet CP aktivieren können, blendet der SIMATIC Manager den folgenden Dialog ein:





Sie können den Dialog für die Einstellung der Schnittstelle jederzeit über den Eigenschaftendialog des CP - dort im Register "Allgemein" - aufrufen.

1 Überprüfen Sie die Adresse und ändern Sie diese ggf. ab. Die Adresse wird automatisch eingetragen, indem die nächste freie Adresse ermittelt wird.

Detailinformationen zu den Adressbereichen gibt Ihnen die Online- Hilfe. Beachten Sie aber bitte folgende Zusatzinformationen:

### **MAC-Adresse**

Die aktuellen PCD7.F655 sowie die PCD Systeme mit integriertem Ethernet Stack werden mit einer voreingestellten MAC-Adresse ausgeliefert (siehe Adressaufdruck auf der Baugruppe). Um eine eindeutige Adressvergabe sicherzustellen, tragen Sie keine MAC-Adresse in der Projektierung ein (die Option ist deaktiviert). Dadurch verwendet die Baugruppe automatisch die eingetragene Werksadresse. Die in diesem Feld eingetragene MAC-Adresse wird von der PCD nicht ausgewertet, bzw. hat keinen Einfluss auf die Ethernetdienste.



Die ISO-Dienste werden von den Saia PCDs nicht unterstützt. Es ist somit nicht möglich, den CP über die Voreingestellte MAC-Adresse anzusprechen und ihn mit weiteren Adressinformationen zu versorgen.

2 Wählen Sie den gewünschten Subnetztyp im Listenfeld "Subnetz" aus.

3 Für das ausgewählte Subnetz können Sie den Eigenschaftendialog anzeigen lassen. Wählen Sie hierzu die entsprechende Schaltfläche.

4 Tragen Sie im Register "Allgemein" spezifische Informationen ein, die den Subnetzknoten charakterisieren.

5 Bestätigen Sie unbedingt Ihre Eingabe mit OK, da nur dann die Vernetzung übernommen wird (siehe Pkt. 2.).



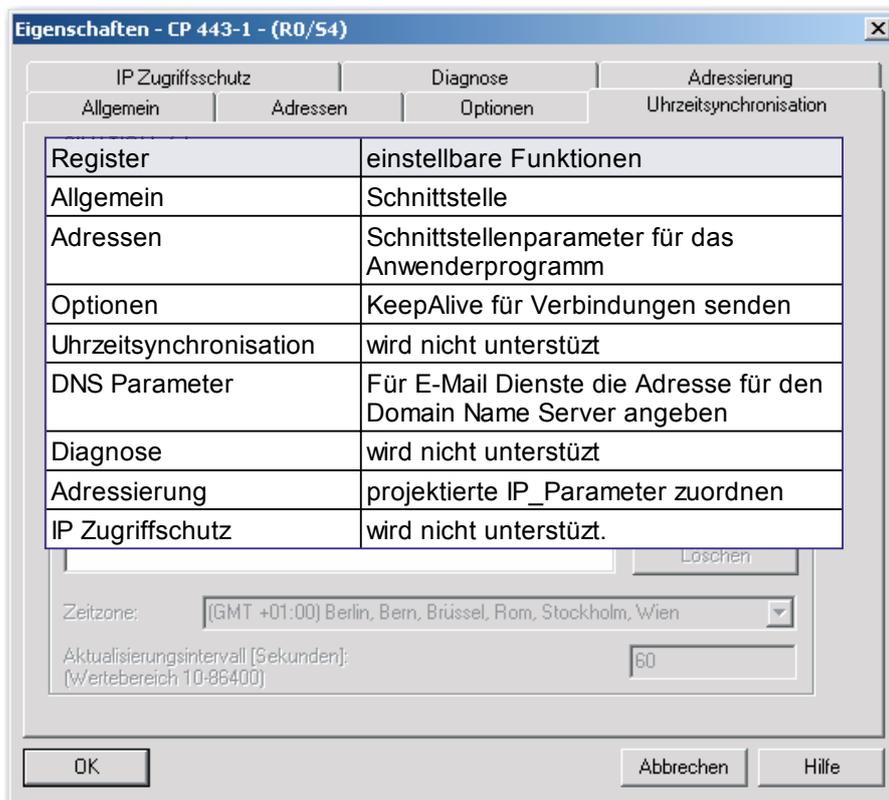
Die hier beschriebenen Adresseinstellungen gelangen erst mit dem Laden der Projektierdaten in die Saia PCD.

## 2.4.4 Weitere CP- Eigenschaften einstellen

Sie können zusätzlich zum Netzanschluss weitere baugruppenspezifische Einstellungen vornehmen.

1. Markieren Sie den Ethernet-CP in der Hardwarekonfiguration.
2. Wählen Sie **Bearbeiten -> Objekteigenschaften**. In der vorgelegten Maske finden Sie abhängig vom Typ des CP neben Register "Allgemein" weitere Register:

2



### Register Adressen

Im Register Adressen wird ausgegeben, unter welcher Adresse die Baugruppe vom Anwenderprogramm angesprochen werden kann. Sie benötigen diese Adresse beim Aufruf der FC-Bausteine für Transportverbindungen.

### Register Optionen

Abhängig vom CP-Typ werden diverse Einstellmöglichkeiten angeboten. Für die Saia PCDs ist nur die Option "Keep Alive für Verbindungen senden" relevant. Alle andern Optionen werden nicht unterstützt, bzw. haben keine Auswirkungen auf die Saia PCD.

Sie können hier die Intervallzeit einstellen, mit der Lebenszeichentelegramme (Keep Alive) an den Partner einer Kommunikationsverbindung gesendet werden. Mit dieser Intervallzeit legen Sie die Zeitspanne fest, nach der der Ausfall eines Kommunikationspartners spätestens bemerkt wird. Der Ethernet-CP ist für alle verbindungsorientierten Dienste so konfiguriert, dass Lebenszeichentelegramme (Keep Alive) gesendet werden. Dadurch ist gewährleistet, dass Verbindungen nach dem Ausfall eines Kommunikationspartners beendet, und die Verbindungsressourcen freigegeben werden. Die hier vorgenommene Einstellung gilt für alle über den CP betriebenen TCP- und ISO-on-TCP-Verbindungen; eine verbindungsorientierte Einstellung ist

nicht möglich.

**Wertebereich:**

Default-Einstellung:	30 Sekunden
Keep Alive ausschalten:	0 Sekunden
Maximalwert:	65535 Sekunden

2



Beachten Sie, dass der Keep Alive Mechanismus dazu führen kann, dass unterlagerte Verbindungen (z.B. eine ISDN-Telefonverbindung) aufrecht erhalten werden, obwohl keine eigentlichen Nutzdaten übertragen werden. Wenn dies unerwünscht ist, müssen Sie die Intervallzeit so hoch einstellen, dass die unterlagerte Verbindung nach dem Ausbleiben von Nutzdaten beendet wird, bevor ein Keep Alive-Telegramm gesendet wird.



Der Keep-Alive Parameter hat auf dem F650-Modul eine andere Bedeutung als auf der PCD3: Dieser Parameter entspricht dem Connection-Timout des F650 Moduls. Werden solange keine Parameter über ein Port empfangen, bis die Connection-Timout Zeit abgelaufen ist, so geht das F650-Modul davon aus, dass die Gegenstation nicht mehr am Netz ist, und schliesst die Verbindung. Anschliessend Versucht das Modul eine neue Verbindung aufzubauen.

**Register Adressierung**

Im Register "Adressierung" können Sie dem CP die zuvor projektierte IP-Adresse und IP-Parameter zuweisen. Erst danach können die Projektierdaten mittels PG/PC über Ethernet in den CPU geladen werden.

Die Einstellungen in den übrigen Registern werden von den Saia PCDs nicht ausgewertet haben keinen Einfluss auf die Ethernetfunktionalitäten.



Weitere Möglichkeiten zur Konfiguration der Ethernet-Parameter sind im Kapitel 5 zu finden,

## 3 Kommunikationsverbindungen projektieren/programmieren

### 3.1 Allgemeines

Eine Kommunikationsverbindung ermöglicht die programmgesteuerte Kommunikation zwischen zwei Teilnehmern am Ethernet mit folgenden Eigenschaften:

- Der Datentransfer ist bidirektional, d.h. auf der Verbindung kann gleichzeitig gesendet und empfangen werden.
- Beide Teilnehmer sind gleichberechtigt, d.h. jeder Teilnehmer kann ereignisabhängig den Sende- und Empfangsvorgang anstoßen.
- Die Adresse des Kommunikationspartners wird per Projektierung/Programmierung festgelegt.

Eine Ausnahme hiervon bilden die Freie UDP-Verbindung. Dort wird die Adresse an der FC-Schnittstelle im Anwenderprogramm angegeben.

Der Verbindungsaufbau erfolgt automatisch, sobald der Partner erreichbar ist.

Es gibt Projekte in welchen zum Zeitpunkt der Projektierung die Partner (- Adressen) noch nicht bekannt sind. z.B die IP- Adressen werden durch einen DHCP - Server vergeben, oder die Anlage wird bei der Inbetriebnahme durch eine Web- Visualisierung fertig konfiguriert (ohne SIMATIC Tool).

Für diese Fälle existieren Systemfunktionen mit welchen während der Programmlaufzeit Verbindungen erzeugt werden können. Diese Funktionen werden in Kapitel 3.6 u.f. beschrieben.

#### 3.1.1 Datenvolumen und Mengengerüst

Wieviele Kommunikationsverbindungen die jeweiligen PCDs unterstützen entnehmen sie Bitte der nachfolgenden Tabelle.

PCD-Typ	max. Anzahl Verbindungen <sup>1)</sup>
PCD2.M157	8
PCD2.M177	64
PCD2.M487	64
PCD2.Mxxx7	64
PCD3.Mxxx7	64

<sup>1)</sup> Die maximale Anzahl projektierbarer Verbindungen verringert sich, falls noch unprojektierte S7 (Server) / PG/OP- Dienste über Ethernet benützt werden.

Eine PCD mit Ethernet kann, über eine Verbindung, pro Auftrag folgende Datenmenge übertragen (Senden / Empfangen):

PCD-Typ	ISO-on-TCP Send/Receive	ISO-on-TCP Fetch/Write passiv (Server)	TCP Send/Receive	TCP Fetch/Write passiv (Server)	UDP	S7 (Server) PG/OP
PCD2.M157	8192	-	8192	-	744	460
PCD2.M177	8192	-	8192	-	744	460
PCD2.M487	8192	460	8192	720	744	460
PCD2.Mxxx7	8192	460	8192	720	744	460
PCD3.Mxxx7	8192	460	8192	720	744	460

### 3.1.2 Neue Verbindung erzeugen

Wenn Sie neue Verbindungen anlegen, gehen Sie von eingetragenen und vernetzten Stationen aus. Eine Verbindung wird dann projiziert, indem ausgehend von einer Station bzw. CPU im aktuellen STEP7-Projekt eine Zielstation selektiert wird.

Aufgrund der Vernetzung sind die Knotenadressen (IP-Adressen) der beiden Stationen bereits festgelegt. Für die lokalen und fernen TSAPs (Transport Service Access Point) bzw. Ports werden auf beiden Verbindungsendpunkten automatisch Defaultwerte vergeben.

Handelt es sich bei der Partnerstation um eine Station ausserhalb des aktuellen Projektes oder um eine Nicht-S7-Station, müssen die fernen TSAPs (Transport Service Access Point) bzw. Ports im Eigenschaftendialog der Verbindung spezifiziert werden.

Für jede CPU wird eine eigene Verbindungstabelle erstellt, in der die Verbindungspartner und die Typen der Verbindungen angezeigt werden.

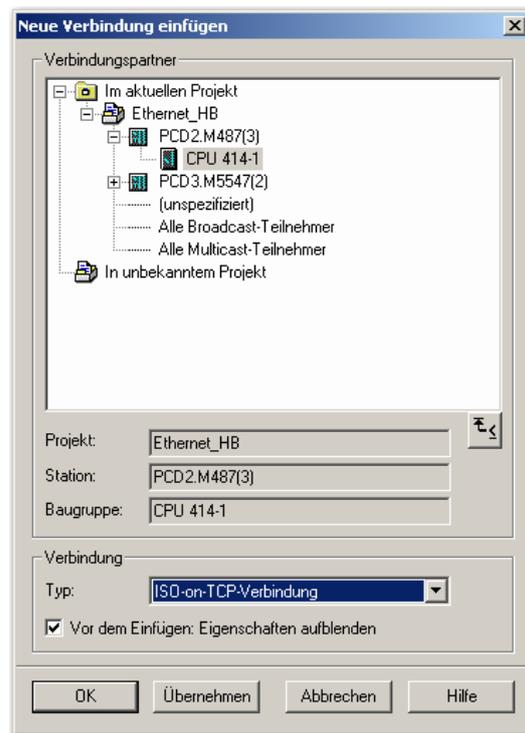
#### Neue Verbindung erzeugen

Voraussetzung für die Projektierung einer neuen Verbindung ist, dass die Stationen mit Ihren CPs konfiguriert und im S7-Projekt vernetzt sind. Um eine neue Verbindung zu erstellen, gehen Sie folgendermassen vor:

Gehen Sie von NetPro aus folgendermassen vor:

1. Selektieren Sie in NetPro die Station oder die CPU in der Station, von der aus Sie die Verbindung aufbauen wollen
2. Wählen Sie den Menübefehl Einfügen Neue Verbindungen (auch über die rechte Maustaste zu erreichen!)

**Ergebnis:** Auf dem Bildschirm erscheint der folgende Dialog:



3

3. Markieren Sie die Partnerstation, zu der Sie eine Verbindung aufbauen möchten (falls mehrere CPUs vorhanden sind, markieren Sie bitte die gewünschte CPU)
4. Wählen Sie im Eingabefeld "Typ" den Verbindungstyp aus, den Sie verwenden wollen, z.B. ISO-on-TCP Verbindung

Wenn Sie Ihre Eingabe mit Übernehmen bestätigen, dann wird die neue Verbindung angelegt und das Dialogfeld "Neue Verbindung" bleibt geöffnet. So haben Sie die Möglichkeit, weitere Verbindungen anzulegen, ohne das Dialogfeld erneut zu öffnen. Gleichzeitig wird die Verbindungstabelle aktualisiert.

### **Verbindungen zu Nicht-S7-Stationen über Stellvertreterobjekte**

Wenn Sie Verbindungen zu Geräten oder Stationen projektieren möchten, die keine S7-Stationen sind, wählen Sie als Zielstation eine Station vom Typ "SIMATIC S5", "PC/PG", "SIMATIC PC-Station" oder "Andere Station" aus.

Aufgrund der Vernetzung, die Sie beim Anlegen dieser Stationen in NetPro vornehmen, sind die Knotenadressen (IP-Adressen) der beiden Stationen bereits festgelegt. Den TSAP (Transport Service Access Point) bzw. Port für die Zielstation müssen Sie angeben.



Verbindungen zu Nicht-S7-Stationen (z.B. SIMATIC S5) werden als unvollständig spezifizierte Verbindungen generiert, d.h. der ferne TSAP bzw. der Port sind leer. Diese Verbindungen müssen im Eigenschaftendialog spezifiziert werden.

## Unspezifizierte Verbindungen

Verbindungen zu einem noch nicht bekannten Gerät (z.B. Diagnosegerät) werden als "unspezifizierte" Verbindungen projiziert. Sie können später im Eigenschaftendialog noch spezifiziert werden.

Eine unspezifizierte Verbindung können Sie anlegen, indem Sie beim Neuanlegen der Verbindung unter Verbindungspartner - Station "unspezifiziert" angeben. Die unspezifizierte Verbindung kann auf 3 Arten genutzt werden (nachfolgend erläutert am Beispiel ISO-on-TCP-Verbindung; auf TCP-Verbindung sinngemäss anzuwenden):

- Kommunikationsbereitschaft erklären - passiver Verbindungsaufbau

Der Verbindungsaufbau ist dann passiv einzustellen (siehe Einstellungen im Register Allgemein). Für die Adresseinstellung bei ISO-on-TCP-Verbindung gilt dann: die ferne IP-Adresse und der ferne TSAP sind leer, d.h. sie sind für den CP nicht relevant. Beim Verbindungsaufbau wird dann jeder Partner akzeptiert (Partner = Verbindungsname, der den CP mit korrekter IP-Adresse und TSAP-adressiert).

Es besteht auch die Möglichkeit der Teilspezifizierung, d.h. die Kommunikation wird mit einem beliebigen Partner zugelassen, der in dem spezifizierten TSAP übereinstimmt.

- Verbindung zu einer bestimmten Station in einem beliebigen Projekt

Für die Adresseinstellung bei ISO-on-TCP-Verbindung gilt dann: Sie können die ferne IP-Adresse und den Port für eine beliebige Zielstation angeben. Die Zielstation kann innerhalb oder ausserhalb des aktuellen STEP®7-Projekts liegen.

Verwenden Sie diese Variante, wenn Sie für die Partnerstation kein Stellvertreterobjekt, z.B. SIMATIC S5, im aktuellen Projekt angelegt haben.

Die folgende Tabelle fasst die Möglichkeiten zusammen.

Bedeutung für Verbindungsaufbau	IP-Adresse (fern)	TSAP / Port (fern)	möglicher Verbindungsaufbau
von beliebigem Partner	leer	leer	passiv
von beliebigem Partner über bestimmten TSAP	leer	spezifiziert	passiv
zu oder von einem bestimmten Partner	spezifiziert	spezifiziert	aktiv / passiv

Eine weitere Variante stellt die Freie UDP-Verbindung dar. Bei dieser Verbindungsart bleibt die Adresse des Verbindungspartners bei der Projektierung offen. Die Kommunikationsteilnehmer sind durch Adressangaben im Kommunikationsauftrag des Anwenderprogrammes bestimmt.

Weitere Informationen hierzu finden Sie nachfolgend in den verbindungspezifischen Einzelkapiteln.

## 3.2 ISO-on-TCP Verbindungseigenschaften projektieren

Beim Anlegen einer Verbindung im Neu-Dialog werden der Verbindungstyp und soweit möglich der Verbindungspartner festgelegt.

Weitere Verbindungsparameter, die beim Neuanlegen zunächst standardmässig vorbesetzt werden, können Sie bei Bedarf wie nachfolgend beschrieben individuell anpassen.

### Dialog aufrufen

Um den Dialog für die speziellen Verbindungseigenschaften aufzurufen, gehen Sie folgendermassen vor:

1. Selektieren Sie in der Verbindungstabelle die gewünschte Verbindung.
2. Wählen Sie im Menü Bearbeiten -> Objekteigenschaften

Ergebnis: Es erscheint der Dialog "Eigenschaften für ISO-on-TCP-Verbindungen".

### Register

Für ISO-on-TCP-Verbindungen stehen folgende Register zur Verfügung:

- **Allgemein**

Anzeige von Parametern, welche die Verbindung identifizieren.

- **Adressen**

Anzeige der lokalen und fernen Adressinformationen.

- **Optionen**

Hier können Sie auswählen, ob Sie die Verbindung für eine der Zugriffsarten SEND/RECV, FETCH passiv oder WRITE passiv nutzen möchten. FETCH passiv oder WRITE passiv müssen auf PCD Systemen nicht zwingend projiziert werden.

- **Übersicht**

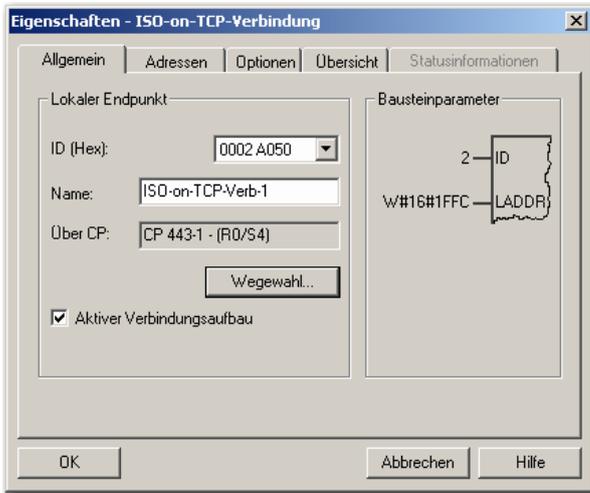
Übersicht aller projizierten ISO-on-TCP-Verbindungen der selektierten Station mit den entsprechenden Parametern und dem Status der Verbindung.

- **Statusinformationen**

Diese Informationen entsprechen der Anzeige von NCM Diagnose. Die NCM Diagnose wird von den Saia PCDs nicht unterstützt.

### 3.2.1 Den lokalen Verbindungsendpunkt festlegen

Im Register Allgemein des Eigenschaftsdialogs werden allgemeine Verbindungsparameter angezeigt, die den lokalen Verbindungsendpunkt identifizieren.



3

Parameter	Beschreibung	Zugriff
<b>Lokaler Endpunkt</b>		
ID	Der Eintrag ist identisch mit dem Parameter Lokale ID der Verbindungstabelle. <sup>1)</sup>	wählbar
Name	Beim Anlegen der Verbindung wird hier ein Vorschlagsname angezeigt, der als Suffix eine Verbindungsnummer enthält. Nutzen Sie dieses Feld bei unspezifizierten Verbindungen zur Kennzeichnung des Partners.	änderbar
Über CP	Da die Saia PCDs nur 1 Ethernet-Modul pro PCD unterstützten, macht es keinen Sinn, mehrere CPs pro CPU zu Projektieren.	nur lesbar

Parameter	Beschreibung	Zugriff
Aktiver Verbindungsaufbau	<p>Legen Sie mit dieser Option fest, ob der Verbindungsaufbau von dieser S7-Station aus erfolgen soll. Die Option ist standardmässig angewählt, wenn die Adresse des Partners spezifiziert ist.</p> <p>EIN: Verbindung wird aktiv aufgebaut</p> <p>AUS: Verbindung wird vom Partner aufgebaut</p> <p>Wenn Sie beim Anlegen der Verbindung als Verbindungspartner "unspezifiziert" gewählt haben, ist die Option standardmässig abgewählt. Wenn Sie die Option aktivieren, müssen Sie die Adresse des Partners im Register "Adressen" spezifizieren.</p> <p><b>Hinweis:</b></p> <p>Beachten Sie bitte die Auswirkung auf die Betriebsart. Wenn die Betriebsart FETCH oder WRITE (siehe Register "Optionen") genutzt werden soll, muss bei der S7-Station die Betriebsart auf "passiv" gestellt werden.</p>	änderbar
<b>Bausteinparameter</b>		
ID	Dieser Bausteinanrufparameter ist beim Aufruf des FC im Anwenderprogramm zur Identifikation der Verbindung zwingend einzutragen (die ID ändert sich bei Änderung der lokalen ID).	nur lesbar
LADDR	Dieser Bausteinanrufparameter ist beim Aufruf des FC im Anwenderprogramm zur Identifikation der Verbindung zwingend einzutragen.	nur lesbar



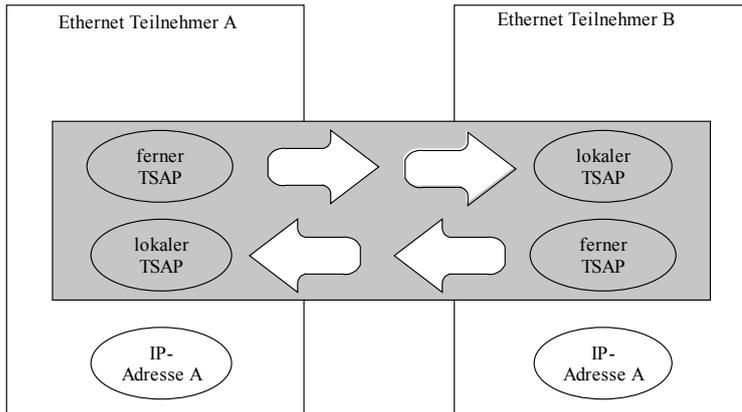
<sup>1)</sup> Die PCD2.M157 unterstützt maximal 8 Verbindungen. Auf der PCD2.M157 sind als ID nur Werte zwischen 1 und 8 erlaubt. Wird ein Wert grösser 8 eingetragen, so wird diese Verbindung auf der PCD ignoriert.

### 3.2.2 ISO-on-TCP Adressen festlegen

Eine Verbindung wird durch den lokalen und fernen Verbindungsendpunkt spezifiziert.

- Lokale Adressen:  
Lokale IP-Adresse und lokaler TSAP (Transport Service Access Point)
- Ferne Adressen:  
Ferne IP-Adresse und ferner TSAP

3



Die Adressparameter werden mit NCM S7 projiziert und in der CP-Datenbasis abgelegt. Änderungen und Erweiterungen der Ethernet-Netztopologie sind dadurch rückwirkungsfrei auf das CPU-Anwenderprogramm.

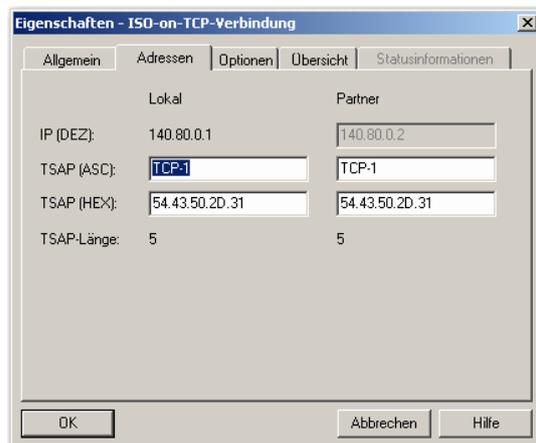


Bei der Projektierung des Ethernet-CP und in der Ethernet-Zielstation müssen die TSAPs einer ISO-on-TCP-Verbindung kreuzweise übereinstimmen:

Ferner TSAP (im Ethernet-CP) = lokaler TSAP (in Ziel-Station);

Lokaler TSAP (im Ethernet-CP) = ferner TSAP (in Ziel-Station);

Im Register Adressen werden die relevanten lokalen und fernen Adressinformationen als Vorschlagswerte angezeigt. Sie haben die Möglichkeit die TSAPs individuell einzustellen.



ISO-on-TCP-Verbindungen haben eine TSAP-Länge von 1 - 16 Byte. Bei der Eingabe wird die aktuelle Länge automatisch angezeigt (sichtbare Anzeige: 16 ASCII-Zeichen). Lokale und ferne TSAPs können als Hexadezimal-Wert oder als ASCII-String eingegeben werden. Bei ASCII-Eingabe werden die eingegebenen Zeichen auch hexadezimal angezeigt. Bei Hexadezimal-Eingabe werden druckbare Zeichen als ASCII-Wert (sichtbar sind 8 Hexadezimalzeichen) dargestellt.

Werden nicht druckbare Zeichen eingegeben, so wird die ASCII Anzeige graut (keine ASCII-Eingabe mehr möglich) und die nicht druckbaren Zeichen werden als Punkt dargestellt.

3



Ferne und lokale TSAPs können identisch sein, da die Verbindung durch die unterschiedlichen IP-Adressen eindeutig ist. Wenn zwischen zwei Stationen mehr als eine Verbindung eingerichtet werden soll, müssen sich auch die TSAPs unterscheiden.

### Unspezifizierte ISO-on-TCP-Verbindung

Wenn Sie den Partnertyp "unspezifiziert" gewählt haben, können Sie je nach Zielsetzung hier Adressangaben zum Kommunikationspartner vornehmen.

Das dargestellte Register Adressen zeigt den Fall, bei dem die ferne IP-Adresse und der ferne TSAP nicht spezifiziert sind.

	Lokal	Partner
IP (DEZ):	140.80.0.1	
TSAP (ASC):	TCP-1	
TSAP (HEX):	54.43.50.2D.31	
TSAP-Länge:	5	0



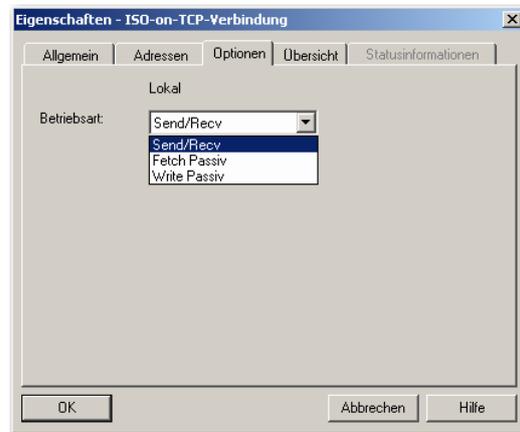
Wenn Sie keine weiteren Angaben machen, erklären Sie damit die Bereitschaft, mit einem beliebigen Kommunikationspartner eine Verbindung aufzubauen.



Diese Möglichkeit wird nur auf der PCD2.M487 und auf den PCD Systemen mit integriertem Ethernet Stack unterstützt. Werden auf der PCD2.M157/M177 keine weiteren Angaben gemacht, so wird diese Verbindung in der PCD nicht erkannt. Auf der PCD2.M157/M177 müssen die ferne IP-Adresse und der ferne TSAP immer spezifiziert werden!

### 3.2.3 ISO-on-TCP Optionen

Im Register Optionen stellen Sie zusätzliche Parameter für die ISO-on-TCP Verbindung ein.



3

#### Betriebsart

##### SEND/RECV:

Dies ist der Standardbetrieb mit den FCs AG\_SEND/AG\_LSEND und AG\_RECV/AG\_LRECV.

##### FETCH passiv

Indem Sie die Betriebsart Fetch wählen, ermöglichen Sie den direkten, lesenden Zugriff auf Datenbereiche in der Saia PCD. Dieser Zugriff kann beispielsweise von einer SIMATIC S5-Station oder einer Fremdstation aus erfolgen. Diese Betriebsart muss nicht zwingend projiziert werden. Von der Saia PCD können immer Daten über Fetch passiv gelesen werden.

##### WRITE passiv

Indem Sie die Betriebsart Write wählen, ermöglichen Sie den direkten, schreibenden Zugriff auf Datenbereiche in der Saia PCD. Dieser Zugriff kann beispielsweise von einer SIMATIC S5-Station oder einer Fremdstation aus erfolgen. Diese Betriebsart muss nicht zwingend projiziert werden. Auf die Saia PCD können immer Daten über Write passiv geschrieben werden.

Diese Verbindung kann dann ausschliesslich für diese Betriebsart genutzt werden. Das Senden oder Empfangen über die FCs AG\_SEND/AG\_LSEND bzw. AG\_RECV/AG\_LRECV ist dann nicht möglich.

Voraussetzung für FETCH/WRITE: Art des Verbindungsaufbaus

Wählen Sie bitte abhängig vom Stationstyp für den Verbindungsaufbau folgende Option im Register "Allgemein":

- Saia PCD-Station: PASSIVER Verbindungsaufbau
- PC-Station: Aktiver Verbindungsaufbau

Option "S7-Adressierungsmodus" (nur bei FETCH/WRITE)

Bei der Projektierung einer ISO-on-TCP Verbindung für die Betriebsart FETCH / WRITE können Sie hier den Adressierungsmodus wählen.

- S7-Adressierungsmodus: Byte-Adresse
- (S5-Adressierungsmodus: Wort-Adresse)

Standardmässig eingestellt ist der Adressierungsmodus für den Zugriff auf SIMATIC S7 (Option angewählt).



Die Saia PCDs unterstützen nur den S7-Adressierungsmodus. Wird die Option abgewählt, so hat das keinen Einfluss auf die PCD Steuerung. Der Adressierungsmodus wird trotzdem als Byte-Adresse interpretiert.



Die Verbindungsarten FETCH/WRITE werden nur von Steuerungen PCD2.M487 und den PCD-Systemen mit integriertem Ethernet Stack unterstützt.

3

### 3.3 TCP Verbindungseigenschaften projektieren

Beim Anlegen einer Verbindung im Neu-Dialog werden der Verbindungstyp und soweit möglich der Verbindungspartner festgelegt.

Weitere Verbindungsparameter, die beim Neuanlegen zunächst standardmässig vorbesetzt werden, können Sie bei Bedarf wie nachfolgend beschrieben individuell anpassen.

#### Dialog aufrufen

Um den Dialog für die speziellen Verbindungseigenschaften aufzurufen, gehen Sie folgendermassen vor:

1. Selektieren Sie in der Verbindungstabelle die gewünschte Verbindung.
2. Wählen Sie im Menü Bearbeiten Objekteigenschaften

Ergebnis: Es erscheint der Dialog "Eigenschaften für TCP-Verbindungen".

#### Register

Für TCP-Verbindungen stehen folgende Register zur Verfügung:

- **Allgemein**

Anzeige von Parametern, welche die Verbindung identifizieren.

- **Adressen**

Anzeige der lokalen und fernen Adressinformationen.

- **Optionen**

Hier können Sie auswählen, ob Sie die Verbindung für eine der Zugriffsarten FETCH oder WRITE nutzen möchten.

- **Übersicht**

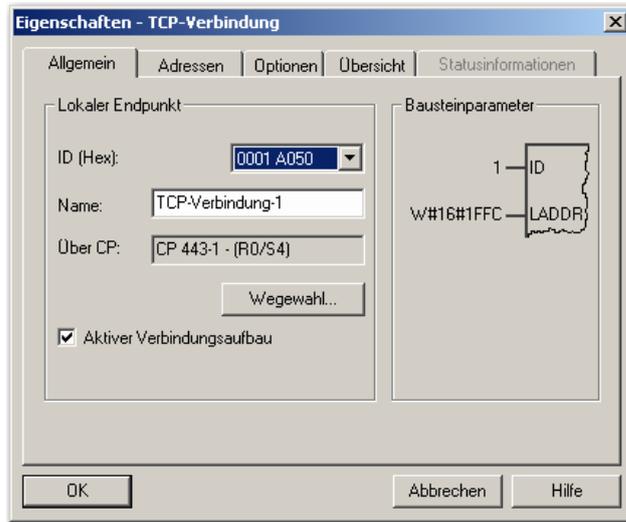
Übersicht aller projektierten TCP-Verbindungen der selektierten Station mit den entsprechenden Parametern und dem Status der Verbindung.

- **Statusinformationen**

Diese Informationen entsprechen der Anzeige von NCM Diagnose. Die NCM Diagnose wird von den Saia PCDs nicht unterstützt.

### 3.3.1 Den lokalen Verbindungsendpunkt festlegen

Im Register Allgemein des Eigenschaftsdialogs werden allgemeine Verbindungsparameter angezeigt, die den lokalen Verbindungsendpunkt identifizieren.



3

Parameter	Beschreibung	Zugriff
<b>Lokaler Endpunkt</b>		
ID	Der Eintrag ist identisch mit dem Parameter Lokale ID der Verbindungstabelle. <sup>1)</sup>	wählbar
Name	Beim Anlegen der Verbindung wird hier ein Vorschlagsname angezeigt, der als Suffix eine Verbindungsnummer enthält. Nutzen Sie dieses Feld bei unspezifizierten Verbindungen zur Kennzeichnung des Partners.	änderbar
Über CP	Da die Saia PCDs nur 1 Ethernet-Modul pro PCD unterstützen, macht es keinen Sinn, mehrere CPs pro CPU zu Projektieren.	nur lesbar
Aktiver Verbindungsaufbau	Legen Sie mit dieser Option fest, ob der Verbindungsaufbau von dieser PCD-Station aus erfolgen soll. Die Option ist standardmässig angewählt, wenn die Adresse des Partners spezifiziert ist.  Wenn Sie beim Anlegen der Verbindung als Verbindungspartner "unspezifiziert" gewählt haben, ist die Option standardmässig abgewählt. Wenn Sie die Option aktivieren, müssen Sie die Adresse des Partners im Register "Adressen" spezifizieren.  EIN: Verbindung wird aktiv aufgebaut  AUS: Verbindung wird vom Partner aufgebaut	änderbar
<b>Bausteinparameter</b>		
ID	Dieser Bausteinaufrufparameter ist beim Aufruf des FC im Anwenderprogramm zur Identifikation der Verbindung zwingend einzutragen (die ID ändert sich bei Änderung der lokalen ID).	nur lesbar

Parameter	Beschreibung	Zugriff
LADDR	Dieser Bausteinaufrufparameter ist beim Aufruf des FC im Anwenderprogramm zur Identifikation der Verbindung zwingend einzutragen.	nur lesbar

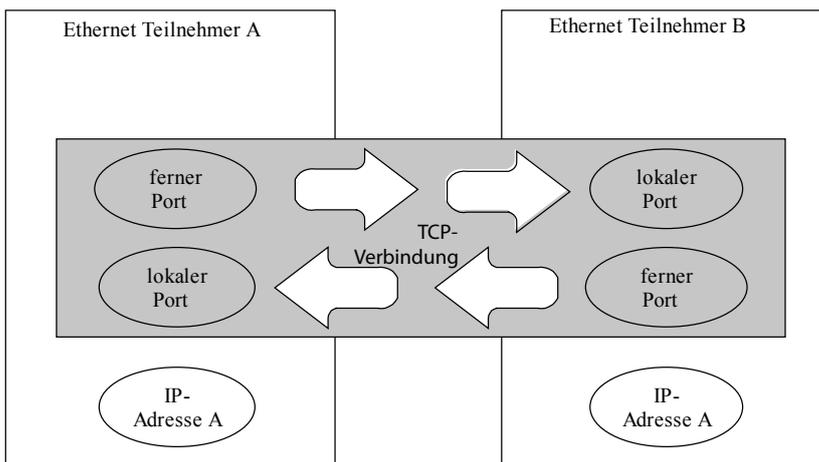


1) Die PCD2.M157 unterstützt maximal 8 Verbindungen. Auf der PCD2.M157 sind als ID nur Werte zwischen 1 und 8 erlaubt. Wird ein Wert grösser 8 eingetragen, so wird diese Verbindung auf der PCD ignoriert.

### 3.3.2 TCP Adressen festlegen

Bei TCP werden die Kommunikationspartner durch den lokalen und fernen Endpunkt wie folgt adressiert.

- Lokale Adressen:  
Lokale IP-Adresse und lokaler Port
- Ferne Adressen:  
Ferne IP-Adresse und ferner Port

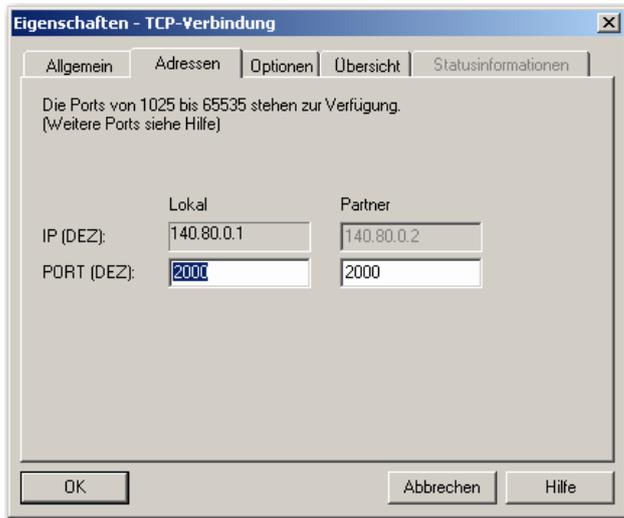


Abhängig vom gewünschten Verbindungstyp werden die fernen Adressparameter bei der Projektierung spezifiziert oder offen gelassen.

- **spezifizierte TCP-Verbindung**  
Sie haben beim Neuanlegen der Verbindung eine Zielstation angegeben.
- **unspezifizierte TCP-Verbindung**  
Sie haben beim Neuanlegen der Verbindung unter Verbindungspartner - Station "unspezifiziert" angegeben.

## Spezifizierte TCP-Verbindung

Im Register Adressen werden die relevanten lokalen und fernen Adressinformationen als Vorschlagswerte angezeigt. Sie haben die Möglichkeit, die Ports individuell einzustellen.



3

## Ports

Die Ports bzw. die Port-Adressen definieren den Zugangspunkt zum Anwenderprogramm innerhalb der Station / CPU. Sie müssen innerhalb der Station / CPU eindeutig sein!



Port Adressen zwischen 0...1024 sollten nicht verwendet werden. Diese sind standardmässig vergeben (well known ports).

## Unspezifizierte TCP-Verbindung

Wenn Sie den Partnertyp “unspezifiziert” gewählt haben, können Sie je nach Zielsetzung hier Adressangaben zum Kommunikationspartner vornehmen.

Das dargestellte Register Adressen zeigt den Fall, bei dem der ferne Port, nicht jedoch die IP-Adresse spezifiziert sind.

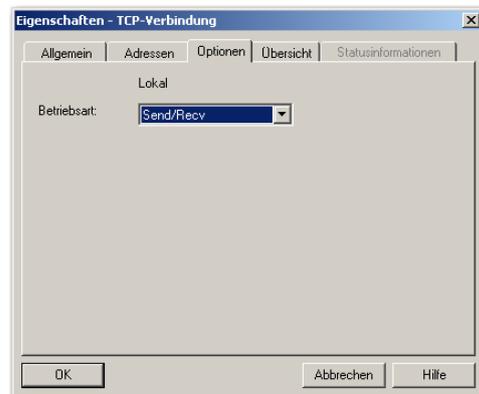


3

Wenn Sie keine weiteren Angaben machen, erklären Sie damit die Bereitschaft, mit jedem Kommunikationspartner eine Verbindung aufzubauen, der den angegebenen Port verwendet.

### 3.3.3 TCP- Optionen

Im Register Optionen stellen Sie zusätzliche Parameter für die TCP Verbindung ein.



3

#### Betriebsart

##### **SEND/RECV:**

Dies ist der Standardbetrieb mit den FCs AG\_SEND/AG\_LSEND und AG\_RECV/AG\_LRECV.

##### **FETCH passiv**

Indem Sie die Betriebsart Fetch wählen, ermöglichen Sie den direkten, lesenden Zugriff auf Datenbereiche in der Saia PCD. Dieser Zugriff kann beispielsweise von einer SIMATIC S5-Station oder einer Fremdstation aus erfolgen.

##### **WRITE passiv**

Indem Sie die Betriebsart Write wählen, ermöglichen Sie den direkten, schreibenden Zugriff auf Datenbereiche in der Saia PCD. Dieser Zugriff kann beispielsweise von einer SIMATIC S5-Station oder einer Fremdstation aus erfolgen.

Diese Verbindung können dann ausschliesslich für diese Betriebsart genutzt werden. Das Senden oder Empfangen über die FCs AG\_SEND/AG\_LSEND bzw. AG\_RECV/AG\_LRECV ist dann nicht möglich.

Voraussetzung für FETCH/WRITE: Art des Verbindungsaufbaus

Wählen Sie bitte abhängig vom Stationstyp für den Verbindungsaufbau folgende Option im Register "Allgemein":

- Saia PCD-Station: PASSIVER Verbindungsaufbau
- PC-Station: Aktiver Verbindungsaufbau

Option "S7-Adressierungsmodus" (nur bei FETCH/WRITE)

Bei der Projektierung einer ITCP Verbindung für die Betriebsart FETCH / WRITE können Sie hier den Adressierungsmodus wählen.

- S7-Adressierungsmodus: Byte-Adresse
- (S5-Adressierungsmodus: Wort-Adresse)

Standardmässig eingestellt ist der Adressierungsmodus für den Zugriff auf SIMATIC S7 (Option angewählt).



Die Saia PCDs unterstützen nur den S7-Adressierungsmodus. Wird die Option abgewählt, so hat das keinen Einfluss auf die PCD Steuerung. Der Adressierungsmodus wird trotzdem als Byte-Adresse interpretiert.

### 3.4 UDP Verbindungseigenschaften projektieren

Beim Anlegen einer Verbindung im Neu-Dialog werden der Verbindungstyp und soweit möglich der Verbindungspartner festgelegt.

3

Weitere Verbindungsparameter, die beim Neuanlegen zunächst standardmässig vorbesetzt werden, können Sie bei Bedarf wie nachfolgend beschrieben individuell anpassen.

Um den Dialog für die speziellen Verbindungseigenschaften aufzurufen, gehen Sie folgendermassen vor:

1. Selektieren Sie in der Verbindungstabelle die gewünschte Verbindung.
2. Wählen Sie im Menü Bearbeiten Objekteigenschaften

Ergebnis: Es erscheint der Dialog "Eigenschaften für UDP-Verbindungen".

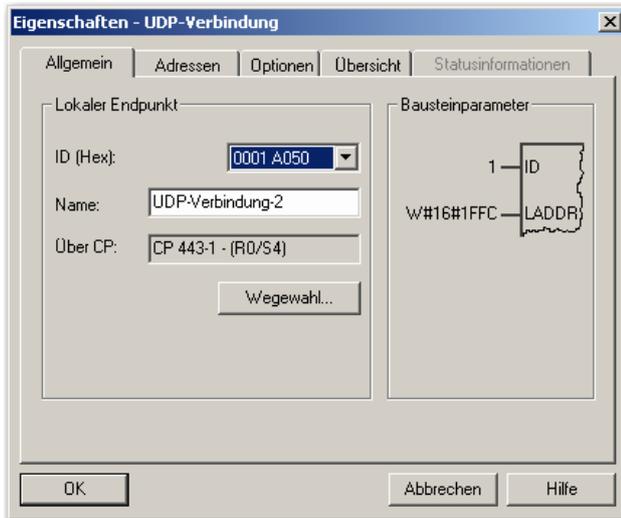
#### Register

Für UDP-Verbindungen stehen folgende Register zur Verfügung:

- Allgemein  
Anzeige von Parametern, welche die Verbindung identifizieren.
- Adressen  
Anzeige der lokalen und fernen Adressinformationen.
- Optionen  
In diesem Register können keine Einstellungen vorgenommen werden.
- Übersicht  
Übersicht aller projektierten UDP-Verbindungen der selektierten Station mit den entsprechenden Parametern und dem Status der Verbindung.
- Statusinformationen  
Diese Informationen entsprechen der Anzeige von NCM Diagnose. Die NCM Diagnose wird von den Saia PCDs nicht unterstützt.

### 3.4.1 Den lokalen Verbindungsendpunkt festlegen

Im Register Allgemein des Eigenschaftsdialogs werden allgemeine Verbindungsparameter angezeigt, die den lokalen Verbindungsendpunkt identifizieren.



3

Parameter	Beschreibung	Zugriff
<b>Lokaler Endpunkt</b>		
ID	Der Eintrag ist identisch mit dem Parameter Lokale ID der Verbindungstabelle. <sup>1)</sup>	wählbar
Name	Beim Anlegen der Verbindung wird hier ein Vorschlagsname angezeigt, der als Suffix eine Verbindungsnummer enthält. Nutzen Sie dieses Feld bei unspezifizierten Verbindungen zur Kennzeichnung des Partners.	änderbar
Über CP	Da die Saia PCDs nur 1 Ethernet-Modul pro PCD unterstützen, macht es keinen Sinn, mehrere CPs pro CPU zu projektieren.	nur lesbar
<b>Bausteinparameter</b>		
ID	Dieser Bausteinaufrufparameter ist beim Aufruf des FC im Anwenderprogramm zur Identifikation der Verbindung zwingend einzutragen (die ID ändert sich bei Änderung der lokalen ID).	nur lesbar
LADDR	Dieser Bausteinaufrufparameter ist beim Aufruf des FC im Anwenderprogramm zur Identifikation der Verbindung zwingend einzutragen.	nur lesbar



<sup>1)</sup> Die PCD2.M157 unterstützt maximal 8 Verbindungen. Auf der PCD2.M157 sind als ID nur Werte zwischen 1 und 8 erlaubt. Wird ein Wert grösser 8 eingetragen, so wird diese Verbindung auf der PCD ignoriert.

### 3.4.2 UDP Adressen festlegen

Bei UDP werden die Kommunikationspartner durch den lokalen und fernen Endpunkt wie folgt adressiert.

Lokale Adressen:

Lokale IP-Adresse und lokaler Port

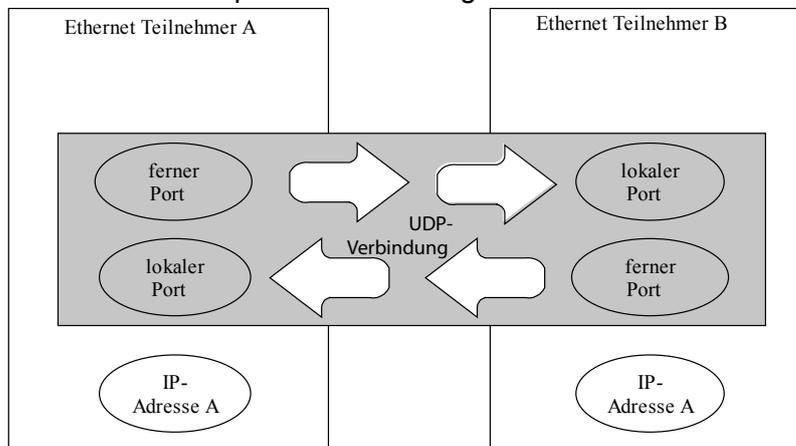
Ferne Adressen:

Ferne IP-Adresse und ferner Port

3



Der Begriff "Verbindung" wird hier auch bei UDP verwendet. Der Grund: Bei der Projektierung werden - wie z.B. auch bei TCP - die Kommunikationspartner einander zugeordnet und somit logisch "verbunden". Tatsächlich erfolgt bei UDP im Betrieb der Stationen kein expliziter Verbindungsaufbau zwischen den Kommunikationspartnern.



Abhängig vom gewünschten Verbindungstyp werden die fernen Adressparameter bei der Projektierung spezifiziert oder offen gelassen.

- spezifizierte UDP-Verbindung

Sie haben beim Neuanlegen der Verbindung eine Zielstation angegeben.

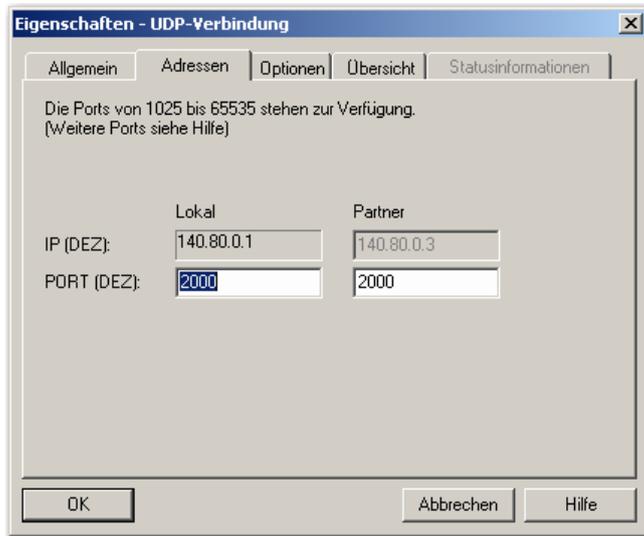
Eine zusätzliche Option stellt die Projektierung von Broadcast zur Verfügung (siehe Folgekapitel).

- unspezifizierte UDP-Verbindung

Sie haben beim Neuanlegen der Verbindung unter Verbindungspartner - Station "un-spezifiziert" angegeben.

## Spezifizierte UDP-Verbindung

Im Register Adressen werden die relevanten lokalen und fernen Adressinformationen als Vorschlagswerte angezeigt. Sie haben die Möglichkeit die Ports individuell einzustellen.



3

## Ports

Die Ports bzw. die Port-Adressen definieren den Zugangspunkt zum Anwenderprogramm innerhalb der Station / CPU. Sie müssen innerhalb der Station / CPU eindeutig sein!



Die Port Adressen 0..1024 sind standardmässig vergeben; Diese sollten nicht verwendet werden. (Well known ports)  
Die Port Adresse 5050 wird für die SBC S-Bus Kommunikation verwendet und steht dem Anwender nicht zur Verfügung.

## Unspezifizierte UDP-Verbindung

Die unspezifizierte UDP-Verbindung kann auf 2 Arten genutzt werden:

- Freie UDP-Verbindung  
Um eine Freie UDP-Verbindung zu projektieren, wählen Sie die Option "Adressvergabe am Baustein". Die Eingabefelder für die ferne IP-Adresse und den fernen Port können dann nicht mehr eingegeben werden, da die Zieladressen jetzt durch das Anwenderprogramm festgelegt werden.
- Verbindung zu einer "Fremdstation" in einem anderen Projekt  
Sie können die ferne IP-Adresse und den Port für eine beliebige Zielstation angeben. Die Zielstation kann innerhalb oder ausserhalb des aktuellen STEP®7-Projekts liegen.



Da bei UDP kein Verbindungsaufbau stattfindet (Datagrammdienst), ist die Kommunikation über die projektierte UDP-Verbindung nur dann möglich, wenn auch die Partner-Adressen (IP-Adresse und Port) spezifiziert sind.



3

### 3.4.3 UDP mit Broadcast und Multicast

Bei der Auswahl des Verbindungspartners haben Sie bei UDP-Verbindungen zusätzlich die beiden Optionen:

- Verbindung zu allen Broadcast-Teilnehmern  
Indem Sie als Partner "alle Broadcast-Teilnehmer" wählen, legen Sie fest, dass UDP-Telegramme an alle erreichbaren Broadcast-Teilnehmer gesendet werden.
- Verbindung zu allen Multicast-Teilnehmern  
Multicast-Telegramme werden von den Saia PCD Steuerungen nicht unterstützt.

Die Telegramme werden ohne Quittung versendet, da das Protokoll UDP keine Quittierungen vorsieht. Dies ist so vorgesehen, um "Datenfluten" durch Quittungen zu verhindern. Werden beispielsweise Telegramme an 100 Partner versendet, würden jetzt 100 Quittungen (pro Partner 1) gleichzeitig ankommen. Solche Datenfluten könnten von der Senderbaugruppe nicht ausgewertet werden.

Oft besteht der Wunsch von einer Station aus Telegramme an eine Vielzahl von Partnerstationen zu verschicken. Wichtig ist hierbei, dass die Telegramme gleichzeitig verschickt werden und auch nahezu gleichzeitig bei den Partnerstationen ankommen. Es wird daher immer ein Versenden und Empfangen von Broadcast-Telegrammen gefordert. Bei einer Broadcast-Nachricht wird das Telegramm auch wirklich von allen Teilnehmern am Netz aufgenommen.

Ein typischer Anwendungsfall ist, dass Broadcast-Telegramme zum Suchen einer MAC-Adresse zu einer IP-Adresse benötigt (ARP-Request) werden.

Daher muss eine Kommunikationsbaugruppe Broadcasttelegramme generell aufnehmen und softwaremässig auswerten. Ein gravierender Nachteil besteht hierbei darin, dass bei zu vielen Broadcasttelegramme auf dem Netz die Performance deutlich sinkt. Dies liegt daran, dass die einzelnen Baugruppen alle Broadcasttelegramme bearbeiten müssen, um festzustellen, ob diese für sie bestimmt waren.

Um die genannten Nachteile zu vermeiden, sollte die Broadcastkommunikation wohl überlegt eingesetzt werden.

### Verbindung zu allen Broadcast-Teilnehmern

Indem Sie als Verbindungspartner "alle Broadcast-Teilnehmer" wählen, legen Sie fest, dass UDP-Telegramme an alle erreichbaren Broadcast-Teilnehmer gesendet werden.

Im Register "Adressen" wird unter der IP-Adresse (IP) für den Partner eine im Netz gültige Broadcast-Adresse vorgeschlagen.

Unter PORT müssen Sie eine für alle zu erreichenden Partner zutreffende PORT Adresse eintragen.

3



### Verbindung zu allen Multicast-Teilnehmern

Diese Verbindungsart wird von den Saia PCD Steuerungen nicht unterstützt!

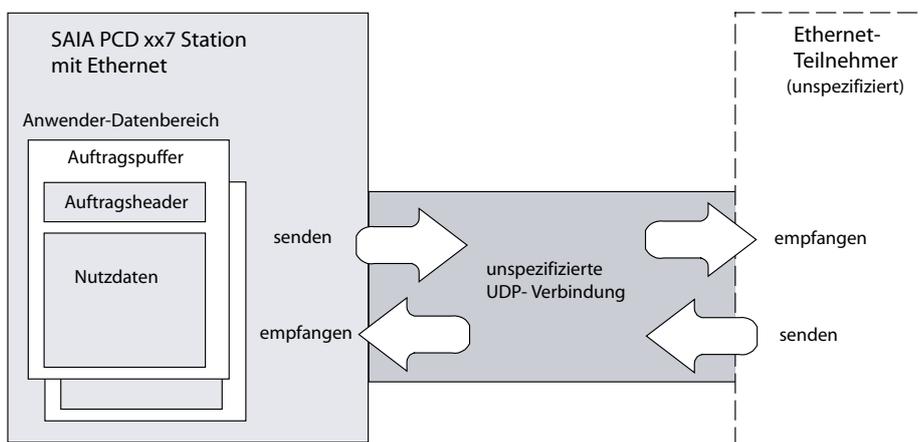
### 3.4.4 Freie UDP-Verbindung

#### Programmgesteuerte Adressierung

Eine Freie UDP-Verbindung ermöglicht die programmgesteuerte Adressierung des Kommunikationspartners. Die Kommunikation zwischen zwei Teilnehmern an Industrial Ethernet hat folgende Eigenschaften:

- Der Datentransfer ist bidirektional, d.h. auf der UDP-Verbindung kann gleichzeitig gesendet und empfangen werden.
- Der lokale Teilnehmer ist per Projektierung festgelegt. Der ferne Teilnehmer wird vom Anwenderprogramm beim AG\_SEND Aufruf im Auftragsheader des Auftragspuffers eingetragen. Damit kann jeder beliebige Teilnehmer am Ethernet/LAN/WAN erreicht werden.
- Aus dem Auftragsheader des AG\_RECV kann die IP-Adresse und der Port des Senders gelesen werden.

3



Pro Auftragspuffer können bis zu 738 Byte Nutzdaten übertragen werden. Der Auftragsheader belegt zusätzlich 6 Byte.

### 3.5 Betriebsart FETCH/WRITE

Die FETCH/WRITE-Dienste ermöglichen den direkten Zugriff auf System Speicherbereiche in der Saia PCD xx7 von SIMATIC, SIMATIC PC-Stationen oder von Fremdgeräten aus:

- FETCH: Daten direkt lesen
- WRITE: Daten direkt schreiben

#### Verbindungstypen

Die FETCH/WRITE-Dienste können bei Saia PCDs (nur PCD2.M487 und PCD3) auf folgenden Verbindungstypen projiziert und genutzt werden:

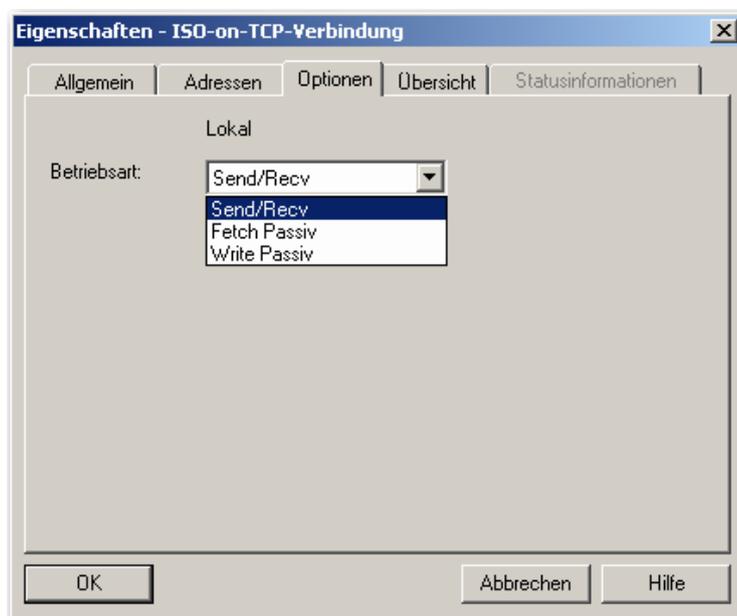
- TCP-Verbindungen

#### Projektierung

Je nach Stationstyp sind folgende Betriebsarten für den Verbindungsendpunkt projektierbar:

- Saia PCD-Station: FETCH PASSIV / WRITE PASSIV

Indem Sie eine der Betriebsarten FETCH PASSIV bzw. WRITE PASSIV für die Transportverbindung wählen, können direkte Zugriffe auf die System Speicherbereiche in der Saia PCD von einer SIMATIC -Station oder einer Fremdstation (unspezifizierte Verbindung) aus erfolgen.



Die Verbindung kann dann ausschliesslich für diese Betriebsart genutzt werden. Senden oder Empfangen über die FCs AG\_SEND/AG\_LSEND bzw. AG\_RECV/AG\_LRECV ist dann nicht möglich.

Der Verbindungsaufbau erfolgt passiv, d.h. nur die Partnerstation (SIMATIC-Station, eine PC-Station oder eine Fremdstation) kann die Verbindung aufbauen. Die entspre-

chende Einstellmöglichkeit im Register "Allgemein" wird automatisch gesetzt und ist nicht mehr bedienbar.

### Systemspeicher

Auf folgende Operandenbereiche im Systemspeicher der SIMATIC S7 können Sie über FETCH oder WRITE zugreifen:

- Datenbausteine (DB) (beachten Sie für den DB-Zugriff folgende Beschränkung: die höchste DB-Nr ist 255)
- Merker (M)
- Prozessabbild der Eingänge (E)
- Prozessabbild der Ausgänge (A)

3

### Kopplung zu Fremdsystemen

Die bei ISO-on-TCP- bzw. TCP-Verbindungen unterstützte Betriebsart für FETCH und WRITE kann prinzipiell von beliebigen Fremdgeräten aus für den Zugriff auf Systemspeicherbereiche genutzt werden.

Damit Sie diesen Zugriff z.B. auch für PC-Anwendungen implementieren können, müssen Sie den PDU-Aufbau für die Aufträge kennen. Die benötigten, S7- bzw. S5-spezifischen Header für Anforderungs- und Quittungstelegramme sind standardmäßig 16 Bytes lang; ihre Struktur in den Siemens Dokumentationen beschrieben.

## 3.6 ISO-on-TCP Verbindung programmieren

Mit der Einführung von neuen Konfigurationsmöglichkeiten, wie DHCP oder Konfigurationsfiles, ist es nicht mehr möglich in NetPro die Verbindungen zu projektieren.

Mit der Systemfunktion SFB 444[Create\_ISO\_CNX] ist es möglich zur Programmlaufzeit eine ISO-on-TCP Verbindung (Server wie Client) zu definieren.

Auf der Support Homepage können diese Systemfunktionen als FB Library mit den korrekten Parameterdefinitionen herunter geladen werden.

Die Parameter des FB Create\_ISO\_CNX sind ähnlich wie die Verbindungsdefinition im NetPro Tool von Siemens.

### Parameter Definition

```
CALL  FB444, DB444
      ConId           := [IN] [INT]
      RemoteIPAddr   := [IN] [DWORD]
      LTsapName      := [IN] [STRING[16]]
      RTsapName      := [IN] [STRING[16]]
      CType          := [IN] [BYTE]
      CPAddr         := [IN] [WORD]
      RetVal         := [OUT] [INT]
```

**ConId:** Verbindungs Identifikation. Derselbe Wert wird beim Sendeaufruf (FC 5 AG\_SEND) bzw. beim Empfangsbaustein (FC6, AG\_RECV) benützt. Jede Verbindung wird durch diese ID spezifiziert, speziell wenn mit NetPro ebenfalls Verbindungen projiziert werden muss darauf geachtet werden, dass er einmalig bleibt. Die Wert 0 ist nicht gültig. Der maximale Wert ist 64.

**RemoteIPAddr:** IP Adresse der Partner Station, mit welcher eine Verbindung aufgebaut werden soll. Der Parameter ist optional, abhängig vom Verbindungstyp. (Siehe auch Verbindungstyp).

**LTsapName:** Lokaler Verbindungsname.

**RTsapName:** Partner Verbindungsname.

**CType:** bestimmt die Art der Verbindung die Aufgebaut werden soll. Siehe Tabelle

CType	Remote IP Addr	Beschreibung
2	0	Erzeugt eine ISO-on-TCP Server Verbindung. Dieser Server akzeptiert den ersten Client der eine Verbindung mit übereinstimmendem lokalem Namen aufbauen will. Alle weiteren Verbindungsaufbauversuche werden abgelehnt, solange die erste Verbindung bestehen bleibt.
2	Any	Erzeugt eine ISO-on-TCP Server Verbindung. Der Server akzeptiert einen Verbindungsaufbau nur vom Client definiert durch die remote IP-Adresse und dem lokalen und remote Verbindungsnamen. Alle andern Verbindungsaufbauversuche werden nicht erlaubt bzw. zurückgewiesen.
3	Any	Erzeugt eine ISO-on-TCP Client Verbindung. Die FW baut eine Verbindung mit dem ISO-on-TCP Server (spezifiziert durch die remote IP Adresse) auf. Sowohl der lokale Verbindungsname wie auch der remote Verbindungsname müssen übereinstimmen, damit die Verbindung aufgebaut werden kann.

**CPAddr:** Adresse des Ethernet CP. Diese Adresse wird entweder in der HW- Konfiguration des entsprechenden CP festgelegt, oder muss im Konfigurationsfile definiert werden.

**RetVal:** Rückgabe Wert der Funktion. Falls die Verbindung angelegt werden konnte ist RetVal = 0. Im Fehlerfall wird eine entsprechende Fehlernummer ausgegeben. (Siehe Tabelle).

RetVal	Beschreibung
0x0000	Die Verbindung konnte erzeugt werden.
0x8A24	Parameter Fehler, ein oder mehrere Parameter enthalten ungültige Werte.
0x8A2C	Keine internen Ressourcen um die Verbindung zu erzeugen.
0x8A2D	Interne Fehler

### 3.7 TCP oder UDP Verbindung programmieren

Mit der Systemfunktion SFB 443[Create\_IP\_CNX] ist es möglich zur Programmlaufzeit eine TCP Verbindung (Server wie Client) und UDP Verbindungen zu definieren.

Auf der Support Homepage können diese Systemfunktionen als FB Library mit den korrekten Parameterdefinitionen herunter geladen werden.

Die Parameter des FB Create\_IP\_CNX sind ähnlich wie die Verbindungsdefinition im NetPro Tool von Siemens.

#### Parameter Definition

```
CALL FB443, DB443
      ConId      := [IN] [INT]
      LocalPort  := [IN] [WORD]
      RemotePort := [IN] [WORD]
      RemoteIPAddr := [IN] [DWORD]
      CType     := [IN] [BYTE]
      SubType   := [IN] [BYTE]
      CAddr     := [IN] [WORD]
      RetVal    := [OUT] [INT]
```

3

**ConId:** Verbindungs Identifikation. Derselbe Wert wird beim Sendeaufruf (FC 5 AG\_SEND) bzw. beim Empfangsbaustein (FC6, AG\_RECV) benützt. Jede Verbindung wird durch diese ID spezifiziert, speziell wenn mit NetPro ebenfalls Verbindungen projiziert werden muss darauf geachtet werden, dass er einmalig bleibt. Der gültige Wertebereich liegt zwischen 1 und 64.

**LocalPort:** Lokale Port Nummer auf welche die IP-Verbindung geöffnet werden. Port Adressen zwischen 0...1024 sollten nicht verwendet werden. Diese sind standardmässig vergeben (well known ports). Normalerweise können für TCP und UDP Verbindungen Port Nummern zwischen 1048 und 65000 gewählt werden. Die entfernte Station muss diesen Wert als RemotePort bei der Verbindungsdefinition angeben.

**RemotePort:** Port Nummer auf welche die IP-Verbindung bei der Partnerstation geöffnet werden soll. Port Adressen zwischen 0...1024 sollten nicht verwendet werden. Diese sind standardmässig vergeben (well known ports). Normalerweise können für TCP und UDP Verbindungen Port Nummern zwischen 1048 und 65000 gewählt werden. Dieser Parameter ist optional, abhängig vom Verbindungstyp. Siehe nachfolgende Tabelle.

**RemoteIPAddr:** IP Adresse der Partner Station, mit welcher eine Verbindung aufgebaut werden soll. Der Parameter ist optional, abhängig vom Verbindungstyp. (Siehe auch Verbindungstyp.

**CType:** Verbindungstyp

- “1”=UDP Verbindung
- “2”=TCP/IP – Server Verbindung
- “3”=TCP/IP – Client Verbindung

**SubType:** Untergruppe von CType (Siehe nachfolgende Tabelle).

CType	SubType	RemotePort/ Remote IPAddr	Beschreibung
0	-	-	Ungültiger Verbindungstyp
1	1	Mandatory	Erzeugt eine spezifizierte UDP Verbindung.
1	5	0 / 0	Erzeugt eine freie UDP Verbindung. Der ferne Teilnehmer wird vom Anwenderprogramm beim AG_SEND Aufruf im Auftragsheader des Auftragspuffers eingetragen. Aus dem Auftragsheader des AG_RECV kann die IP-Adresse und der Port des Senders gelesen werden.
1	Alle Werte ausser 1 und 5	Any	Parameter Fehler
2	0	0 / 0	Erzeugt eine unspezifizierte TCP Server Verbindung. Dieser Server akzeptiert den ersten Client der eine Verbindung auf dem lokalen Port aufbauen will. Alle weiteren Verbindungsaufbauversuche auf dem lokalen Port werden abgelehnt, solange die erste Verbindung bestehen bleibt.
2	0	Mandatory	Erzeugt eine spezifizierte TCP server Verbindung. Der Server akzeptiert nur eine Verbindung vom Client definiert durch die remote IP- Adresse und remote IP-Port. Alle andern Verbindungsaufbauversuche werden nicht erlaubt
2	4	0 / 0	Erzeugt eine unspezifizierte S5 FETCH Passive Verbindung. Dieser Server akzeptiert den ersten Client der eine Verbindung auf dem lokalen Port aufbauen will. Alle weiteren Verbindungsaufbauversuche auf dem lokalen Port werden abgelehnt, solange die erste Verbindung bestehen bleibt
2	4	Mandatory	Erzeugt eine spezifizierte S5 FETCH Passive Verbindung. Der Server akzeptiert nur eine Verbindung vom Client definiert durch die remote IP- Adresse und remote IP- Port. Alle andern Verbindungsaufbauversuche werden nicht erlaubt
2	8	0 / 0	Erzeugt eine unspezifizierte S5 WRITE Passive Verbindung. Dieser Server akzeptiert den ersten Client der eine Verbindung auf dem lokalen Port aufbauen will. Alle weiteren Verbindungsaufbauversuche auf dem lokalen Port werden abgelehnt, solange die erste Verbindung bestehen bleibt.
2	8	Mandatory	Erzeugt eine spezifizierte S5 WRITE Passive Verbindung. Der Server akzeptiert nur eine Verbindung vom Client definiert durch die remote IP- Adresse und remote IP- Port. Alle andern Verbindungsaufbauversuche werden nicht erlaubt.
2	Alle anderen Werte	Any	Parameter Fehler
3	0	Mandatory	Erzeugt eine TCP-IP Client Verbindung zu einem Server definiert durch die remote IP- Adresse und remote IP- Port.
3	Alle Werte ungleich 0	Any	Parameter Fehler

**CPAddr:** Adresse des Ethernet CP. Diese Adresse wird entweder in der HW- Konfiguration des entsprechenden CP festgelegt, oder muss im Konfigurationsfile definiert werden.

**RetVal:** Rückgabe Wert der Funktion. Falls die Verbindung angelegt werden konnte ist RetVal = 0. Im Fehlerfall wird eine entsprechende Fehlernummer ausgegeben. (Siehe Tabelle).

3

RetVal	Beschreibung
0x0000	Die Verbindung konnte erzeugt werden.
0x8A24	Parameter Fehler, ein oder mehrere Parameter enthalten ungültige Werte.
0x8A2C	Keine internen Ressourcen um die Verbindung zu erzeugen.
0x8A2D	Interne Fehler

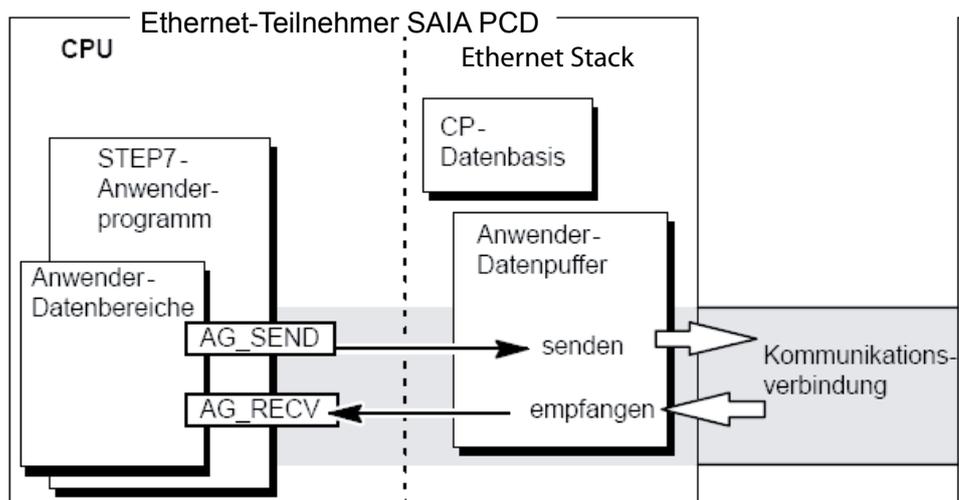
## 4 SEND/RECEIVE- Schnittstelle im Anwenderprogramm

### 4.1 Arbeitsweise der SEND/RECEIVE-Schnittstelle in der CPU

Für die Abwicklung der Kommunikation über Verbindungen stehen folgende Bausteine vom Typ FC zur Verfügung:

- **AG\_SEND/AG\_LSEND**  
Der Baustein übergibt die Nutzdaten aus dem angegebenen Anwender-Datenbereich zur Übertragung an den Ethernet-CP.
- **AG\_RECV/AG\_LRECV**  
Der Baustein übernimmt die empfangenen Nutzdaten in den im Aufruf angegebenen Anwender-Datenbereich.

Die untenstehende Darstellung verdeutlicht den Sachverhalt: Über die FC-Bausteine AG\_SEND/AG\_LSEND und AG\_RECV/AG\_LRECV beauftragt das Anwenderprogramm den Ethernet-Stack, auf der projektierten Verbindung Daten zu senden oder zu empfangen.



Die Saia PCD mit Ethernet kann über eine Verbindung pro Auftrag folgende Datenmengen übertragen:

	ISO-on-TCP	TCP	UDP
Senden	8192	8192	744
Empfangen	8192	8192	744

## 4.2 SEND/RECEIVE-Schnittstelle programmieren

### Prinzip der Auftrags- und Datenübergabe

Das Anwenderprogramm stösst durch die FC-Bausteinaufrufe die Übertragung der Anwender-Datenbereiche an und überwacht die Ausführung, indem es die Anzeigen der FCs auswertet.

Unter anderem werden beim Aufruf der FC-Bausteine folgende Parameter übergeben:

- die Verbindungsnummer der Verbindung (ID);
- die Lage des Anwender-Datenbereichs in der CPU.

### Aufgabe der FC-Bausteine

Der Aufruf der FC-Bausteine bewirkt folgende Aktion:

- Der Anwender-Datenbereich wird an den Ethernet-Stack übergeben bzw. wird vom Ethernet-Stack übernommen.
- Die Ausführung des Auftrags wird im Status positiv oder negativ bestätigt.

### Gehen Sie so vor

Programmieren Sie die SEND/RECEIVE-Schnittstelle im Anwenderprogramm wie folgt:

1. Verwenden Sie zur Datenübertragung mit Verbindungen folgende FC-Bausteine:

- AG\_SEND / AG\_LSEND zur Übergabe des Anwender-Datenbereichs an den Ethernet-Stack;
- AG\_RECV / AG\_LRECV zur Übernahme der vom Ethernet-Stack empfangenen Daten in den Anwender-Datenbereich;

2. Werten Sie die Anzeigen der FC-Bausteine aus:

- bei AG\_SEND / AG\_LSEND die Parameter DONE, ERROR, STATUS;
- bei AG\_RECV / AG\_LRECV die Parameter NDR, ERROR, STATUS;



Verbindungsnummern (IDs) müssen bei der Programmierung von der Projektierung übernommen werden.



Um eine korrekte Parametrierung der Bausteinaufrufe zu gewährleisten, bietet STEP®7 im KOP/AWL/FUP-Editor die Möglichkeit, sämtliche relevanten Parameter aus der Hardware-Konfiguration (HWKonfig) und aus der Verbindungsprojektierung automatisch zu übernehmen.

### 4.3 Allgemeine Hinweise zu den FCs / FBs

Die hier beschriebenen Funktionen (Bausteine vom Typ FC) und FBs (Funktionsbausteine) werden, soweit nicht anders angegeben, zusammen mit dem Basispaket STEP®7 geliefert.

Die Bausteinnummern können von Ihnen geändert werden.

#### Welche Bausteinversion verwenden?

Die aktuellen PCDs unterstützen nur die Bausteine für die S7-400 von Siemens.

### 4.4 FC-Aufrufe parametrieren

Bevor die FCs im Detail beschrieben werden, folgen an dieser Stelle einige allgemeine Hinweise zum Aufruf und zur Parametrierung der FCs.

#### 4.4.1 Parameter zur CP- und Verbindungszuordnung (Eingangsparameter)

Beim Aufruf eines FC-Bausteines übergeben Sie im Parameter CPLADDR bzw. LADDR die Baugruppen-Anfangsadresse des Ethernet-CP. Die Baugruppen-Anfangsadresse des Ethernet-CP ist dem Eigenschaftendialogfeld des CP im Register "Adresse/Eingang" zu entnehmen (wählbar im SIMATIC Manager oder in HWKonfig).

Bei verbindungsorientierten Aufträgen müssen Sie zusätzlich die zu verwendende Verbindung über deren Verbindungs-ID referenzieren. Die Angabe hierzu finden Sie im Eigenschaftendialog der Verbindung unter "Bausteinparameter".

#### 4.4.2 Parameter zur Angabe eines CPU-Datenbereichs (Eingangsparameter)

Beim Aufruf eines FC-Bausteines übergeben Sie die Adresse und Länge des Datenbereiches in der CPU, in dem Nutzdaten bereitgestellt oder abgelegt werden sollen oder weitere Parametrierinformationen enthalten sein können. Zur Adressierung dieses Bereiches wird der Datentyp ANY-Zeiger verwendet.

#### 4.4.3 Statusinformationen (Ausgangsparameter)

Für die Statusauswertung sind im Anwenderprogramm die Parameter auszuwerten:

- **DONE bzw. NDR**  
Diese Parameter (DONE bei Sendeaufträgen bzw. NDR bei Empfangsaufträgen) melden den (positiven) Abschluss einer Auftragsausführung.
- **ERROR**  
Meldet, wenn der Auftrag nicht fehlerfrei ausgeführt werden konnte.
- **STATUS**  
Der Parameter liefert Detailinformation zur Auftragsausführung. Statusanzeigen können bereits während der Auftragsausführung geliefert werden (DONE=0 und ERROR=0).



Beachten Sie, dass die Statusanzeigen DONE, NDR, ERROR, STATUS bei jedem Bausteinaufruf aktualisiert werden.

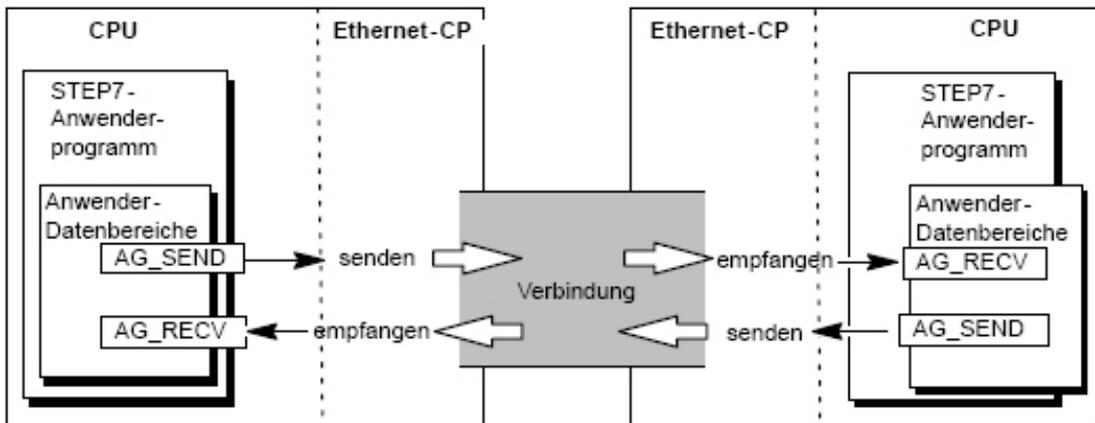
**4.5 FCs für die SEND/RECEIVE-Schnittstelle**

Für die Übertragung von Daten an der SEND/RECEIVE-Schnittstelle stehen folgende FCs zur Verfügung:

- AG\_SEND (FC5) für Daten senden
- AG\_RECV (FC6) für Daten empfangen
- AG\_LSEND (FC50) für Daten senden
- AG\_LRECV (FC60) für Daten empfangen

Bei den FC AG\_SEND / AG\_RECV ist die Datenlänge pro Auftrag auf <=240 Byte beschränkt. Längere Datensätze (bis zu 8192 Byte) können mit Hilfe der FCs AG\_LSEND oder AG\_LRECV übertragen werden.

Die folgende Darstellung verdeutlicht die Anwendung der hier beschriebenen FCs für den bidirektionalen Datentransfer über eine projektierte Verbindung.



Beim Aufruf eines FC-Bausteines übergeben Sie Adresse und Länge des Datenbereiches in der CPU. Beachten Sie bitte, dass die maximale Länge des Datenbereiches vom verwendeten Bausteintyp abhängig ist.

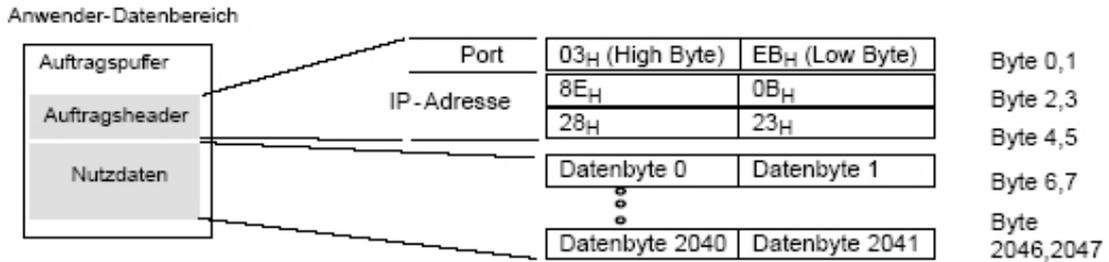
FC	Iso-on-TCP	TCP	UDP
AG_SEND	240	240	240
AG_LSEND	8192	8192	744
AG_RECV	240	240	240
AG_LRECV	8192	8192	744

**4.5.1 Datenbereich für UDP Verbindungen**

Bei spezifizierter Verbindung sind Adress- und Auftragsparameter durch die Verbindungsprojektion festgelegt. Das Anwenderprogramm stellt daher nur die Nutzdaten im UDP-Datenbereich beim Senden mit AG\_SEND / AG\_LSEND bereit, bzw. empfängt diese mit AG\_RECV / AG\_LRECV.

Freie UDP-Verbindungen erfordern einen Auftragsheader im Anwender-Datenbereich. Entnehmen Sie der folgenden Darstellung die Struktur des Auftragspuffers und

die Bedeutung und Plazierung der Parameter im Auftragsheader.



Im Bild (Eingaben sind hexadezimal) wird beispielhaft von folgender IPAdresse ausgegangen: 142.11.40.35;

Für die Portadresse 1003 ist z.B. einzutragen: für High Byte: 03H; für Low Byte: EBH.

Der Anwender-Datenbereich kann bis zu 744 Byte umfassen. Es können bis zu 738 Byte Nutzdaten übertragen werden. 6 Byte sind für den Auftragsheader reserviert.

Beachten Sie, dass die beim Bausteinaufruf angegebene Datenlänge (Parameter LEN) den Header und die Nutzdaten umfassen muss!

**4.5.2 FC5 AG\_SEND / FC50 AG\_LSEND**

Der FC-Baustein AG\_SEND / AG\_LSEND übergibt Daten an den Ethernet-Stack zur Übertragung über eine projektierte Verbindung.

Der angegebene Datenbereich kann ein PA-Bereich, ein Merkerbereich oder ein Datenbausteinbereich sein.

Eine fehlerfreie Ausführung wird signalisiert, wenn der gesamte Anwenderdatenbereich dem Ethernet Stack übergeben werden konnte.

Beispiel: Aufruf in AWL

```

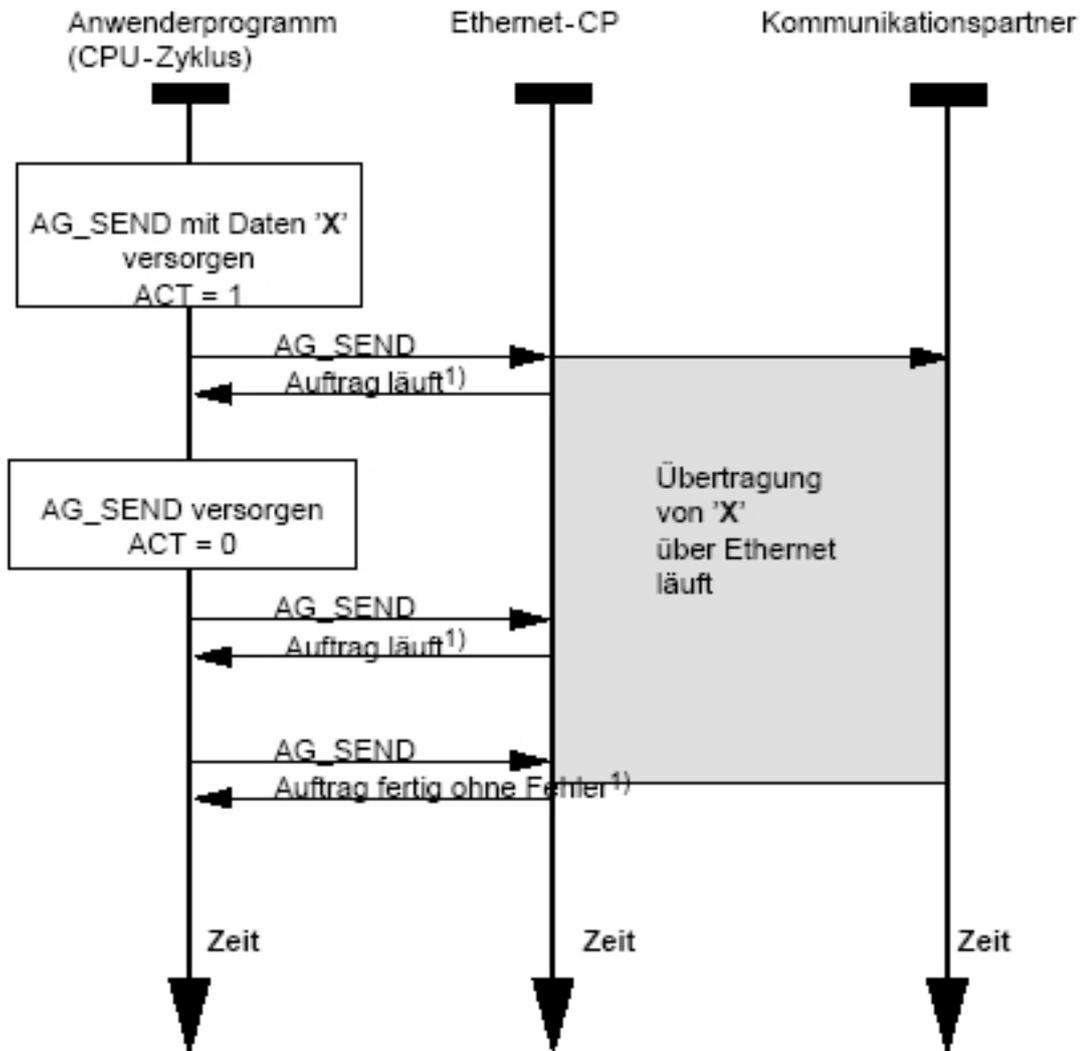
call fc 5 //AG_SEND / AG_LSEND Bausteinaufruf
ACT := M 10.0 //Auftragsanstoss über Merkerbit
ID := MW 12 //Verbindungs-ID gemäss Projektierung
LADDR := W#16#1FFC //LADDR 8188 dez.in Hardware-Konfiguration
SEND := P#db99.dbx10.0 byte 240 //Puffer mit Sendedaten
LEN := MW 14 //Längenangabe für Sendedaten
DONE := M 10.1 //Ausführungsanzeige
ERROR := M 10.2 //Fehleranzeige
STATUS := MW 16 //Statusanzeige
    
```

**Arbeitsweise**

Der Sendeauftrag wird ausgeführt, sobald der Parameter ACT = 1 übergeben wird. Anschliessend müssen Sie in mindestens einem weiteren Aufruf den Parameter ACT = 0 übergeben.

Die Statusanzeige in den Ausgabeparametern DONE, ERROR und STATUS wird bei jedem Bausteinaufruf aktualisiert und kann ausgewertet werden. Zur weiteren Aktualisierung der Statusanzeige ohne erneuten Sendeauftrag ist daher jeweils ein weiterer

Bausteinanruf mit dem Parameter ACT = 0 abzusetzen.



1) Parameterübergabe DONE, ERROR, STATUS

Es ist prinzipiell möglich, den FC innerhalb eines CPU-Zyklus mehrfach aufzurufen. Sie sollten jedoch bedenken, dass dadurch der CPU-Zyklus entsprechend belastet wird (je nach CPU Typ unterschiedlich)!

**Erläuterung der Formalparameter**

Die folgende Tabelle erläutert alle Formalparameter für die Funktion AG\_SEND / AG\_LSEND:

Parameter	Deklaration	Typ	Wertebereich	Bemerkung
ACT	INPUT	BOOL	0,1	<p>Beim FC-Aufruf mit ACT = 1 werden LEN Bytes aus dem mit dem Parameter SEND angegebenen Datenbereich gesendet.</p> <p>Beim FC-Aufruf mit ACT = 0 werden die Statusanzeigen DONE, ERROR und STATUS aktualisiert.</p>
ID	INPUT	INT	1..64 1..8 (M137&M157)	Im Parameter ID wird die Verbindungsnummer der Verbindung angegeben.
LADDR	INPUT	WORD		<p>Baugruppen-Anfangsadresse Bei der Konfiguration des CP mit STEP®7 HWKonfig wird die Baugruppen-Anfangsadresse in der Konfigurationstabelle ausgegeben.</p> <p>Geben Sie diese Adresse hier an.</p>
SEND	INPUT	ANY		<p>Angabe von Adresse und Länge Die Adresse des Datenbereiches verweist alternativ auf:</p> <ul style="list-style-type: none"> <li>- PA-Bereich</li> <li>- Merkerbereich</li> <li>- Datenbausteinbereich</li> </ul>
LEN	INPUT	INT	<p><b>bei ISO-on-TCP / TCP:</b> 1..8192</p> <p><b>Bei UDP:</b> 1..744</p>	<p>Anzahl der Bytes, die mit dem Auftrag aus dem Datenbereich gesendet werden sollen. Die Angabe kann im Bereich von 1 bis "Längenangabe beim Parameter SEND" liegen.</p> <p>Mit dem FC AG_SEND ist der Datenbereich generell auf maximal 240 Byte beschränkt.</p> <p>Erhöhte Performance bei kurzen Datensätzen: Die Übertragung von Datensätzen bis zu 240 Byte erfolgt mit erhöhter Performance! Dies gilt unabhängig vom verwendeten Bausteintyp.</p>
DONE	OUTPUT	BOOL	0: Auftrag läuft 1: Auftrag ausgeführt	Der Zustandsparameter zeigt an, ob der Auftrag fehlerfrei abgewickelt wurde. Solange DONE = 0 kann kein weiterer Auftrag angestossen werden. DONE wird vom CP bei Auftragsannahme auf 0 gesetzt Zur Bedeutung im Zusammenhang mit den Parametern ERROR und STATUS siehe nachfolgende Tabelle.

Parameter	Deklaration	Typ	Wertebereich	Bemerkung
ERROR	OUTPUT	BOOL	0: - 1: Fehlerfall	Fehleranzeige Zur Bedeutung im Zusammenhang mit den Parametern DONE und STATUS siehe nachfolgende Tabelle.
STATUS	OUTPUT	WORD	siehe nachfolgende Tabelle	Statusanzeige Zur Bedeutung im Zusammenhang mit den Parametern DONE und ERROR siehe nachfolgende Tabelle.

## Anzeigen

Die folgende Tabelle informiert über die vom Anwenderprogramm auszuwertende Anzeige, gebildet aus DONE, ERROR und STATUS.

DONE	ERROR	STATUS	Bedeutung
1	0	0000 <sub>H</sub>	Auftrag fertig ohne Fehler.
0	0	0000 <sub>H</sub>	Kein Auftrag in Bearbeitung.
0	0	8181 <sub>H</sub>	Auftrag läuft.
0	1	7000 <sub>H</sub>	Der FC wurde mit ACT=0 aufgerufen; der Auftrag wird jedoch nicht bearbeitet.
0	1	8183 <sub>H</sub>	Die Projektierung fehlt oder der ISO-/TCP-Dienst in der Ethernet-PCD ist noch nicht gestartet.
0	1	8185 <sub>H</sub>	Parameter LEN grösser als Quell-Bereich SEND.
0	1	8186 <sub>H</sub>	Parameter ID ungültig. ID != 1,2,...64.
0	1	8304 <sub>H</sub>	Die Verbindung ist nicht aufgebaut. Der Sendeauftrag sollte erst nach einer Wartezeit >100 ms erneut abgesetzt werden.
0	1	8311 <sub>H</sub>	Zielstation ist unter der angegebenen Ethernet-Adresse nicht erreichbar.
0	1	8F22 <sub>H</sub>	Quell-Bereich ungültig. z.B.: Bereich im DB nicht vorhanden Parameter LEN < 0
0	1	8F24 <sub>H</sub>	Bereichsfehler beim Lesen eines Parameters.
0	1	8F28 <sub>H</sub>	Ausrichtungsfehler beim Lesen eines Parameters.
0	1	8F32 <sub>H</sub>	Parameter enthält zu grosse DB-Nummer.
0	1	8F33 <sub>H</sub>	DB-Nummer Fehler.
0	1	8F3A <sub>H</sub>	Bereich nicht geladen (DB).
0	1	8F44 <sub>H</sub>	Adresse des zu lesenden Parameters in der Zugriffsspur gesperrt.
0	1	8F7F <sub>H</sub>	Interner Fehler. z.B. unzulässige ANY-Referenz z.B. Parameter LEN = 0
0	1	8090 <sub>H</sub>	Baugruppe mit dieser Baugruppen-Anfangsadresse nicht vorhanden oder CPU in STOP.
0	1	8092 <sub>H</sub>	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
0	1	80B0 <sub>H</sub>	Baugruppe kennt den Datensatz nicht.
0	1	80B1 <sub>H</sub>	Die Längenangabe (im Parameter LEN) ist falsch.
0	1	80C0 <sub>H</sub>	Datensatz kann nicht gelesen werden.

DONE	ERROR	STATUS	Bedeutung
0	1	80C1 <sub>H</sub>	Der angegebene Datensatz ist gerade in Bearbeitung.
0	1	80C2 <sub>H</sub>	Es liegt ein Auftragsstau vor.
0	1	80C3 <sub>H</sub>	Die Betriebsmittel (Speicher) der CPU sind temporär belegt.
0	1	80C4 <sub>H</sub>	Kommunikationsfehler (tritt temporär auf; daher ist eine Wiederholung im Anwenderprogramm sinnvoll.)
0	1	80D2 <sub>H</sub>	Baugruppen-Anfangsadresse ist falsch.

### Beispiel für AG\_SEND

4

Nachfolgend finden Sie ein ablauffähiges Beispiel für den Aufruf und die Parameterauswertung eines FC 5 (AG\_SEND).

Zum hier gewählten FC 100, in dem der Sendeaufruf erfolgt, gehört der unten aufgeführte OB100; der OB100 stellt das ACT-Bit beim Anlauf der CPU korrekt ein.

Zur korrekten Funktion muss ein DB100 mit der Grösse von mindestens 240 Byte geladen sein.

Das Programm setzt einen CP auf Adresse 8188, sowie eine projektierte Verbindung vom Typ ISO-on-TCP oder TCP mit der ID=1 voraus (bitte ggf. anpassen !).

```
//-----
FUNCTION FC 100: VOID
TITLE = SENDE_DEMO
AUTHOR : Tester
FAMILY : SBC
NAME : FC5_Demo
VERSION : 1.0
//-----
BEGIN
CALL FC 5 (
ACT := M100.0,
ID := 1,
LADDR := W#16#1FFC,
SEND := P#DB100.dbx0.0 BYTE 240,
LEN := 240,
DONE := M100.1,
ERROR := M100.2,
STATUS := MW102 );

//-----
R M100.0;           //Reset des Parameters ACT für alle weiteren
SET;                //Aufrufe von FC 5;
U M100.1;
SPB done;          //Testen, ob DONE = TRUE
SET;
U M100.2;
SPB err;           //Testen, ob ERROR = TRUE

//-----
BEA;                //Weder DONE noch ERROR sind gesetzt; der Auftrag
//läuft noch.

//-----
done: S M100.0;     //Auftrag ist ohne Fehler beendet. Setze ACT =
BEA;                //TRUE, damit der folgende Aufruf den neuen Auftrag
```

```

//anstossen kann.

//-----
err:  NOP 1;           //Ein Fehler ist aufgetreten. Hier kann eine
      NOP 1;           //Auswertung des Statuswortes erfolgen. Auf jeden
      S M100.0;        //Fall ACT auf TRUE setzen, damit ein neuer
      BEA;             //Sendeauftrag angestossen werden kann, falls der
                       //Fehler verschwindet.
//-----
END_FUNCTION

ORGANIZATION_BLOCK OB100
TITLE      = Init_for_FC100
FAMILY:    SBC
NAME:      SENDE_DEMO_INIT
VERSION:    1.0

VAR_TEMP

OB1_System: array [1..20] of byte;
END_VAR

BEGIN
  SET
    S M100.0           //Initialisiere den Parameter ACT
END_ORGANIZATION_BLOCK

```

### 4.5.3 FC6 AG\_RECV / FC60 AG\_LRECV

Der FC-Baustein AG\_RECV / AG\_LRECV übernimmt vom Ethernet-Stack die über eine projektierte Verbindung übertragenen Daten. Der für die Datenübernahme angegebene Datenbereich kann ein PA-Bereich, ein Merkerbereich oder ein Datenbausteinbereich sein.

Eine fehlerfreie Ausführung wird signalisiert, wenn die Daten vom Ethernet-Stack übernommen werden konnten.

Beispiel: Aufruf in AWL

```

call fc 6           //AG_RECV / AG_LRECV Bausteinaufruf

ID      := MW 40           //Verbindungs-ID gemäss Projektierung
LADDR   := W#16#1FFC       //=LADDR 8188 dez.in Hardware-Konfiguration
RECV    := P#db9.dbx0.0 byte 240 //Puffer mit Sendedaten
NDR     := M 10.1         //Empfangsanzeige
ERROR   := M 10.2         //Fehleranzeige
STATUS  := MW 16          //Statusanzeige
LEN     := MW 14          //Längenangabe für Empfangsdaten

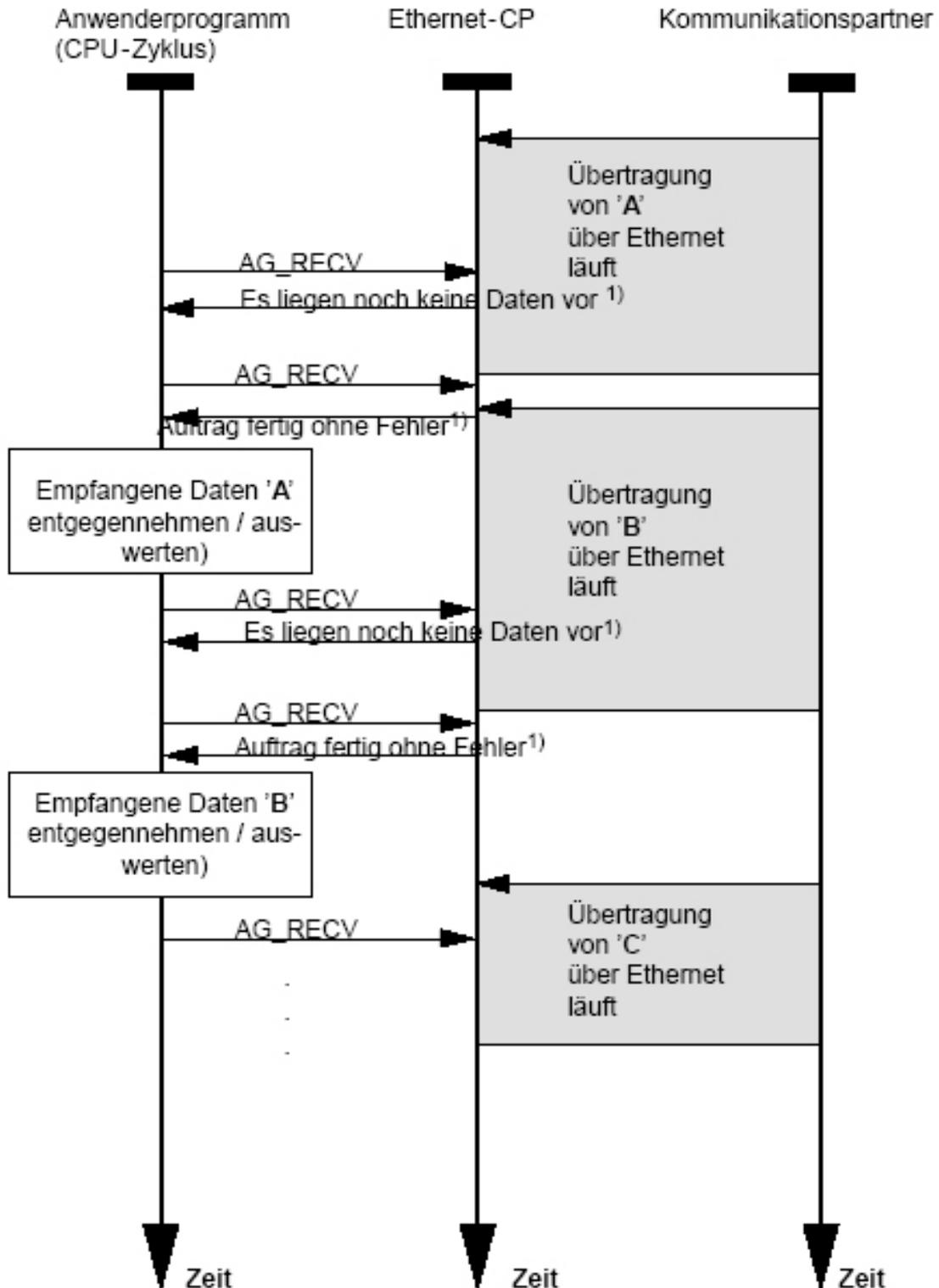
```

#### Arbeitsweise

Die folgende Ablaufdarstellung zeigt den normalen zeitlichen Ablauf einer mit AG\_RECV im Anwenderprogramm angestossenen Datenübernahme.

Mit dem Aufruf des FC stellt das Anwenderprogramm den Puffer für die Empfangsdaten bereit und veranlasst den CP, empfangene Daten dort einzutragen. Sobald ein Datensatz vollständig und konsistent im Empfangspuffer abgelegt wurde, wird dies bei einem der weiteren FC-Aufrufe im Parameter NDR=1 angezeigt.

Die Statusanzeige in den Ausgabeparametern NDR, ERROR und STATUS wird bei jedem Bausteinanruf aktualisiert und kann ausgewertet werden.



## 1) Parameterübergabe NDR, ERROR, STATUS

**Erläuterung der Formalparameter**

Die folgende Tabelle erläutert alle Formalparameter für die Funktion AG\_RECV / AG\_LRECV:

Parameter	Deklaration	Typ	Wertebereich	Bemerkung
ID	INPUT	INT	1..64 1..8 (M137 M157)	Im Parameter ID wird die Verbindungsnummer der Verbindung angegeben.
LADDR	INPUT	WORD		Baugruppen-Anfangsadresse  Bei der Konfiguration des CP mit STEP®7 HWKonfig wird die Baugruppen-Anfangsadresse in der Konfigurationstabelle ausgegeben. Geben Sie diese Adresse hier an.
RECV	INPUT	ANY		Angabe von Adresse und Länge Die Adresse des Datenbereiches verweist alternativ auf:  - PA-Bereich - Merkerbereich - Datenbausteinbereich  Die Übertragung von Datensätzen bis zu 240 Byte erfolgt mit erhöhter Performance, wenn Sie die Längenangabe am Parameter RECV ebenfalls auf 240 beschränken!
NDR	OUTPUT	BOOL	0: - 1: neue Daten	Der Parameter zeigt an, ob neue Daten übernommen wurden. Zur Bedeutung im Zusammenhang mit den Parametern ERROR und STATUS siehe nachfolgende Tabelle.
ERROR	OUTPUT	BOOL	0: - 1: Fehlerfall	Fehleranzeige Zur Bedeutung im Zusammenhang mit den Parametern NDR und STATUS siehe nachfolgende Tabelle.
STATUS	OUTPUT	WORD	siehe nachfolgende Tabelle	Statusanzeige Zur Bedeutung im Zusammenhang mit den Parametern NDR und ERROR siehe nachfolgende Tabelle.
LEN	OUTPUT	INT	<b>bei ISO-on-TCP / TCP:</b> 1..8192 <b>Bei UDP:</b> 1..744	Gibt die Anzahl der Bytes an, die vom Ethernet- Stack in den Datenbereich übernommen wurden.  Mit dem FC AG_SEND ist der Datenbereich generell auf maximal 240 Byte beschränkt.

Die folgende Tabelle informiert über die vom Anwenderprogramm auszuwertende Anzeige, gebildet aus NDR, ERROR und STATUS.

NDR	ERROR	STATUS	Bedeutung
1	0	0000 <sub>H</sub>	Neue Daten übernommen.
0	0	8180 <sub>H</sub>	Es liegen noch keine Daten vor.
0	0	8181 <sub>H</sub>	Auftrag läuft.
0	1	8183 <sub>H</sub>	Die Projektierung fehlt oder der ISO-/TCP-Dienst in der Ethernet-PCD ist noch nicht gestartet.
0	1	8184 <sub>H</sub>	Systemfehler (Der Quelldatenbereich ist fehlerhaft).
0	1	8185 <sub>H</sub>	Ziel- Puffer (RECV) ist zu klein.
0	1	8186 <sub>H</sub>	Parameter ID ungültig. ID != 1,2....64.
0	1	8304 <sub>H</sub>	Die Verbindung ist nicht aufgebaut. Der Sendeauftrag sollte erst nach einer Wartezeit >100 ms erneut abgesetzt werden.
0	1	8F23 <sub>H</sub>	Quell-Bereich ungültig. z.B.: Bereich im DB nicht vorhanden
0	1	8F25 <sub>H</sub>	Bereichsfehler beim Schreiben eines Parameters.
0	1	8F29 <sub>H</sub>	Ausrichtungsfehler beim Schreiben eines Parameters.
0	1	8F32 <sub>H</sub>	Parameter enthält zu grosse DB-Nummer.
0	1	8F33 <sub>H</sub>	DB-Nummer Fehler.
0	1	8F3A <sub>H</sub>	Zielbereich nicht geladen (DB).
0	1	8F7F <sub>H</sub>	Interner Fehler. z.B. unzulässige ANY-Referenz z.B. Parameter LEN = 0
0	1	8090 <sub>H</sub>	Baugruppe mit dieser Baugruppen-Anfangsadresse nicht vorhanden.
0	1	8092 <sub>H</sub>	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
0	1	80B0 <sub>H</sub>	Baugruppe kennt den Datensatz nicht.
0	1	80B1 <sub>H</sub>	Die Längenangabe (im Parameter LEN) ist falsch.
0	1	80C0 <sub>H</sub>	Datensatz kann nicht gelesen werden.
0	1	80C1 <sub>H</sub>	Der angegebene Datensatz ist gerade in Bearbeitung.
0	1	80C2 <sub>H</sub>	Es liegt ein Auftragsstau vor.
0	1	80C3 <sub>H</sub>	Die Betriebsmittel (Speicher) der CPU sind temporär belegt.
0	1	80C4 <sub>H</sub>	Kommunikationsfehler (tritt temporär auf; daher ist eine Wiederholung im Anwenderprogramm sinnvoll.)
0	1	80D2 <sub>H</sub>	Baugruppen-Anfangsadresse ist falsch.

## 5 IP-Konfiguration Programmieren

Es gibt Anwendungsbereiche, bei denen es vorteilhaft ist, die Kommunikationsverbindungen nicht über die Projektierschnittstelle von STEP®7 sondern über eine spezifische Applikation programmgesteuert einzurichten.

Typische Anwendungsfälle finden sich beispielsweise bei Herstellern von Serienmaschinen, die ihren Kunden eine einfache Bedienoberfläche bieten möchten, aber die Kommunikationsdienste an die Bedieneingaben anpassen müssen. Der Endanwender soll keine STEP®7-Kenntnisse benötigen.

Für diese Anwendungen steht ab STEP®7 V5.2 SP1 ein Funktionsbaustein zur Verfügung, der die flexible Übergabe von Datenbausteinen mit Projektierdaten an das Ethernet-Modul ermöglicht.

### 5.1 Übersicht

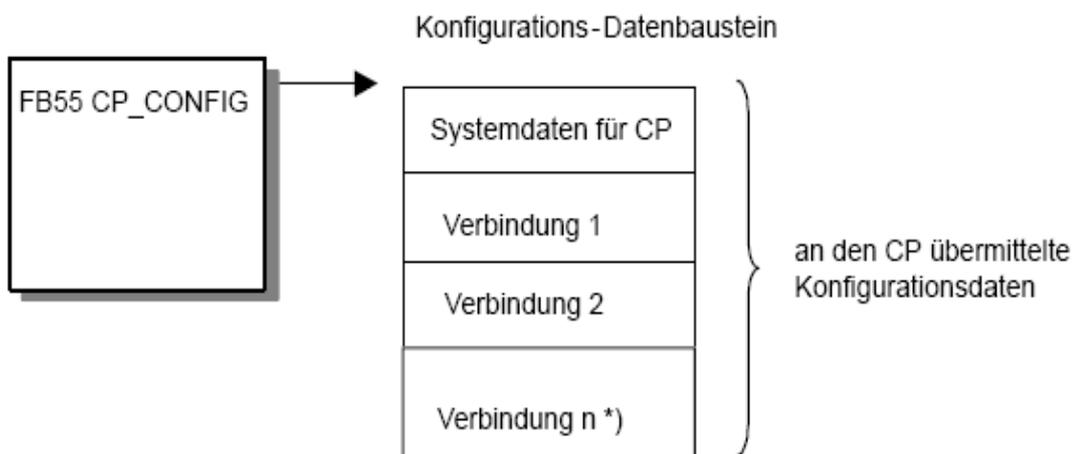
Mit dem FB55 könnten sowohl Verbindungen wie auch Systemdaten für die Ethernet-Module programmiert werden. Mit den SBC xx7 Steuerungen werden aber nur die Systemdaten unterstützt. Verbindungen können nicht programmiert werden. Damit die Systemdaten umprogrammiert werden können, muss ein gültiger Siemens CP-443-1 konfiguriert sein.

#### Anwendungsbereich

Die IP- Konfiguration kann programmgesteuert frei umkonfiguriert werden. Bei den Ethernet-Modulen können keine Verbindungen programmiert werden

#### Prinzip

Über einen Funktionsbaustein, der im Anwenderprogramm aufgerufen wird, können die Konfigurationsdaten an das Ethernet Modul übertragen werden.



\*)  $n_{max} = 64$

Der Konfigurations-DB kann jederzeit in den CP geladen werden. Die aktuellen Verbindungen werden hierbei nicht überschrieben und bleiben bestehen..

Sobald das Anwenderprogramm über den FB55 IP\_CONFIG die Verbindungsdaten übergibt, schaltet die CPU den das Ethernet-Modul kurzzeitig in STOP. Das Modul übernimmt die Systemdaten (inklusive IP-Adresse) und arbeitet diese im Anlauf ab (RUN).

### **Besonderheiten / Einschränkungen**

Konsistenzprüfung nur unter STEP®7: Die Verbindungsprojektierung in STEP®7 ist mit Konsistenzprüfungen verbunden, die bei der programmierten Konfiguration nicht oder nur bedingt möglich sind!

Verbindungsprojektierung beim Partner erforderlich: Bei der Projektierung spezifizierter Verbindungen in STEP®7 wird bei der Projektierung implizit die Verbindung für den Partner angelegt; bei der programmierten Konfiguration ist dies nicht möglich! Hier müssen für die Partner entsprechende Verbindungen projektiert bzw. konfiguriert werden.



HW- Konfiguration erforderlich: Im unterschied zur Orginal Siemens CP- muss eine gültige HW-Konfiguration in die SPS geladen werden, bevor über den FB55 die Systemparameter umprogrammiert werden können.

## **5.2 Konfigurations-Datenbaustein**

### **Bedeutung**

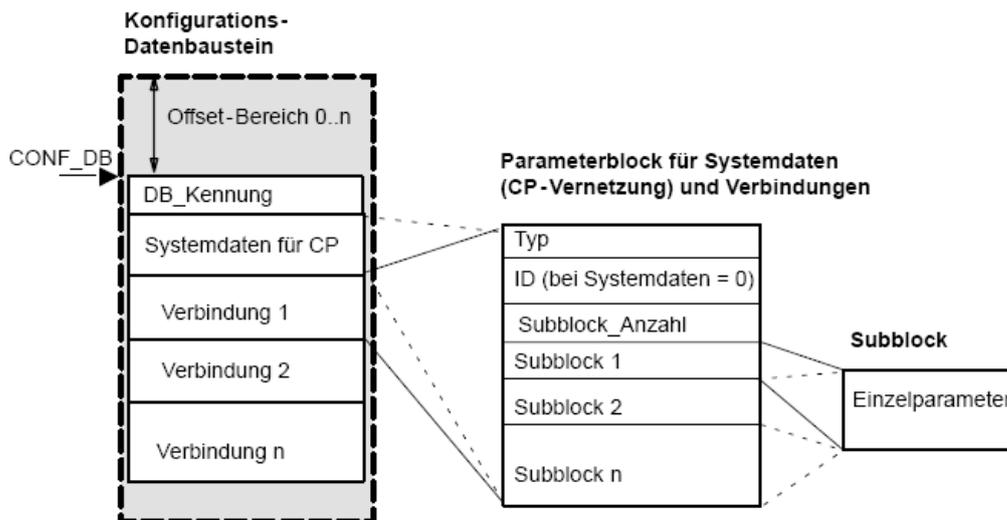
Der Konfigurations-Datenbaustein (CONF\_DB) enthält sämtliche Verbindungsdaten für einen Ethernet-Modul. Der Konfigurations-Datenbaustein wird mit dem Funktionsbaustein FB55 an das Modul übergeben.

### **Aufbau / Datenstruktur**

Sie können der folgenden Darstellung entnehmen:

- Strukturierung durch Parameterblöcke und Subblöcke  
Jede Verbindung sowie besondere Systemdaten werden durch einen identisch aufgebauten Parameterblock beschrieben.  
Einzelne Parameter werden durch Subblöcke typisiert.
- Offset -Bereich

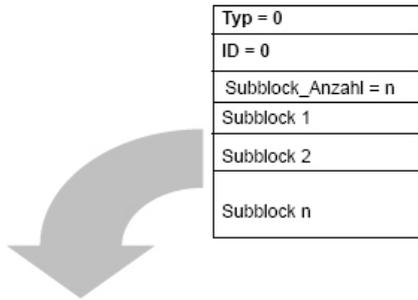
Der CONF\_DB kann über einen Offset-Bereich an beliebiger Stelle innerhalb eines Datenbausteines beginnen.



### 5.2.1 Parameterblock für Systemdaten (IP- Konfiguration)

#### Bedeutung

Sie finden nachfolgend die für die Vernetzung des CPs relevanten Subblöcke. Diese sind im Parameterblock für Systemdaten anzugeben. Je nach Anwendungsfall werden nicht alle Sub-Block-Typen benötigt – entsprechende Angaben finden Sie in der Tabelle.

**Aufbau****Verwendbare Subblöcke**

5

Sub-Block		Parameter	
ID	Typ	Beschreibung	Anwendung (zwingend/option)
1	SUB_IP_V4	Lokale IP-Adresse	z
2	SUB_NETMASK	Lokale Subnet Maske	z
8	SUB_DEF_ROUTER	Default Router Adresse	z

## Beispiel

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	DB_TYP	WORD	W#16#1	Kennung
+2.0	Sysdata_Block	STRUCT		Block 1 for Systemdata
+0.0	Block_Typ	INT	0	0=System data block
+2.0	ID	INT	0	Connection_ID, (For System Data = 0)
+4.0	Subblock_Cnt	INT	3	
+6.0	SUB_IP_V4	STRUCT		Subblock 1 IP_V4
+0.0	Sub_Block_ID	INT	1	Mbr of Subblock
+2.0	Sub_Block_Len	INT	8	Length of Subblocks in Bytes
+4.0	Parameter	STRUCT		IP_Adresse in Hex
+0.0	AdresseByte4	BYTE	B#16#C0	
+1.0	AdresseByte3	BYTE	B#16#6F	
+2.0	AdresseByte2	BYTE	B#16#DE	
+3.0	AdresseBytel	BYTE	B#16#6A	
=4.0		END_STRUCT		
=8.0		END_STRUCT		
+14.0	SUB_NET_MASK	STRUCT		Subblock 2 Subnetmask
+0.0	Sub_Block_ID	INT	2	Mbr of Subblock
+2.0	Sub_Block_Len	INT	8	Length of Subblocks in Bytes
+4.0	Parameter	STRUCT		Subnetmask in Hex
+0.0	AdresseByte4	BYTE	B#16#FF	
+1.0	AdresseByte3	BYTE	B#16#FF	
+2.0	AdresseByte2	BYTE	B#16#FF	
+3.0	AdresseBytel	BYTE	B#16#0	
=4.0		END_STRUCT		
=8.0		END_STRUCT		
+22.0	DEFAULT_Router	STRUCT		Subblock 3 Default Router
+0.0	Sub_Block_ID	INT	8	Mbr of Subblock
+2.0	Sub_Block_Len	INT	8	Length of Subblocks in Bytes
+4.0	Parameter	STRUCT		IP_Adresse of the Default Router in Hex
+0.0	AdresseByte4	BYTE	B#16#C0	
+1.0	AdresseByte3	BYTE	B#16#6F	
+2.0	AdresseByte2	BYTE	B#16#DE	
+3.0	AdresseBytel	BYTE	B#16#6A	
=4.0		END_STRUCT		
=8.0		END_STRUCT		
=30.0		END_STRUCT		
=32.0		END_STRUCT		

### 5.2.2 Parameterblöcke für Verbindungstypen

Die Sub-Block-Typen für Verbindungen werden nicht unterstützt.

#### Sub-Block-Typen

Je nach Parameterblock werden unterschiedliche Parameter benötigt. Jeder Parameter wird durch einen Sub-Block beschrieben. Welche Subblöcke benötigt werden, können Sie den Beschreibungen zu den Systemdaten und zu den Verbindungstypen in den vorhergehenden Kapiteln entnehmen.

Jeder Sub-Block besteht aus dem spezifischen Parameterabschnitt sowie einem Header (4 Byte).

#### Beispiel

Der folgende Auszug aus einem CONF\_DB zeigt den Aufbau eines Sub-Blockes am Beispiel des Sub-Block-Typs SUB\_NETMASK.

	Adresse	Name	Typ	Anfangswert	Kommentar
Header	+14.0	Sub_Block_2	STRUCT		// Subblock 2 Typ SUB_NETMASK
	+0.0	Sub_Block_ID	INT	2	// Subblock ID
	+2.0	Sub_Block_Len	INT	8	// Gesamtlänge des Subblocks in Byte
	+4.0	Parameter	STRUCT		Parameterbereich SUB_NETMASK
Parameter	+0.0	Wert_1	BYTE	B#16#FF	
	+1.0	Wert_2	BYTE	B#16#FF	
	+2.0	Wert_3	BYTE	B#16#FF	
	+3.0	Wert_4	BYTE	B#16#0	
	=4.0	END_	STRUCT		
	=8.0	END_	STRUCT		

Folgende Sub-Block-Typen stehen insgesamt zur Verfügung:

Sub-Block ID <sup>1)</sup>	Sub-Block-Typ	Sub-Block-Länge (in Byte)	Bedeutung des Parameters
1	SUB_IP_V4	4+4	IP- Adresse gemäss IPv4
2	SUB_NETMASK	4+4	Subnetzmaske
8	SUB_DEF_ROUTER	4+4	IP-Adresse des Default Routers

1) Hinweis: nicht aufgeführte ID-Nummern werden derzeit noch nicht genutzt.

### 5.3 FB55 IP\_CONFIG für programmierte Kommunikationskonfiguration

#### Bedeutung des Bausteins

Es lassen sich die IP-Konfiguration in einem DB (Konfigurations-DB) spezifizieren und mittels FB an die Ethernet Stack übertragen.

Je nach Grösse des Konfigurations-DB erfolgt die Übertragung zum Stack in mehreren Segmenten. Sie müssen daher den FB so lange erneut aufrufen, bis der FB mit dem DONE-Bit=1 die vollständige Übertragung signalisiert.

#### Aufruf

```

Call fb 55, DB55           //IP_CONFIG Bausteinaufruf
ACT      := M10.0         //Auftragsanstoss über Merkerbit
LADDR    := W#16#1FFC     //LADDR 8188 des. In Hardware Konfi-
                          //guration
CONF_DB   := P#db99.dbx10.0 Byte 32 //Datenbaustein mit IP Konfi-
                          //guration
LEN       := MW14         //Längenangabe für Verbindungsdaten
DONE      := M10.1       //Ausführungsanzeige
ERROR     := M10.2       //Fehleranzeige
STATUS    := MW16        //Statusanzeige
Ext_Status := MW18       //Fehlerursache in Verbindungsdaten

```

5



Der FB55 ist in der Version 1.0 nicht multiinstanzfähig. In der Simatic Hilfe bzw. den Objekteigenschaften wird das Gegenteil behauptet.

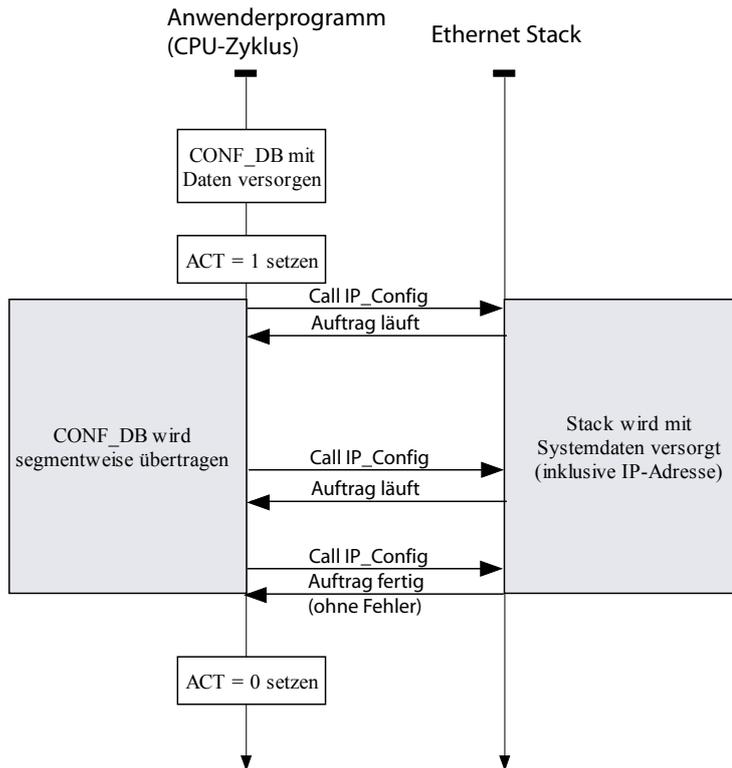
#### Arbeitsweise

Die folgende Ablaufdarstellung zeigt den normalen zeitlichen Ablauf einer mit IP\_CONFIG im Anwenderprogramm angestossenen Verbindungskonfiguration.

Der Auftrag wird ausgeführt, sobald der Parameter ACT = 1 übergeben wird.

Anschliessend müssen Sie den Auftrag wegen der segmentweisen Übertragung des CONF\_DB so lange erneut mit ACT =1 aufrufen, bis der Abschluss mit der entsprechenden Anzeige in den Parametern DONE, ERROR, STATUS angezeigt wird.

Falls zu einem späteren Zeitpunkt erneut eine Verbindungskonfiguration übertragen werden soll, muss zuvor in mindestens einem weiteren Aufruf der Parameter ACT = 0 übergeben werden.



Die mit dem Konfigurations-DB übermittelten Daten werden (falls eine Batterie steckt) spannungsausfallsicher in der SPS abgelegt; sie müssen nach einer Spannungsunterbrechung nicht erneut geladen werden!

**Erläuterung der Formalparameter**

Die folgende Tabelle erläutert alle Formalparameter für die Aufrufschnittstelle des Funktionsbausteins IP\_CONFIG:

Parameter	Deklaration	Typ	Wertebereich	Bemerkung
ACT	INPUT	BOOL	0,1	Beim FB-Aufruf mit ACT = 1 wird der DBxx an die Ethernet Karte gesendet.  Beim FB-Aufruf mit ACT = 0 werden nur die Statusanzeigen DONE, ERROR und STATUS aktualisiert.
LADDR	INPUT	WORD		Baugruppen-Anfangsadresse  Bei der Konfiguration des CP mit STEP®7 HWKonfig wird die Baugruppen-Anfangsadresse in der Konfigurationstabelle ausgegeben. Geben Sie diese Adresse hier an.
CONF_DB	INPUT	ANY		Der Parameter zeigt auf die Anfangsadresse des Konfigurations-Datenbereiches in einem Datenbaustein

Parameter	Deklara-tion	Typ	Wertebereich	Bemerkung
LEN	INPUT	INT		Längenangabe in Byte für den Konfigurations Datenbereich.
DONE	OUTPUT	BOOL	0: - 1: neue Daten	Der Parameter zeigt an, ob der Konfigurations-Datenbereich vollständig übertragen wurde. Beachten Sie, dass der FB je nach Grösse des Konfigurations-Datenbereiches mehrfach angestossen (in mehreren Zyklen) werden muss, bis die Anzeige DONE=1 den Abschluss signalisiert. Zur Bedeutung im Zusammenhang mit den Parametern ERROR und STATUS siehe nachfolgende Tabelle.
ERROR	OUTPUT	BOOL	0: - 1: Fehlerfall	Fehleranzeige Zur Bedeutung im Zusammenhang mit den Parametern NDR und STATUS siehe nachfolgende Tabelle.
STATUS	OUTPUT	WORD	siehe nachfolgende Tabelle	Statusanzeige Zur Bedeutung im Zusammenhang mit den Parametern NDR und ERROR siehe nachfolgende Tabelle.
EXT_Status	OUTPUT	WORD		Der Parameter zeigt bei einer fehlerhaften Auftragsausführung an, welcher Parameter im Konfigurations-DB als Fehlerursache erkannt wurde. High Byte: Index des Parameterblockes Low Byte: Index des Sub-Blockes innerhalb des Parameterblockes

Die folgende Tabelle informiert über die vom Anwenderprogramm auszuwertende Anzeige, gebildet aus DONE, ERROR und STATUS.

DONE	ERROR	STATUS	Bedeutung
1	0	0000 <sub>H</sub>	Auftrag fertig ohne Fehler
0	0	8181 <sub>H</sub>	Auftrag läuft.
0	1	80B1 <sub>H</sub>	Die Anzahl der zu sendenden Daten überschreitet die für diesen Dienst zulässige Obergrenze. (Obergrenze = 16 kByte)
0	1	80D2 <sub>H</sub>	Projektierungsfehler: Die eingesetzte Baugruppe unterstützt diesen Dienst nicht.
0	1	8184 <sub>H</sub>	Systemfehler bzw. unzulässiger Parametertyp. (Datentyp des ANY-Pointers CONF_DB nicht ok) (Aktuell wird nur der Datentyp Byte akzeptiert)
0	1	8185 <sub>H</sub>	Der Wert des Parameters LEN ist grösser als der CONF_DB abzüglich des reservierten Headers (4 Byte) oder die Längenangabe ist falsch.
0	1	8186 <sub>H</sub>	Unzulässiger Parameter erkannt: Der ANY-Pointer CONF_DB verweist nicht auf einen Datenbaustein.
0	1	8B02 <sub>H</sub>	Parameterfehler: Doppelter Parameterblock
0	1	8B03 <sub>H</sub>	Parameterfehler: Der Sub-Block im Parameterblock ist nicht erlaubt.

DONE	ERROR	STATUS	Bedeutung
0	1	8B04 <sub>H</sub>	Parameterfehler: Die Länge, die am FB angegeben wurde, stimmt nicht mit der Länge der Parameterblöcke / Subblöcke überein.
0	1	8B05 <sub>H</sub>	Parameterfehler: Die Länge des Parameterblockes ist ungültig.
0	1	8B06 <sub>H</sub>	Parameterfehler: Die Länge des Sub-Blockes ist ungültig.
0	1	8B07 <sub>H</sub>	Parameterfehler: Die ID des Parameterblockes ist ungültig
0	1	8B08 <sub>H</sub>	Parameterfehler: Die ID des Sub-Blockes ist ungültig
0	1	8B0A <sub>H</sub>	Datenfehler: Der Inhalt des Sub-Blockes ist nicht korrekt.
0	1	8B0D <sub>H</sub>	Datenfehler: Der CONF_DB enthält keinen Parameterblock für Systemdaten.
0	1	8B0E <sub>H</sub>	Datenfehler / Strukturfehler: Der Typ des CONF_DB ist ungültig.
0	1	8B11 <sub>H</sub>	Datenfehler: Der angegebene Typ des Parameterblockes ist ungültig.
0	1	8F22 <sub>H</sub>	Bereichslängenfehler beim Lesen eines Parameters (z.B. DB zu kurz).

## 5.4 DHCP / DNS Konfiguration

Auf den PCD Systemen mit integriertem Ethernet Stack existiert die Möglichkeit, einen DHCP Client zu aktivieren.

Der DHCP Client kann entweder über Konfigurations- Files bzw. im CDB oder über WEB-CGI Befehle konfiguriert werden. Die genauen Konfigurationsdetails werden in den nachfolgenden Kapiteln weiter erläutert.

Damit während des laufenden Betriebes die IP Adresse (oder der Host Name einer Partnerstation konfiguriert werden kann, existieren zwei Systemfunktionen die beim konfigurierten DHCP/DNS Server die entsprechenden Informationen abfragen können.

Da die entsprechenden Systemfunktionen keine Parameter Namen unterstützten existieren entsprechende FB's mit definierten Parameter Namen, welche die erforderlichen Systemfunktionen aufrufen. Diese Funktionen können von der Support Homepage runtergeladen werden.

In den nachfolgenden Kapiteln werden deshalb die entsprechenden FB's beschrieben.

### 5.4.1 DNSQueryByName

Die Funktion FB 441 [DNSQueryByName] verlangt vom DNS Server die IP Adresse von einem Host Namen.

Der FB muss solange aufgerufen werden, bis entweder das Done oder Error Bit gesetzt wird.

Die Funktion arbeitet nur, falls der DNS Server konfiguriert wurde (entweder mit einer Anwender Konfiguration oder über den DHCP Server).

Aufruf:

```
CALL FB441 , DB441
    REQ      := [IN] [BOOL]
    Option   := [IN] [BYTE]
    Hostname := [IN] [STRING[64]]
    IPAddr   := [OUT] [DWORD]
    Done     := [OUT] [BOOL]
    Error    := [OUT] [BOOL]
    RetVal   := [OUT] [INT]
```

Eine positive Flanke am REQ Eingang startet die Abfrage der IP- Adresse.

Wird der Parameter Option auf 0x04 gesetzt, so wird eine Abfrage zum DNS Server erzwungen. Andernfalls wird die zuletzt Empfangene IP- Adresse zurückgegeben.

Der Hostname kann mit oder ohne Domain Informationen spezifiziert werden. (z.B. MyPCD oder MyPCD.test.query.ch)

Folgende Rückgabe-Werte sind möglich:

Done	Error	RetVal	Beschreibung
0	0	0x7001	
0x7002	Die aktuelle Abfrage ist noch nicht beendet.		
0	1	0x8E70	Die aktuelle Abfrage ist noch nicht beendet und das REQ Bit wurde erneut gesetzt.
		0x8E71	Interner Ressourcen Fehler
		0x8E72	Der Hostname konnte nicht aufgelöst werden
		0x8E73	Der DNS Server ist nicht konfiguriert oder Antwortet nicht.
1	0	0	Aktuelle Abfrage wurde erfolgreich abgeschlossen.

Wenn das Done Bit gesetzt ist, wird die Angeforderte IP- Adresse unter IPAddr zurückgegeben. Diese IP- Adresse kann nun für die Verbindungsprojektierung (Programmierung) oder für das Senden von Emails, usw. weiterverwendet werden.

Jede Anfrage wird zwischengespeichert. Wird die Abfrage ein zweites Mal gestartet, so wird im gleichen Aufruf das DONE oder ERROR Bit gesetzt und die gespeicherte IP- Adresse zurückgegeben.

#### 5.4.2 DNSQueryByAddr

Die Funktion FB 442 [DNSQueryByAddr] verlangt vom DNS Server den Host Namen einer IP Adresse.

Der FB muss solange aufgerufen werden, bis entweder das Done oder Error Bit gesetzt wird.

Die Funktion arbeitet nur, falls der DNS Server konfiguriert wurde (entweder mit einer Anwender Konfiguration oder über den DHCP Server).

Aufruf:

```
CALL FB442 , DB442
    REQ      := [IN] [BOOL]
    Option   := [IN] [BYTE]
    IPAddr   := [IN] [DWORD]
    Hostname := [OUT] [STRING[64]]
    Done     := [OUT] [BOOL]
    Error    := [OUT] [BOOL]
    RetVal   := [OUT] [INT]
```

Eine positive Flanke am REQ Eingang startet die Abfrage des Host Namens.

Wird der Parameter Option auf 0x04 gesetzt, so wird eine Abfrage zum DNS Server erzwungen. Andernfalls wird der zuletzt Empfangene Host Name zurückgegeben.

5

Folgende Rückgabe Werte sind möglich:

Done	Error	RetVal	Beschreibung
0	0	0x7001	
0x7002	Die aktuelle Abfrage ist noch nicht beendet.		
0	1	0x8E70	Die aktuelle Abfrage ist noch nicht beendet und das REQ Bit wurde erneut gesetzt.
		0x8E71	Interner Ressourcen Fehler
		0x8E72	Die IP Adresse konnte nicht aufgelöst werden
		0x8E73	Der DNS Server ist nicht konfiguriert oder antwortet nicht.
1	0	0	Aktuelle Abfrage wurde erfolgreich abgeschlossen.

Wenn das Done Bit gesetzt ist, wird der Angeforderte Host Name im String Hostname wie vom DNS Server empfangen zurückgegeben (z.B. MyPCD oder MyPCD.test.query.ch).

Jede Anfrage wird zwischengespeichert. Wird die Abfrage ein zweites Mal gestartet, so wird im gleichen Aufruf das DONE oder ERROR Bit gesetzt und der gespeicherte Hostname zurückgegeben.

## 5.5 Anpassen IP- Konfiguration über WEB- CGI

Über die WEB- CGI Befehle hat man die Möglichkeit, die aktuelle IP- Konfiguration auszulesen, oder die Konfiguration zu ändern.

Geänderte IP- Konfigurationsparameter werden erst nach einem STOP-> RUN Übergang wirksam.

Nach einem erneuten Einschalten der PLC (Power off → On) wird die gespeicherte Konfiguration (SDB, CDB, oder File) übernommen.

Das Web-Interface ist von einem PC aus über Ethernet, oder über USB erreichbar.

Damit die USB Schnittstelle benützt werden kann, muss auf dem PC das WEB-Connect installiert und benutzt werden.

### 5.5.1 Lesen der fixen IP- Konfiguration / Status

Über folgendes Kommando kann die fixe IP-Konfiguration ausgelesen werden:  
<http://<ip addr>/cgi-bin/readVal.exe?CFG-IPC,<parameter name>>

Parameter Name	Beschreibung
IPAddr	Die fixe IP Adresse der PCD.
SubNetmask	Die Subnet Maske des Netzwerkes.
GatewayAddr	Optional: Die Gateway IP Adresse
GarbageTimeout	Die keep-alive time in Sekunden. Damit ein Verbindungsabbruch festgestellt werden kann, wird falls innerhalb dieser Zeit keine Daten ausgetauscht wurden, ein keep-alive Telegramm gesendet. Wird die GarbageTimeout Zeit auf 0 gesetzt, ist der keep-alive Mechanismus ausgeschalten. Min Wert = 0 Max Wert = 3600 Default Wert = 0
CPAddress	Die CPAddress wird benützt wenn Verbindungen erzeugt werden, damit das korrekte Interface angesprochen werden kann. Der Wert muss ungleich "0" sein. Min value = 1 Max value = 0xFFFF (65'535) Default value = 0 (keine Verbindungsprojektierung möglich)

5

Über folgendes Kommando kann der Status der IP- Konfiguration ausgelesen werden:  
<http://<ip addr>/cgi-bin/readVal.exe?SYS-IPC,<parameter name>>

Parameter Name	Beschreibung
CurIPMode	Dieser Parameter zeigt den aktuellen Konfigurationsmechanismus an: "0" wird angezeigt mit NO IP CONFIGURATION "1" wird angezeigt mit HW CONFIGURATION "2" wird angezeigt mit IP CONFIGURATION "3" wird angezeigt mit DHCP CONFIGURATION
CurState	Dieser Parameter stellt den aktuellen Status des DHCP Clients dar: "0" wird angezeigt mit INIT "1" wird angezeigt mit SELECTING "2" wird angezeigt mit REQUESTING "3" wird angezeigt mit BOUND "4" wird angezeigt mit RENEWING "5" wird angezeigt mit REBINDING "6" wird angezeigt mit INIT_REBOOT "7" wird angezeigt mit REBOOTING
CurIPAddr	Unabhängig vom Definierten Mode, wird die aktuelle IP Adresse angezeigt. (Eingestellt über HW-Konfiguration, Konfigurations-File oder gesetzt durch den DHCP Server).
CurGTWAddr	Unabhängig vom Definierten Mode, wird die aktuelle Gateway Adresse angezeigt. (Eingestellt über HW-Konfiguration, Konfigurations-File oder gesetzt durch den DHCP Server).
CurNetMask	Unabhängig vom Definierten Mode, wird die aktuelle Netzwerk Maske angezeigt.

ip addr: Die aktuelle IP Adresse der PCD. Die IP Adresse kann durch den PCD Namen ersetzt werden, falls eine erfolgreiche Registrierung bei einem DNS Server erfolgt ist. Wird die USB Verbindung benützt, so wird anstelle der IP-Adresse localhost/<station> eingegeben. In diesem Fall muss das WEB-Connect Tool benutzt werden.

### 5.5.2 Lesen der DNS – Konfiguration / Status Information

Über folgendes Kommando kann die DNS- Konfiguration ausgelesen werden:  
`http://<ip addr>/cgi-bin/readVal.exe?CFG-IPC,<parameter name>`

Parameter Name	Beschreibung
DNSSVRa	Erste DNS Adresse um Host Informationen anzufordern. Default Einstellung ist 0.0.0.0
DNSSVRb	Zweite DNS Adresse um Host Informationen anzufordern. Default Einstellung ist 0.0.0.0
Domainname	Dieser Parameter spezifiziert den Domain Name des DNS-Servers.

Über folgendes Kommando kann der Status der DNS-Konfiguration ausgelesen werden:  
`http://<ip addr>/cgi-bin/readVal.exe?SYS-IPC,<parameter name>`

5

Parameter Name	Beschreibung
CurDNSSRVa	Unabhängig vom spezifizierten Mode kann die IP-Adresse des erste DNS-Server ausgelesen werden. Es spielt keine Rolle ob die IP-Adresse mit einem Konfigurationsfile oder über den DHCP Server konfiguriert wurde.
CurDNSSRVb	Unabhängig vom spezifizierten Mode kann die IP-Adresse des zweiten DNS-Server ausgelesen werden. Es spielt keine Rolle ob die IP-Adresse mit einem Konfigurationsfile oder über den DHCP Server konfiguriert wurde.
CurDomainname	Unabhängig vom spezifizierten Mode kann der Domain Name ausgelesen werden. Es spielt keine Rolle ob der Domain Name mit einem Konfigurationsfile oder über den DHCP Server konfiguriert wurde.

**ip addr:** Die aktuelle IP-Adresse der PCD. Die IP-Adresse kann durch den PCD Namen ersetzt werden, falls eine erfolgreiche Registrierung bei einem DNS Server erfolgt ist. Wird die USB Verbindung benutzt, so wird anstelle der IP-Adresse localhost/<station> eingegeben. In diesem Fall muss das WEB-Connect Tool benutzt werden.

### 5.5.3 Lesen der DHCP Status Information

Über folgendes Kommando kann der Status der DHCP-Konfiguration ausgelesen werden:  
`http://<ip addr>/cgi-bin/readVal.exe?SYS-IPC,<parameter name>`

Parameter Name	Beschreibung
CurSVRAddr	Falls DHCP konfiguriert ist, zeigt diese Variable die IP-Adresse des aktiven DHCP Servers an.
CurSNTPAddr	Falls DHCP konfiguriert ist, zeigt diese Variable die SNTP Adresse an.

**ip addr:** Die aktuelle IP-Adresse der PCD. Die IP-Adresse kann durch den PCD Namen ersetzt werden, falls eine erfolgreiche Registrierung bei einem DNS Server erfolgt ist. Wird die USB Verbindung benutzt, so wird anstelle der IP-Adresse localhost/<station> eingegeben. In diesem Fall muss das WEB-Connect Tool benutzt werden.

### 5.5.4 Setzen der IP- Konfiguration

Über folgendes Kommando kann die IP-Konfiguration geändert werden:

`http://<ip addr>/cgi-bin/writeVal.exe?CFG-IPC,<parameterName>+<value>`

ParameterName	Beschreibung
IPMode	Um eine fixe IP-Adresse zu setzten, muss <value> auf 2 gesetzt werden.
IPAddr	Die IP-Adresse der PCD. Default value ist 0.0.0.0
SubNetmask	Subnet mask die benützt wird auf dem verbundenen Netzwerk. Default value ist 0.0.0.0
GatewayAddr	Optional: Die Gateway IP-Adresse. Default value ist 0.0.0.0
UpdateSDB	TRUE (1): Beim nächsten STOP/RUN Übergang wird, falls der IPMode Parameter auf 2 gesetzt ist, die aktuelle HW-Konfiguration mit der neuen IP-Konfiguration überschrieben. FALSE (0): Die HW-Konfiguration wird nicht geändert. Beim nächsten STOP/RUN Übergang werden die IP-Parameter wieder mit der aktuellen HW-Konfiguration rückgesetzt. Falls keine HW-Konfiguration in die PCD geladen wurde, hat dieser Parameter keine Auswirkung.

5

**ip addr:** Die aktuelle IP-Adresse der PCD. Die IP-Adresse kann durch den PCD Namen ersetzt werden, falls eine erfolgreiche Registrierung bei einem DNS Server erfolgt ist. Wird die USB Verbindung benützt, so wird anstelle der IP-Adresse localhost/<station> eingegeben. In diesem Fall muss das WEB-Connect Tool benutzt werden.

### 5.5.5 Setzen der DNS – Konfiguration

Über folgendes Kommando kann die DNS- Konfiguration geändert werden:

`http://<ip addr>/cgi-bin/writeVal.exe?CFG-IPC,<parameterName>+<value>`

ParameterName	Beschreibung
DNSSVRa	Erste DNS Adresse um Host Informationen anzufordern. Default value ist 0.0.0.0
DNSSVRb	Zweite DNS Adresse um Host Informationen anzufordern. Default value ist 0.0.0.0
Domainname.	Optionaler Parameter Dieser Parameter spezifiziert den Domain Name des DNS-Servers Falls der Domain Name vom DHCP Server gesetzt wird, hat dieser Parameter keine Auswirkung. Maximale Länge des Namens: 64 Zeichen.

**ip addr:** Die aktuelle IP-Adresse der PCD. Die IP-Adresse kann durch den PCD Namen ersetzt werden, falls eine erfolgreiche Registrierung bei einem DNS Server erfolgt ist. Wird die USB Verbindung benützt, so wird anstelle der IP-Adresse localhost/<station> eingegeben. In diesem Fall muss das WEB-Connect Tool benutzt werden.

### 5.5.6 Setzen der DHCP – Konfiguration

Über folgendes Kommando kann die DHCP- Konfiguration geändert werden:  
`http://<ip addr>/cgi-bin/writeVal.exe?CFG-IPC,<parameterName>+<value>`

Parameter Name	Beschreibung
IPMode	Wird der IPMode auf 3 gesetzt, werden die DHCP Funktionalitäten freigeschalten. Default Value ist 0. Dieser Parameter kann nicht überschrieben werden, falls eine HW- Konfiguration mit gültiger IP- Adresse in die PCD geladen wurde.
DHCPSetGateway	Wird dieses Flag auf 1 gesetzt und der Parameter DHCPGetGW wurde ebenfalls auf "1" gesetzt, so wird die Gateway Adresse automatisch mit der vom DHCP Server empfangenen Gateway Adresse gesetzt. Min value = 0 Max value = 1 Default value = 0
DHCPSetDNSR	Wird dieses Flag auf "1" gesetzt, so sendet die PCD die Information zum DHCP Server, dass ein DNS update erforderlich ist (z.B. mit dem PCD Namen für eine zukünftige Namensauflösung) Auch wenn diese Option gesetzt ist, kann es vorkommen, dass der DNS Server kein Update durchführt. In diesem Fall kann die PCD nicht über ihren Namen erreicht werden, sondern nur über Ihre IP Adresse. Min value = 0 Max value = 1 Default value = 0
DHCPGetDNS	Wird dieses Flag auf 1 gesetzt, so sendet der DHCP Server mit der Antwort auch die DNS IP Adresse zurück. Falls diese Funktion vom DHCP Server unterstützt wird, so wird die vom Anwender definierte DNS IP Adresse überschrieben. Min value = 0 Max value = 1 Default value = 0
DHCPGetGW	Wird dieses Flag auf 1 gesetzt, so sendet der DHCP Server mit der Antwort auch die Gateway Adresse zurück. Falls diese Funktion vom DHCP Server unterstützt wird, so wird die vom Anwender definierte Gateway Adresse überschrieben. Min value = 0 Max value = 1 Default value = 0
DHCPGetDN	Wird dieses Flag auf 1 gesetzt, so sendet der DHCP Server mit der Antwort auch den Domain Name zurück. Falls diese Funktion vom DHCP Server unterstützt wird, so wird der vom Anwender definierte Domain Name überschrieben. Min value = 0 Max value = 1 Default value = 0
Hostname	Dieser Parameter entspricht dem Namen der PCD. Damit diese Information verfügbar wird, muss der DNS Server mit den DHCP Server Informationen abgeglichen werden. Maximale länge des Host Namens beträgt 32 Zeichen.

**ip addr:** Die aktuelle IP-Adresse der PCD. Die IP-Adresse kann durch den PCD Namen ersetzt werden, falls eine erfolgreiche Registrierung bei einem DNS Server erfolgt ist. Wird die USB Verbindung benützt, so wird anstelle der IP-Adresse localhost/<station> eingegeben. In diesem Fall muss das WEB-Connect Tool benutzt werden.

## 5.6 Konfigurationsfile / CDB

### 5.6.1 Konfigurationsfile

Die Saia PCD unterstützt die Möglichkeit Konfigurationen in einem Textfile vorzunehmen und diese in einem Flash Filesystem abzulegen. Beim Aufstarten, wird das Flash Filesystem nach einem Konfigurationsfile "Config.txt" im Verzeichnis <FLASH name>:/Config directory gesucht. Sobald das erste Config.txt File gefunden wird, wird dieses Flash markiert und alle weiteren Konfigurationen von diesem Flash gelesen.

Die IP- Konfiguration kann in den beiden folgenden Files abgelegt werden:

- Config.txt: Muss in einem Flash File System vorhanden sein.
- IPConfig.txt: Dieses File wird für die IP / DNS / DHCP Konfiguration benutzt.

5

Folgendes Format muss verwendet werden:

<parameter name> = <value>

Ein optionaler Kommentar Startet mit dem # Zeichen.

Die Konfigurations-Parameter werden im Kapitel 5.9.3 beschrieben.

### 5.6.2 CDB (Configuration Data Block)

Die Saia PCD unterstützt die Möglichkeit Konfigurationen in einem CDB (Configuration Data Block) zu definieren. Ein CDB ist ein spezieller DB, welcher mit dem Anwenderprogramm geladen wird.

Folgende Eigenschaften muss ein DB erfüllen, damit er als CDB erkannt wird:

- Der DB1, DB511 oder DB1023 können als CDB verwendet werden.
- Der erste Eintrag im CDB ist ein String mit folgendem Text : "SBC xx7 CDB"
- Die Konfigurationen werden als Text in einem String abgelegt. Der String darf länger sein als der eigentliche Text.

Die IP-Konfiguration beginnt immer mit folgendem CDB Eintrag:  
"PARSECDB:RESETCAP"

Die IP Konfiguration endet immer mit folgendem CDB Eintrag:  
"PARSECDB:SETCAP"

Zwischen diesen beiden Strings können IP-Konfigurationen mit folgendem Format gespeichert werden:

CFG-IPC,<parameter name>=<value>

### 5.6.3 Konfigurationsparameter

Parameter Name	Beschreibung
IPMode	2 = IP CONFIGURATION 3 = DHCP CONFIGURATION
IPAddr	Die IP-Adresse der PCD; Default value ist 0.0.0.0
SubNetmask	Subnet mask die auf dem verbundenen Netzwerk benutzt wird. Default value ist 0.0.0.0
GatewayAddr	Optional: Die Gateway IP-Adresse; Default value ist 0.0.0.0
DNSSVRa	Erste DNS Adresse um Host Informationen anzufordern. Default Einstellung ist 0.0.0.0
DNSSVRb	Zweite DNS Adresse um Host Informationen anzufordern. Default Einstellung ist 0.0.0.0

Parameter Name	Beschreibung
Domainname	Optionaler Parameter Dieser Parameter spezifiziert den Domain Name des DNS-Servers. Falls der Domain Name vom DHCP Server gesetzt wird, hat dieser Parameter keine Auswirkung. Maximale Länge des Namens: 64 Zeichen.
DHCPSetGateway	Wird dieses Flag auf 1 gesetzt und der Parameter DHCPGetGW wurde ebenfalls auf "1" gesetzt, so wird die Gateway Adresse automatisch mit der vom DHCP Server empfangenen Gateway Adresse gesetzt. Min value = 0 Max value = 1 Default value = 0
DHCPSetDNSR	Wird dieses Flag auf "1" gesetzt, so führt der DHCP Server, ein DNS update durch (z.B. mit dem PCD Namen für eine zukünftige Namensauflösung) Auch wenn diese Option gesetzt ist, kann es vorkommen, dass der DNS Server kein Update durchführt. In diesem Fall kann die PCD nicht über ihren Namen erreicht werden, sondern nur über Ihre IP-Adresse. Min value = 0 Max value = 1 Default value = 0
DHCPGetDNS	Wird dieses Flag auf 1 gesetzt, so sendet der DHCP Server mit der Antwort auch die DNS IP-Adresse zurück. Falls diese Funktion vom DHCP Server unterstützt wird, so wird die vom Anwender definierte DNS IP Adresse überschrieben. Min value = 0 Max value = 1 Default value = 0
DHCPGetGW	Wird dieses Flag auf 1 gesetzt, so sendet der DHCP Server mit der Antwort auch die Gateway Adresse zurück. Falls diese Funktion vom DHCP Server unterstützt wird, so wird die vom Anwender definierte Gateway Adresse überschrieben. Min value = 0 Max value = 1 Default value = 0
DHCPGetDN	Wird dieses Flag auf 1 gesetzt, so sendet der DHCP Server mit der Antwort auch den Domain Name zurück. Falls diese Funktion vom DHCP Server unterstützt wird, so wird der vom Anwender definierte Domain Name überschrieben. Min value = 0 Max value = 1 Default value = 0
Hostname	Dieser Parameter entspricht dem Namen der PCD. Damit diese Information verfügbar wird, muss der DNS Server mit den DHCP Server Informationen abgeglichen werden. Maximale länge des Host Namens beträgt 32 Zeichen.
GarbageTimeout	Die keep-alive time in Sekunden. Damit ein Verbindungsabbruch festgestellt werden kann, wird falls innerhalb dieser Zeit keine Daten ausgetauscht wurden, ein keep-alive Telegramm gesendet. Wird die Garbage Timeout Zeit auf 0 gesetzt, ist der keep-alive Mechanismus ausgeschaltet. Min Wert = 0 Max Wert = 3600 Default Wert = 0

Parameter Name	Beschreibung
CPAddress	Die CPAddress wird benützt wenn Verbindungen erzeugt werden, damit das korrekte Interface angesprochen werden kann. Der Wert muss ungleich "0" sein. Min value = 1 Max value = 0xFFFF (65'535) Default value = 0 (keine Verbindungsprojektierung möglich)

## 6 Weitere Kommunikationsdienste

### 6.1 PG/OP Kommunikation

Die PG/OP-Kommunikation dient zum Laden von Programmen und Konfigurationsdaten, zum Durchführen von Test- und Diagnosefunktionen sowie zum Bedienen und Beobachten einer Anlage über OPs. Sie stellt Funktionen zur Verfügung, die in ethernetfähigen Saia PCD-Geräten bereits integriert sind. Zu unterscheiden sind die beiden Funktionsarten:

- Der PG-Betrieb mit STEP<sup>®</sup>7 an Industrial Ethernet ermöglicht es, einen grossen Teil des Funktionsumfangs von STEP<sup>®</sup>7 über Industrial Ethernet zu nutzen und die PCD Steuerungen über Industrial Ethernet zu programmieren, diagnostizieren, zu bedienen und zu beobachten.
- Der OP-Betrieb über Industrial Ethernet erlaubt das Bedienen und Beobachten der SBC-Steuerungen über Bedien- und Beobachtungsgeräte.

Für die PG/OP-Kommunikation über Industrial Ethernet ist eine explizite Programmierung nicht nötig.

6

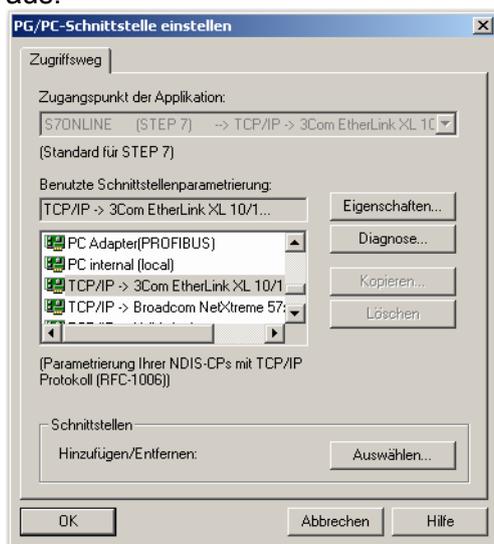


Der PG-Betrieb ist nur mit den PCD Steuerungen mit integriertem Ethernet Stack sowie mit der PCD2.M487 möglich

#### 6.1.1 Fernprogrammierung über Industrial Ethernet

Für die Fernprogrammierung über ISO-on-TCP Industrial Ethernet sind folgende Voraussetzungen zu beachten:

- 1) Installieren Sie nun das TCP/IP-Protokoll bei den Netzwerkeinstellungen Ihres Rechners, wenn das Protokoll noch nicht installiert ist.
- 2) Öffnen Sie die Systemsteuerung über Start > Einstellungen > Systemsteuerung und führen Sie einen Doppelklick auf das Icon PG/PC Schnittstelle einstellen aus.



- 3) Stellen Sie den Zugangspunkt S7ONLINE(STEP<sup>®</sup>7) auf TCP/IP > Netzwerkkarte ein (Bild).

Nun können Sie anstelle von MPI Ihre SPS über Industrial Ethernet (ISO-on-TCP) fernprogrammieren, bedienen und beobachten.

## 6.2 S7 Kommunikation (Server)

Die S7-Kommunikation ermöglicht den einfachen Datenaustausch zwischen PCD-Stationen und SIMATIC-S7-Stationen sowie zum PC. Bei Ethernet setzt die S7-Kommunikation auf das Schicht-4-Protokoll ISO-on-TCP auf. Hinter der Bezeichnung S7-Kommunikation verbergen sich mehrere Funktionen, die über verschiedene Systemfunktionsbausteine (SFB/SFC) auf der SIMATIC-Steuerung aufgerufen werden können. Diese stellen die Anwenderschnittstelle zur S7-Kommunikation dar. Die folgenden Eigenschaften der Funktionen bestimmen die Eigenschaften der Kommunikation:

- Programmgesteuertes Schreiben und Lesen von Variablen in einem SIMATIC-S7-Gerät ohne zusätzliche Kommunikationsfunktion im Anwenderprogramm der Saia PCD Steuerung. Diese Funktionen werden auf der Server-Seite im Betriebssystem ausgeführt. Es erfolgt keine Benachrichtigung des Partners wenn Daten übertragen wurden. Verwendete Systemfunktionen im SIMATIC-S7-Gerät: PUT/GET.

6

## 6.3 Dateiverwaltung und Dateizugriff über FTP

Der Einzug von Ethernet und TCP/IP in die Produktion ermöglicht den direkten Datenaustausch über alle Automatisierungsebenen hinweg. Webservices auf Basis XML, Anbindungen über JAVA mit HTTP, bis zu Kopplungen von Steuerungen über OPC und Ethernet stellen die komfortabelsten Arten der Datenkopplung dar. Doch oft stehen Automatisierer jetzt vor dem Problem, dass trotz Ethernet und TCP/IP keine direkte Kommunikation zwischen einer SPS und anderen beteiligten Komponenten oder Rechnern verschiedenster Hersteller möglich ist. Zum Beispiel ist die weit verbreitete OPC Datenschnittstelle meist nur auf gängigen PC-Betriebssystemen anzutreffen. Ein Grossrechner unter UNIX oder ein kleines Robotersystem mit einem embedded Betriebssystem sind in der Regel von der direkten Ethernet-Kommunikation mit der SPS ausgeschlossen.

Der Wunsch, mit einer SPS über Ethernet und TCP/IP mit allen angeschlossenen Kommunikationspartnern direkt zu kommunizieren, ist daher klar nachzuvollziehen. FTP (File Transfer Protocol) ist eines der wichtigsten und meistgenutzten Protokolle im Internet. Hauptverwendungszweck von FTP ist der programmgesteuerte Datenaustausch zwischen kleinen und grossen Rechnerplattformen auf verschiedenen Rechnerarchitekturen. Daher bietet sich FTP als verfügbare Lösung für diese direkte Kommunikation an.

Innerhalb der Saia PCD3-Systems wird der FTP-Zugriff Standardmässig mit einer Reihe von FTP-Kommandos realisiert. Mit den File-Transfer-Funktionen können Dateien zu und von Ihrer PCD3-Station übermittelt werden. Die Übermittlung ist vom PG/PC zur PCD-Station möglich.

### 6.3.1 FTP im Produktionsumfeld

Das FTP, verbunden mit dem Dateisystem der SPS, bietet in Produktionsstätten über Ethernet die direkte und sichere Kopplungsmöglichkeit zwischen Geräten der Feldebene und unterschiedlichsten Rechnersystemen der oberen Leitebene. Daraus ergeben sich eine Reihe von Vorteilen:

- Höhere Verfügbarkeit und geringere Hardwarekosten durch Einsparung von Gateway-Funktionen durch direkte Kommunikation über Ethernet mit Rechnern

- verschiedenster Betriebssystemplattformen.
- Einfaches Versionshandling durch Ablage verschiedener aktueller Projektdaten über FTP im Dateisystem der SPS (HTML-Dokumentation, Programme, Visualisierung).
- Einfache, leicht zu wartende und flache Kommunikationsstrukturen sogar über Firmennetzwerkgrenzen hinaus (WAN-Zugriff).
- Direkter Zugriff auf aktuelle SPS-Daten im Dateiformat erhöht die Transparenz von Produktionsprozessen.
- Flexible Produktionssteuerung durch direkte Dateikopplung (Rezeptverwaltung, aktuelle Statistiken).

### 6.3.2 Kommunikation einer SPS über FTP

PCD Systeme mit integriertem Ethernet Stack unterstützen den FTP-Serverbetrieb.

6

Soll von einem externen Rechner auf Daten wie z. B. Statistiken der SPS zugegriffen werden, so kann dies im Modus „SPS als FTP-Server“ erfolgen. Dabei gibt es die Möglichkeit direkt auf den Speicher der SPS (Datenbausteine) oder auf das Flash-Filesystem der Flash Karten PCD3/7.R550M04 oder PCD3.R600.

Folgende Funktionen stehen dem Anwender zur Verfügung:

Von einem FTP-Client aus, beispielsweise PG/PC, können Sie auf die Dateien im WEB-Dateisystem der PCD zugreifen; dort liegen vorwiegend die zur Darstellung im Web-Browser vorgesehenen HTML-Seiten.

Von einem FTP-Client aus, beispielsweise PG/PC, können Sie auf Datenbausteine in einer CPU der PCD-Station zugreifen.

Das Anwenderprogramm in der CPU kann in das Flash Filesystem Datenfiles anlegen und schreibend oder lesend bearbeiten. Diese Files können Sie vom FTP-Client aus für den Transfer von Daten ansprechen.

Mit den zusätzlichen steckbaren Flash-Modulen hat die SPS ausreichend Massenspeicher, die neben Statistik-Daten auch Rezepturen, Dokumentationen oder Anleitungen zur Fehlerbehebung im HTML-Format beinhalten kann. Der Zugriff auf die Dateien kann mit Standard-Tools wie Browsern oder FTP-Client-Software erfolgen.

## 6.4 HTTP-Direkt / Webserver

Das World Wide Web, oder einfach WWW, ist der flexibelste Dienst für das Internet, da er aufgrund seines Aufbaus je nach Bedarf auch andere Internet-Dienste beinhalten kann.

Das WWW ist ein System, lokal aufbereitete Informationen global abrufbar zu machen. Seine konzeptionelle Grundlage ist das so genannte Hypertextsystem, das es ermöglicht, verschiedene Informationen wie Webseiten, Dateien und Rechner weltweit miteinander zu verbinden und zu verknüpfen. Solche Verknüpfungen bezeichnet man als Hyperlinks. Vergleichen kann man die Hyperlinktechnik mit einem Lexikonzugriff: Sie schlagen in einem Lexikon einen Begriff nach und finden dort neben dem gesuchten Begriff auch Querverweise auf verwandte Begriffe.

Im Hypertextsystem des WWW enthalten die angezeigten Inhalte ebenfalls Querverweise zu anderen Dokumenten oder Dateien im Internet. Diese Hyperlinks sind in der Regel vom Anzeigeprogramm für dieses System (Browser) in besonderer Weise aus dem übrigen Text hervorgehoben. Wird ein Hyperlink per Mausklick aktiviert, verzweigt der Browser automatisch zum Ziel des gewählten Links. Mit dieser Technik ist es möglich, sich durch einfachen Mausklick von einer Webseite zu irgendeiner anderen Webseite auf der Welt zu bewegen. Das WWW erlaubt auch die Einbindung von Bildern, Sounds oder Animationen in die Hypertext-Dokumente.

6

Die Grundlage für das WWW bilden prinzipiell drei Dinge:

- 1) Die Adressierung über URL (Uniform Resource Locator) ist eine leicht zu bedienende Schreibweise für den Anwender. In einer URL wird das zu verwendende Protokoll und der Speicherort der Information angegeben.
- 2) Die Sprache HTML (Hypertext Markup Language) ist eine Seitenbeschreibungssprache, die über Formatierungsbefehle (Tags) vorgibt, wie Informationen von einem Browser darzustellen sind.
- 3) Das Protokoll HTTP (Hypertext Transfer Protocol) regelt die Anforderung und Übertragung von Informationen (http-Seiten) zwischen dem HTTP-Server und dem Browser. Sobald ein Teilnehmer mit Browser über URL eine HTML-Seite aufruft, wird diese Information über das HTTP-Protokoll im Internet übertragen und ebenso an den entsprechenden Server geleitet. Dieser beantwortet die Teilnehmeranfrage und sendet - ebenfalls wieder über HTTP-Protokoll - die gewünschte HTML-Seite über Internet an den anfragenden Teilnehmer.

Eine WWW-Bedienung läuft im Allgemeinen wie folgt ab: Der Teilnehmer-PC mit Browser ruft über die eingegebene URL Adresse eine HTML-Seite auf. Diese Anfrage wird mit dem HTTP-Protokoll (eingepackt in das TCP/IP-Protokoll) über das Internet geleitet, so dass der gewünschte Ziel-Server diese Anfrage beantwortet und die gewünschte HTML-Seite dem Anfragenden Teilnehmer wieder über HTTP zurücksendet.

### Der Browser

Die Steuerung, Adressierung und Darstellung der Multimediainformationen erfolgen über speziell dazu entwickelten Programmen, so genannte Browser. Die Entwicklung der Browser ist in den letzten Jahren rasant vorangeschritten. Mittels so genannter Plug-Ins können die multimedialen Fähigkeiten von Web-Browsern um neue Standards erweitert und der technologischen Entwicklung ständig angepasst werden.

Die beliebtesten Browser mit grafischer Benutzeroberfläche sind der Microsoft Internet Explorer gefolgt vom Netscape Communicator und Opera. Bei vielen Internet-Anbietern liegt der Browser der Zugangssoftware bei. Die Bedienung des Browsers ist einfach: Um eine Seite im Internet zu erreichen, tippt man im Feld Adresse die URL der gesuchten HTML-Seite ein und bestätigt mit der Eingabetaste. Eine detaillierte Information zur Bedienung Ihres Browsers und dessen Konfiguration erhalten Sie normalerweise durch die im Browser integrierte Online-Hilfe.

Die besten Resultate mit Web Projekten auf den PCD3 erzielt man beim Einsatz von MS Internet Explorer 6.0, der MS Java Virtual Machine oder der Sun Java Virtual Machine 5 (speziell für S-Web Editor Projekte).

In der folgenden Tabelle sind weitere Browser aufgeführt, die für Web Projekte mit der PCD3 eingesetzt werden können:

	<sup>*)</sup> Permanent connections	S-Web Projects (Applet)	Html
MS Internet Explorer 6.0			
MS Java Virtual Machine 6.0			
Netscape Browser 7.2			
Sun Java Virtual Machine 1.4	X		
Sun Java Virtual Machine 5			

x: wird nicht unterstützt.

<sup>\*)</sup> Beim Einsatz eines Browsers der die «Permanent connections» nicht unterstützt, werden die dargestellten PLC-Medien im Web-Projekt nicht regelmässig aktualisiert.

**URL - Adressierung im Internet**

Erreichen kann man WWW-Dokumente über ein standardisiertes Schema für die Adressierung. Dazu hat sich das URL-System durchgesetzt. URL wird im Netz verwendet um Informationen vollständig zu bezeichnen. Mit einer URL werden nicht nur eine Datei und das zugehörige Verzeichnis, sondern auch der Rechner festgehalten, auf dem die Datei zu finden ist. Da es im Internet verschiedene Dienste (z. B. FTP, E-Mail, WWW) und somit verschiedene Protokolle gibt, wird schliesslich noch die Zugriffsmethode festgehalten. Die allgemeine Schreibweise einer URL lautet:

Protokoll://Dienst.Rechneradresse:Port/Dateipfad/Dateiname

Eine URL besteht also aus mehreren Teilen, wobei nicht immer alle Teile aufgeführt werden müssen. Da beim TCP/IP jedem Dienst eine Portnummer zur Unterscheidung zugewiesen wird, kann bei den allgemein festgelegten Ports (well known ports), wie zum Beispiel Port 80 für das HTTP-Protokoll, auf die Portangabe verzichtet werden.

Beispiel für ein URL: http://179.168.10.12/Start.htm

http:// Übertragungsprotokoll: Hyper Text Transfer Protocol  
 192.168.10.12 IP-Adresse der PCD3 Steuerung  
 /start.htm Name des Dokuments „start“, dieses liegt im HTML- Format (.htm) vor.

Weitere default Seiten auf den PCD3 Steuerungen: passwd.htm und status.htm.

## HTTP - Das Netzwerkprotokoll

HTTP (Hyper Text Transfer Protocol) ist das dem WWW zugrunde liegende Netzwerkprotokoll. Es bildet die technische Grundlage des World Wide Web.

Die Aufgabe von HTTP ist die Übertragung von HTML-Dokumenten, Bildern und anderen Dateien, welche allgemein auch als Ressourcen bezeichnet werden. Die Übertragung findet zwischen Web-Client (Browser) und Web-Server statt.

Die wesentliche Schwäche von HTTP ist die mangelnde Abhörsicherheit. Alle Informationen werden im Klartext übertragen. Inzwischen wurden Verbesserungen entwickelt, bei denen entweder eine zusätzliche Sicherheitsschicht zwischen TCP/IP und den höheren Internet-Diensten aufgebaut wird, oder HTTP mittels Verschlüsselungstechniken nur codiert übertragen wird.

Zur Ausführung dynamischer Anzeigen, bestimmter Aktionen oder Programme im Internet bedarf es weiterer Techniken. Grundsätzlich wird zwischen Programmen, die auf einem entfernten Server laufen, und Programmen, die im lokalen Rechner laufen, unterschieden. Für die Automatisierungstechnik haben sich Techniken wie Java, Java Script und Active-X durchgesetzt, welche alle auf dem lokalen Rechner ausgeführt werden. Mit den SBC PCD3 Steuerungen können auch die entfernten Servertechniken wie CGI aufgeführt werden.

CGI (Common Gateway Interface) ist eine Scriptsprache, welche auf Servern läuft. Klassische Anwendungen für CGI sind Überprüfung und Versand von Formularen, Besucherzähler für Webseiten, Gästebücher und Diskussionsbereiche sowie Datenbankzugriffe.

## Verfügbare Speicherbereiche für die Webapplikationen

Auf der PCD3 steht ein Flash-File System zur Verfügung. Für die Speicherung der Webapplikationen stehen auf der PCD3 neu 2 Möglichkeiten zur Verfügung.

- 1) Die verschiedenen Files des Web-Projektes werden mit Hilfe des Web-Builder Tools von SBC in SPS Datenblocks konvertiert und mit Hilfe des Programmier-tools STEP®7 in die Steuerung geladen.



Dieser Weg ist ausführlich im Dokument 26/775 Handbuch Web-Server xx7 beschrieben.

- 2) Über FTP: Auf den PCD Systemen mit integriertem Ethernet Stack können die Files des Web Projektes direkt in Flash- oder SD-Card Devices kopiert werden.

Dazu gehen Sie wie folgt vor:

- Öffnen Sie eine FTP-Verbindung zu Ihrer PCD. (Siehe auch Kapitel 6.3)
- Erstellen Sie auf dem Flash-Device das Verzeichnis : /WEBPAGES
- Kopieren Sie das Web Projekt in dieses Verzeichnis.

Folgende Flash-Devices stehen auf der PCD3 und PCD2.M5xx7 zur Verfügung:

Produkt Name	Device Grösse	Steckplatz	Device-Name
PCD7.R550M04	4MByte	Slot M1 Slot M2	M1_Flash M2_Flash
PCD3.R550M04 PCD3.R6000	4MByte 128MByte - 1GByte	I/O Slot 0 I/O Slot 1 I/O Slot 2 I/O Slot 3	SL0Flash SL1Flash SL2Flash SL3Flash



Weitere Informationen über die Projektierung und Konfigurierung von Web-Projekten mit Saia PCD-Steuerungen finden Sie im Dokument 26/775 Handbuch Web-Server xx7 beschrieben.

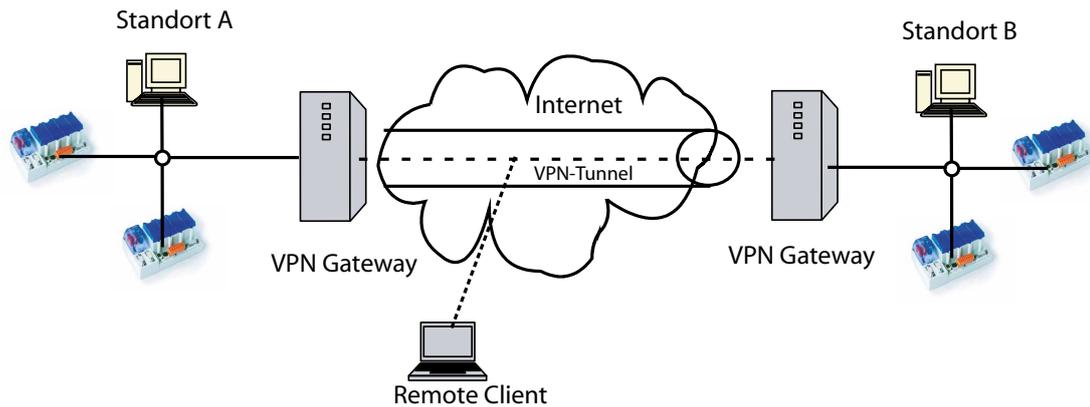
## 6.5 Virtuelle Private Netzwerke (VPN)

6

Entfernte Standorte werden immer seltener über reine Standleitungen miteinander vernetzt. Hier wird zunehmend das Internet als alternativer Übertragungsweg genutzt. Das Internet ist ein offenes Netzwerk, was es eigentlich für die Übertragung von vertraulichen internen Unternehmensdaten ungeeignet erscheinen lässt. Datenpakete passieren bei ihrer Reise durch das Internet eine Vielzahl von Punkten wie z. B. Server und Router bei Providern und können dort mit entsprechenden Tools ausgelesen werden. Unverschlüsselte Datenströme sind so in etwa vergleichbar mit dem Versand einer Nachricht auf einer Postkarte. Jedem ist das Lesen, Löschen oder Verändern der Nachricht möglich, der die Postkarte in seinen Besitz bringt.

Um nun die schnelle und preisgünstige Kommunikation über das öffentliche Internet mit der Sicherheit von privaten Standleitungen zu kombinieren, wurde die Technologie des VPN entwickelt. VPN steht für Virtuelles Privates Netzwerk - also ein privates Netz im öffentlichen Netz. Dies wird realisiert mit Hilfe von Chiffrier- und Authentifizierungstechniken. Vertrauliche Daten werden so über ein öffentliches Netz abhör- und manipulationssicher zwischen genau zwei Kommunikationspartnern ausgetauscht. Einstellbare Sicherheits-Parameter erlauben, dass nur solche Systeme miteinander kommunizieren können, die hierfür geeignet konfiguriert wurden. Auf diese Weise wird ein quasi-privates Netz innerhalb eines öffentlichen Netzes erzeugt, somit ein Netz im Netz, weil alle Fremdsysteme an dieser Kommunikation nicht teilnehmen können, so, als wenn sie technisch gar nicht angeschlossen wären.

Durch den Einsatz von VPN-Sicherheitstechnologien werden über das Internet authentifizierte und verschlüsselte Kommunikationskanäle bzw. so genannte Tunnel aufgebaut (siehe Bild). Die Informationen sind geschützt und ihre Übertragung wird überwacht. Generell handelt es sich bei einem VPN um ein geschlossenes, logisches Netz, das für eine bestimmte Benutzergruppe eingerichtet wird. Ein VPN nutzt immer die öffentlich zugänglichen Übertragungsnetzwerke, bei denen die Verbindungen durch einen öffentlichen Anbieter bereitgestellt werden. Der Anwender bildet sich über diese Übertragungswege praktisch sein privates Netz. Er verfügt über Sicherheitsmechanismen, wie die Identifikation und die Authentifikation der Netzteilnehmer, so dass sich Unbefugte keinen Zugang zum VPN verschaffen können.



Es ergibt sich aus der Nutzung der Internet-Infrastruktur ein dauerhafter und entscheidender Vorteil: Kosteneinsparung. Kostenintensive Datendirektverbindungen zwischen einzelnen Unternehmensstandorten können durch Internet-Anbindungen ersetzt werden. Wählverbindungen ins Internet sind bereits günstiger als Ortsgespräche. Erst recht bei Standleitungen ergibt sich ein immenser Kostenvorteil. Das Internet wird somit nicht nur zur allgemeinen Informationsbeschaffung und Kommunikation genutzt, sondern auch als Infrastruktur für die unternehmensinterne Vernetzung. Aufwändige zusätzliche Hardware für Direkteinwahlen oder mehrere Standleitungen entfallen ebenfalls. Diese eignen sich technisch hervorragend als universelle Infrastruktur zur Datenübertragung. Diese Netze stehen zwar dem Benutzer nicht exklusiv zur Verfügung, verursachen aber einen Bruchteil der Kosten von Standleitungen bei fast gleicher Verfügbarkeit mit allerdings meist nicht vorhersagbarer Bandbreite.

6

### VPN-Gateways

VPN-Gateways koppeln komplette lokale Netze über das Internet. In der Regel werden zwei oder mehrere Filialen über Standleitung an einen nahen Internet-Knoten angeschlossen und dann über ein VPN verbunden. Grundsätzlich wäre auch eine Wählverbindung möglich, ist aber meistens kostenmässig nicht attraktiv. VPN-Gateways können Stand-alone-Geräte sein, aber es werden auch viele Router, Firewalls und InternetServer mit VPN-Funktionalität geliefert. Stand-alone-Gateways bieten sich insbesondere bei hohen Verschlüsselungsdurchsätzen ( $> 2$  MBit/s) an, da diese Geräte mit spezieller Verschlüsselungshardware oft preiswerter und flexibler einzusetzen sind als universelle Systeme. Im Bereich von 2 oder 4 MBit/s arbeiten viele moderne Router und Firewalls auch ohne spezielle Verschlüsselungshardware komfortabel, schnell und preiswert.

### VPN-Remote-Clients

Ein Remote-Client ist ein einzelner Rechner (oftmals ein Notebook), der sich in das Internet einwählt und dann über einen VPN-Tunnel die Verbindung in das zentrale Netzwerk (mit den PCDs) herstellt. Hierzu wird auf dem System ein VPN-Client benötigt, der die Verschlüsselung durchführt. Microsoft liefert mit dem Windows-Betriebssystem direkt einen PPTP-VPN-Client.

## 6.6 SBC S-Bus-Kommunikation

Die xx7 Steuerungen unterstützen das reduzierte S-Bus Protokoll. Die xx7 PCD arbeitet immer im Servermode. Damit ein Client über S-Bus Daten lesen kann, muss die S-Bus Stationsnummer und die MPI- Adresse identisch gewählt werden. Nachfolgend werden drei mögliche Anwendungen für das Ether-S-Bus Protokoll aufgezeigt.

### Web Connect

Mithilfe des Web Connect ist es möglich, auf Daten der xx7 PCD zuzugreifen wie auf einer Classic PCD. Zu beachten gilt, dass die S-Bus Stationsnummer der MPI Adresse der xx7 PCD entspricht.

Damit ist es für ein WinCE Panel möglich, die xx7 PCD über Ethernet anzusprechen.

### PC Drivers / OPC Server

Die Integration des reduzierten S-Bus Protokolls erlaubt es, mithilfe von SBS SW Tools (OPC Server) auf Daten der xx7 zuzugreifen ohne dass dritte Produkte zugekauft werden müssen. Auch hier gelten die gleichen Bedingungen wie bei der Benutzung von Web-Connect.

### Datenaustausch zwischen xx7 und Classic

Das S-Bus Protokoll erlaubt den Datenaustausch zwischen xx7 und Classic PCD's. Die xx7 Steuerung arbeitet immer als Server, d.h. jeder Datenaustausch wird von der Classic PCD angestoßen. Die S-Bus Stationsnummer der xx7 Steuerung entspricht der MPI Adresse.

#### 6.6.1 Unterstützte Telegramme

Folgende Telegramme werden von der xx7 PCD unterstützt:

Telegram	Number
RdFlag	2
RdInput	3
RdRTC	4
RdOutput	5
RdRegister	6
WrFlag	11
WrRTC	12
WrOutput	13
WrRegister	14
RdStatus	20
RdFWVersion	32
RdDB	150
WrDB	151
RdBlocSize	156

Da Counter und Timer unterschiedliche Gültigkeitsbereiche besitzen, ist es nicht möglich, diese Datentypen zu lesen.

## 6.6.2 Media Mapping

Die S-Bus Telegramme unterstützen nur den Zugriff auf Classic PCD Medien. Da die Adressierung der Medien auf xx7 PCD's unterschiedlich zur Classic PCD erfolgt, muss die xx7 PCD eine Adressmapping durchführen.

### Flags

Die Classic Flags werden nach folgendem Schema in xx7 Flags transferiert:

Classic Flag	xx7 Merker
F 0	M 0.0
F 1	M 0.1
...	...
F 7	M 0.7
F 8	M 1.0
F 9	M 1.1
F 16	M 2.0
F 80	M 10.0

6

### Register

Die Classic Register werden nach folgendem Schema in xx7 Merker Doppelworte transferiert:

Classic Register	xx7 Merker Doppelworte
R 0	MD 0
R 1	MD 4
R 2	MD 8
...	...
R 500	MD 2000

### Inputs

Die Classic Inputs werden nach folgendem Schema in xx7 Inputs transferiert:

Classic Input	xx7 Inputs
I 0	I 0.0
I 1	I 0.1
...	...
I 7	I 0.7
I 8	I 1.0
I 9	I 1.1
I 16	I 2.0
I 80	I 10.0

## Outputs

Die Classic Outputs werden nach folgendem Schema in xx7 Outputs transferiert:

Classic Output	xx7 Output
O 0	O 0.0
O 1	O 0.1
...	...
O 7	O 0.7
O 8	O 1.0
O 9	O 1.1
O 16	O 2.0
O 80	O 10.0

## Data Blocks

Die Classic Data Blocks werden nach folgendem Schema in xx7 DB Doppelworte transferiert:

Classic DB Element	xx7 DB Doppelworte
DB-Element 0	DBD 0
DB-Element 1	DBD 4
DB-Element 2	DBD 8
...	...
DB-Element 500	DBD 2000

## 7 Email

### 7.1 Einleitung

Der beliebteste aller Internet-Dienste ist nach wie vor Electronic Mail, kurz: Email. Mit Email lassen sich Texte elektronisch versenden und empfangen. Je nach der verwendeten Email-Software, auch Email-Client genannt, können ebenfalls binäre Dateien oder Text-Dokumente als Anhang (Attachment) in unveränderter Form leicht an einen oder mehrere Adressaten versandt werden.

Damit nicht Sender und Empfänger immer beim Nachrichtenaustausch gleichzeitig im Internet oder die Endgeräte immer eingeschaltet sein müssen, um eine Übertragungsverbindung zwischen den beiden Teilnehmern aufzubauen, wurde ein Zwischenspeicher für die übermittelten Nachrichten eingeführt. Dieser Speicher ist realisiert auf einem Mail-Server und wird umgangssprachlich Mailbox (Postfach) genannt. Hier werden die eingehenden Nachrichten zunächst abgelegt und zwischengespeichert. Der Besitzer der Mailbox kann wie bei einem echten Briefkasten nach Belieben die Mailbox öffnen und die Nachrichten abrufen und lesen.

Diese Postfächer haben wie richtige Briefkästen auch Namen, nach denen der Verteildienst die Nachrichten dem jeweils bestimmten Empfänger zustellt. Dazu sind eindeutige Adressen, die so genannten Email Adressen, im Internet notwendig. Der Aufbau der Email Adressen ist international standardisiert. Ein Beispiel für eine Email Adresse ist in nachfolgendem Bild dargestellt:

Hans.mustermann@musterfirma.de

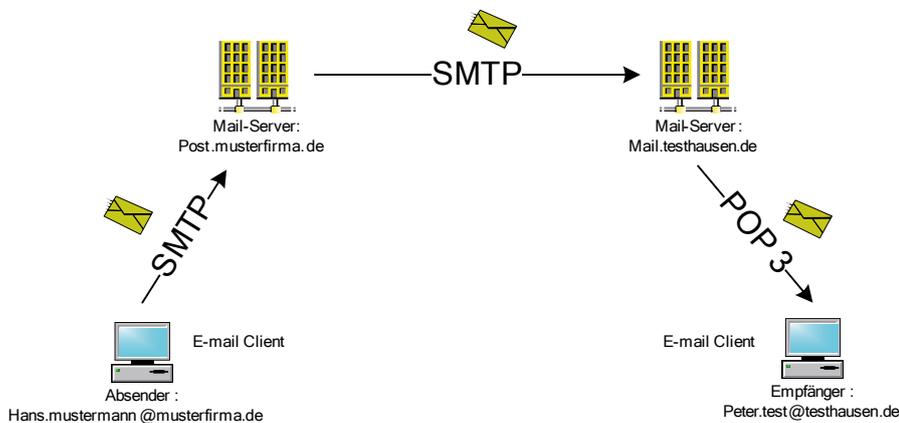
Benutzername      Rechnername

Die Adressen setzen sich immer aus dem Benutzernamen und dem Rechnernamen zusammen. In der Mitte der Adresse steht der so genannte Klammeraffe @, das at-Zeichen, wobei at im Sinne der englischen Wörter „bei“ oder „an“ zu verstehen ist. Es trennt den Namen einer Person (hans.mustermann) von der Bezeichnung des Email Servers, auf dem die digitale Post gesammelt wird (Rechner bei der musterfirma.de). Der Benutzername wird vom jeweiligen Internetprovider zugeteilt, der das Postfach zur Verfügung stellt. Der Benutzername darf keine Leerstelle enthalten, wohl aber etwa einen Punkt, einen Bindestrich oder einen Unterstrich. Er muss nicht mit dem persönlichen Namen des Benutzers identisch sein. Der zweite Teil der Adresse, der Rechnername des Email Servers, wird vom entsprechenden Internetprovider bekannt gegeben und gliedert sich in mehrere Teile. Der erste Teil, die Top-Level-Domain (in unserem Beispiel „de“ für die Länderkennung für Deutschland), wird immer durch einen Punkt abgetrennt. Die nachfolgende Second-Level-Domain „musterfirma“ gibt an, dass der Rechner Bestandteil des Rechnernetzes der Musterfirma ist.

Eine Email setzt sich aus dem Nachrichtenkopf und der eigentlichen Nachricht zusammen. Der Nachrichtenkopf besteht aus der Adresse des Empfängers (Feld: To oder An) und der Adresse des Absenders (Feld: From oder Von). Weiterhin kann zusätzlich ein Thema (Feld: Subject oder Betreff) eingegeben werden. Es ist möglich ein Massenschreiben zu entwickeln, indem man unter CC (carbon copy) weitere Email-Adressen für Kopien der Nachricht eintragen kann. CC-Empfänger werden allen beteiligten Empfängern angezeigt.

Der Absender verfasst mit seinem Email-Programm (Client) eine Nachricht. Der Weg

der Email vom Absender zum Empfänger besteht aus zwei Teilabschnitten, wobei der eigentliche Transport über zwei unterschiedliche Protokolle geregelt wird. Zum Übertragen der Nachricht vom Rechner des Absenders über den Email Server des Absenderproviders bis zum Postfach beim Provider des Empfängers wird das SMTP-Protokoll (Simple Mail Transfer Protocol) benutzt. Die Email wird erst zum Postausgangsserver des Absenders übertragen (siehe Server `post.musterfirma.de` im Bild). Von dort aus wird die Email über das Internet geroutet bis sie schliesslich am Server des gewünschten Postfachs ankommt. Dort wird die Email im Posteingangsserver (`mail.testhausen.de`) als Datei abgelegt, bzw. wenn bereits Mails in diesem Postfach liegen, an die vorhandene Datei angehängt. Nach der Zwischenspeicherung im Postfach des Empfängers muss die Email nur noch auf den Client des Mailempfängers transportiert werden. Dazu dient meist das POP3-Protokoll (Post Office Protocol Version 3). Der Server, der die Postfächer unterhält, nimmt über Port 110 Verbindungen entgegen und stellt die Emails eines bestimmten Postfachs nach Anmeldung mit Postfachname und Passwort zur Verfügung. Bei POP3-Verbindungen müssen alle empfangenen Emails komplett heruntergeladen werden um sie dann zu lesen.



Für eine wesentlich komfortablere Verwaltung wurde IMAP (Internet Message Access Protocol) entwickelt. Damit können die Emails vollständig auf dem Server bleiben, der Client lädt sich jeweils nur die Informationen herunter, die er für die aktuelle Anzeige benötigt. So ist es möglich, sich beispielsweise nur die Kopfzeilen, nicht aber die ganze Email mit allen Anhängen anzuzeigen, um so Übertragungszeit einzusparen. Der größte Vorteil von IMAP ist, dass das Postfach von verschiedenen Clients abgefragt werden kann, da die Email auf dem Server verbleibt und nur eine Kopie zur Ansicht angefertigt wird. Bei POP3 hingegen, ist eine Email, die von einem Client am Server abgeholt wurde, auf dem Server gelöscht. Ein anderer Client wird diese Mail nicht mehr lesen können. Sollen binäre Daten über Email verschickt werden, so wird der MIME-Standard (Multipurpose Internet Mail Extension) automatisch vom eingesetzten Email-Programm verwendet.

### Prozessmeldungen über E-Mail versenden

Bei Ereignissen in einer Saia PCD-Steuerung können automatisch Informationen durch Email weitergegeben werden. Mit der Email-Funktion der PCD kann das Automatisierungssystem prozess- oder zeitabhängig Nachrichten mit Prozessinformationen versenden. So ist beispielsweise eine direkte Benachrichtigung des Servicepersonals mit ersten Fehlerinformationen des Prozesses über Email möglich.

Gemäss den üblichen Merkmalen von Emails können Nachrichten mit oder ohne Anhang über die PCD versandt werden. Bei Anhängen können Informationen aus einer

Steuerung zur Auswertung an den Empfänger übermittelt werden.

Die PCD übernimmt beim Mailversand folgende Funktionen:

- Als Email-Client unterstützt die PCD den SMTP-Dienst
- Emails können vom Automatisierungssystem gesendet, nicht jedoch empfangen werden.

### **Vorgehensweise zum Versand von E-Mails**

Um mit der PCD ereignisgesteuert Emails versenden zu können, müssen Sie folgende Vorbereitungen treffen:

- 1) Einen Email-Server bereitstellen
- 2) Die Email vom Anwenderprogramm aus mittels eines SFB-Bausteins `SendMail` versenden. Die mit der Email zu versendenden Informationen einschliesslich der Adressinformationen werden in einem Datenbaustein (DB) hinterlegt.

Die PCD schickt als Email-Client über das SMTP seine Meldung an einen Email-Server. Dieser speichert die Nachricht und muss nun entscheiden, ob der Empfänger und damit die Empfängerdomain bei ihm verwaltet wird oder nicht (d.h. klären, ob der Empfänger bei ihm ein Postfach hat). Ist dies nicht der Fall, routet der Server die Meldung nach seiner Routing-Tabelle an den nächsten Email-Server in der nächsthöheren Domänebene.

7

## **7.2 System Funktion SFB SendMail**

Damit eine Email versendet werden kann, genügt es den SFB `SendMail` aufzurufen. Bei einer positiven Flanke am Parameter `Act`, wird der Email Send Prozess angestoßen. Über das `Done` oder `Error` Flag sowie den Status wird der Anwender informiert, ob die Email erfolgreich abgesetzt werden konnte.

Calling the SendMail Function:

```
CALL „SendMail“, „I_DB_SendMail“
  Act           :=TRUE
  MTAConfig     :="MailConfig".MTAConfig
  Authentication :="MailConfig".authentication
  Priority       := 1
  MailContent   :=P#DB3.DBX0.0 BYTE 200
  Done          :=M10.0
  Error         :=M10.1
  Status        :=MD12
```

Status Werte

Status Word Values	
0	No error
0x8080	Busy Email sending active
0x8181	No Email sending active
-1	Unknown, general error
-2	Error when setting up the socket interface
-3	Error when connecting the server
-4	Error when closing the socket interface
-5	Reception/sending on the socket interface time out
-6	Error sending data to the interface
-7	Error receiving data on the interface
-20	Error on sending the request into the mail queue
-21	To many asynchronous requests waiting, queue full
-22	MTA rejected "QUIT" command
-23	MTA rejected "DATA" command
-24	MTA rejected "RCTP TO" command
-25	MTA rejected "MAIL FORM" command
-26	MTA rejected "HELO" command
-27	MTA return code unknown
-28	MTA rejected relaying mail
-29	MTA rejected "AUTH" command
-100	Interpreted parameter was to long
-101	Interpreted MTA address was to long
-102	Interpreted sender addresses to long
-103	Interpreted reception address to long
-104	Interpreted subject to long
-105	Interpreted user name to long
-106	Interpreted password to long
-107	Interpreted file path to long
-108	File open command failed (file already open ?)
-109	File to send exceeds 1 MB

7.3 Konfigurationen

7.3.1 MTA Konfiguration

Die PCD arbeitet als Email Client. Sie sendet die Email im SMTP Format an einen Email-Server (Message Transfer Agent (MTA)). In der MTAConfig wird die IP- Adresse sowie der Port und eine allfällige Timeout Zeit des Email-Servers konfiguriert.

MTA	STRUCT		vorläufige Platzhaltervariable
port	DINT	L#25	Port of the message transfer agent, default is port 25
Adresse	STRING[15]	'xxx.xxx.xxx.xxx'	IP adress of the message transfer agent
Timeout	DINT	L#0	Send and receive time out. If the time expires send mail returns with an error.
	END_STRUCT		

7.3.2 Authentication

In der authentication structure ist es möglich einen User Name sowie das zugehörige Passwort für den SMTP mail Server zu definieren. Falls diese Felder nicht ausgefüllt werden, wird die Authentifikation ausgeschaltet.

authentication	STRUCT		
UserName	STRING[64]	''	User name credentials for the SMTP mail server. With empty field no authentic
UserPassWord	STRING[64]	''	User password for the SMTP mail server. Not Entry -> no authentication
	END_STRUCT		

## 7.4 Mail Inhalt

Der Mail Content enthält mehrere Strings. Nicht alle Strings sind für das Senden der Email erforderlich.

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	TO	STRING[40]	'TO: ITUser@IT4Automation.de;'
+42.0	FROM	STRING[40]	'FROM: ITCP@IT4Automation.de;'
+84.0	SUB	STRING[40]	'SUB: Meldung der Demoanlage;'
+126.0	MIME	STRING[10]	'MIME: TEXT;'
+138.0	TXT	STRING[40]	'TXT: Dies ist eine Testmail !!!'
+180.0	ATTACHMENT	STRING[40]	'FILE: M1_FLASH: /myfile.pdf;'
=222.0		END_STRUCT	

Die ID's der Strings müssen korrekt geschrieben werden (gross/klein Schreibung).

ID	Description	Example
FROM:	Email address or account name of the sender (max 63 chars)	
TO:	Email addresses of the recipients separated by “,” (max 127 char.)	
SUB:	Subject of the mail (max 127 characters)	
MIME:	Type of the Mail Format: <ul style="list-style-type: none"> <li>• TEXT sends plain text mails (default)</li> <li>• HTML sends HTML mails. HTML formatting of the content is allowed</li> </ul>	
TXT:	Text content of the email.	
TXTF:	Filename of a file that contains the content of the Email (Plain Text or HTML Text).	
FILE:	Name of a file to be send as attachment.	

Die ID 'TXT:' kann in einem Email DB mehrfach benützt werden. Alle Texte werden zu einem Text im Email zusammenkopiert (ohne ID).

Die ID 'FILE:' kann mehrfach in der gleichen Email benützt werden. Jedes File wird als Attachment im Email angehängt und gesendet.

Jeder Text muss mit ';' abgeschlossen werden, Ausnahme ist TXT:

## 8 SBC Modbus

### 8.1 Einleitung

Modbus ist ein request/response Protokoll. Die xx7 PCD stellt Systemfunktionen zur Verfügung, um über Modbus Daten auszutauschen.

Die Saia PCD xx7 unterstützen das Modbus Protokoll auf der seriellen Schnittstelle sowie über TCP/IP (sowohl TCP als auch UDP). In diesem Dokument werden nur die Systembausteine beschrieben, die für die TCP/IP Kommunikation notwendig sind.

Für eine Modbus Kommunikation sind auf der Client Seite folgende Schritte notwendig:

- 1) Erzeugen einer Client Instanz
- 2) Aufbau einer Verbindung zu einem Server
- 3) Sende Auftrag
- 4) Auswerten der Antwort
- 5) Optional ist es möglich nach dem Empfangen der Antwort oder nach Ablauf einer Idle Zeit, die Verbindung zu schliessen,

8

Auf der Server Seite, sind für die Kommunikation folgende Schritte notwendig:

- 1) Erzeugen einer Server Instanz
- 2) Auswerten der empfangenen Telegramme
- 3) Senden einer Antwort

In den nachfolgenden Kapitel sind die einzelnen Schritte, sowie die Systemfunktionen für die Modbus Kommunikation näher beschrieben.

#### 8.1.1 SBC Modbus Server

Auf den SBC Modbus Server kann über TCP/UDP zugegriffen werden. Es können mehrerer Server Instanzen auf einer Saia PCD definiert werden, welche mit mehreren verschiedenen Clients über IP kommunizieren können. Eine Modbus Server Instanz wird über sogenannte Unit Identifiers (UID) adressiert. Einer UID ist ebenfalls ein spezifisches Media mapping sowie die Datenverarbeitung zugeordnet. Es ist möglich mehrerer unterschiedliche UID's auf einer PCD zu definieren.

Der Modbus Server verarbeitet alle Telegramme die an eine UID gesendet werden. Der Server sendet entweder einen Fehler Code oder eine positive Antwort zum Client zurück.

Durch die Möglichkeit, mehrere UID's auf einer PCD zu konfigurieren, ist es möglich unterschiedliche Media mappings mit unterschiedlicher Verarbeitung zu definieren.

#### 8.1.2 SBC Modbus Client

Der SBC Modbus Client ermöglicht es dem Anwender Lese- und Schreib Telegramme für das Modbus Protokoll zu Senden und Empfangen. Auf der PCD können mehrerer Kanäle definiert werden, um mit verschiedenen Modbus Servern zu kommunizieren.

Ein Modbus Telegramm wird immer an eine spezifische Adresse gesendet. Die Adresse besteht aus der UID sowie der IP Adresse. Es ist ebenfalls möglich die Daten

Typen zu definieren, die gesendet bzw. empfangen werden sollen.

Auf jedes gesendete Telegramm wird eine Antwort innerhalb einer spezifizierten Zeit erwartet. Falls in dieser Zeit keine Antwort empfangen wurde, geht der Client davon aus, dass der Server nicht existiert.

### 8.1.3 SBC Modbus Limitierungen

Für den SBC Modbus existieren folgende Limitierungen:

Server: Es können maximal 4 Modbus Server auf einer PCD gestartet werden. Z.B 1 Server TCP Port 502, 1 Server TCP Port 503, 1 Server UDP Port 502, 1 Server RTU auf der 2. seriellen Schnittstelle.

UID: Es können maximal 10 UID auf einer PCD definiert werden (10= 9 + UID 255). Die UID 0 bis 254 sind gültige Werte. Die UID 255 existiert immer als default UID. Sie muss aber vor der ersten Benützung konfiguriert werden.

Mapping Areas: pro UID können maximal 10 verschiedene Mappings definiert werden. Werden mehr als 10 verschiedenen Mappings benötigt, kann eine zusätzliche UID definiert werden.

Verbindungen: Auf einer PCD können maximal 26 Modbus Verbindungen gleichzeitig geöffnet werden. Von diesen 26 Verbindungen können maximal 10 Verbindungen vom Modbus Client benützt werden. Die verbleibenden Verbindungen können als Server Verbindungen benützt werden.

Eine Verbindung ist definiert durch ein eindeutiges Trippel (Port – Protokoll- IP Adresse). Die Verbindungen werden automatisch beim ersten Modbus request geöffnet, und nach einer Anwender definierten Zeit wieder geschlossen.

8

## 8.2 Modbus Medien und xx7 Medien

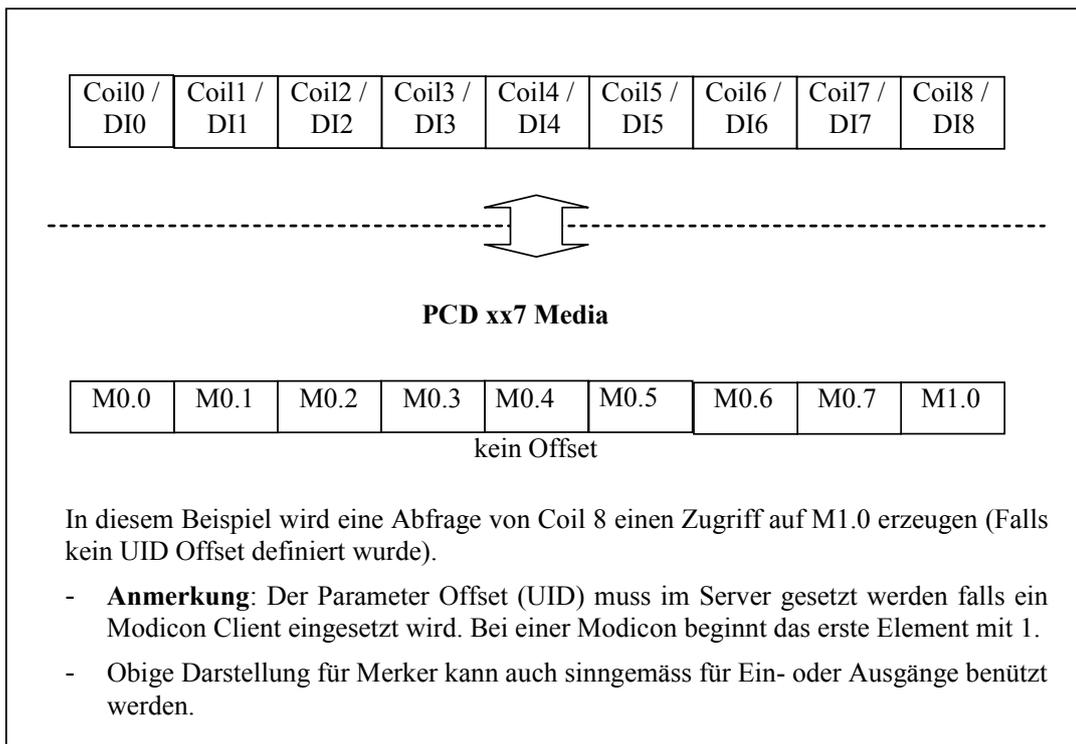
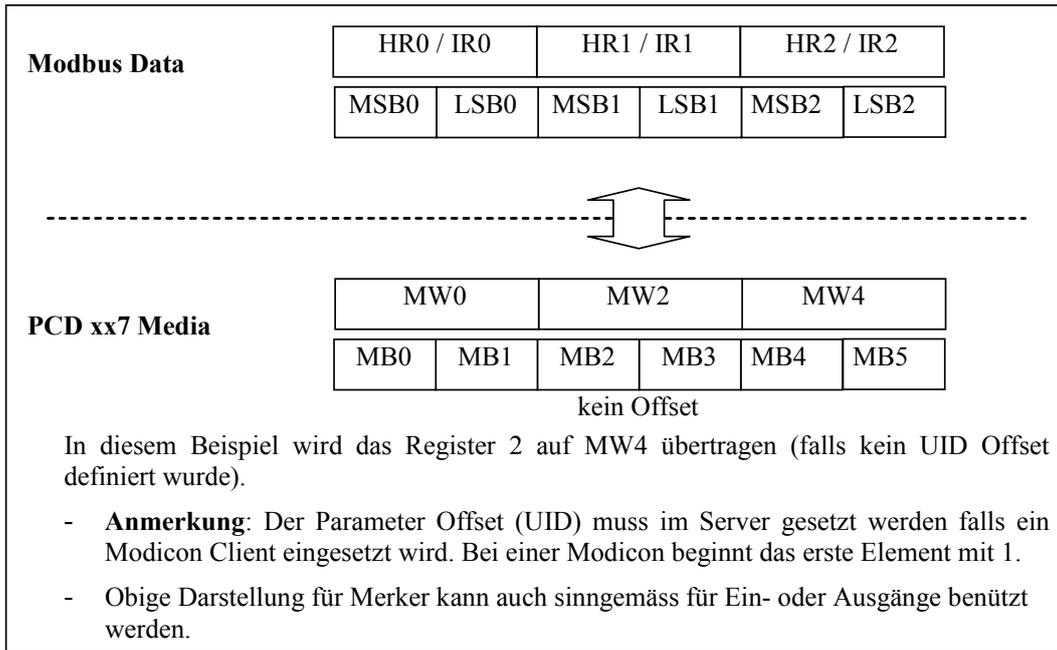
In Modbus sind vier Medientypen definiert: Coils, digitale Eingänge (Discrete Inputs), speichernde Register (Holding Registers), und Eingangs Register (Input Registers).

Damit der Modbus Server auf die PCD- Medien zugreifen kann, müssen die Modbus Medien auf die PCD Medien gemappt werden. Mit Hilfe der UIDs und der Mapping Bereiche kann der Anwender das Media mapping konfigurieren. Mindestens eine UID muss immer definiert werden.

Coils und digitale Eingänge können auf Merker, Eingänge oder Ausgänge gemappt werden. Die Eingänge der PCD können über Modbus nur gelesen, nicht aber beschrieben werden.

Speichernde Register und Eingangs- Register können auf Merker, Eingänge, Ausgänge, Zeiten und Zähler gemappt werden.

Die folgenden Schemata zeigt wie die Daten in einem Modbus Telegramm bearbeitet werden:



## Default Mapping

Access Type	Area Processing	Modbus Request			PCD		
		Media Type	Start Addr.	Range	Media Type	Start Addr.	Range
ReadWrite	NO_PROCESSING	MB_HOLDING_REG MEDIA	1	10000	PCD_XX7_MERKER	0	20000
ReadWrite	SWAP_WORDS	MB_HOLDING_REG MEDIA	10001	10000	PCD_XX7_MERKER	0	20000
ReadOnly	NO_PROCESSING	MB_INPUT_REG MEDIA	1	10000	PCD_XX7_COUNTER	0	10000
ReadOnly	NO_PROCESSING	MB_INPUT_REG MEDIA	10001	10000	PCD_XX7_TIMER	0	10000
ReadWrite	NO_PROCESSING	MB_COILS_MEDIA	1	10000	PCD_XX7_MERKER	0	10000
ReadWrite	NO_PROCESSING	MB_COILS_MEDIA	10001	10000	PCD_XX7_OUTPUT	0	10000
ReadOnly	NO_PROCESSING	MB_DISCR_INPUT MEDIA	1	10000	PCD_XX7_MERKER	0	10000
ReadOnly	NO_PROCESSING	MB_DISCR_INPUT MEDIA	10001	10000	PCD_XX7_INPUT	0	10000

8

### 8.3 System Funktionen für Modbus Client

#### 8.3.1 OpenIPChannel (FB490)

Der FB490 „OpenIPChannel“ erzeugt eine TCP/IP Verbindung für den Modbus Client. Ein Kanal (Port/Protokoll) darf nicht mehrfach verwendet werden.

Aufruf:

```
CALL FB 490 , DB490
ChannelID      :=
Port           :=
Protocol       :=
CloseTimeout   :=
RespTimeout    :=
Retries        :=
Status        :=
```

Parameter Beschreibung:

Name	Typ	Format	Beschreibung
Channel ID	IN	INT	Ein Modbus Client kann mehrere logische Kanäle besitzen. Die Kanal Id wird zum erzeugen, ändern des Kanals benutzt als auch zum senden von Daten zum Modbus Server. Erlaubter Bereich: 1...10
Port	IN	INT	IP- Port Nummer auf welcher der Modbus Server erreichbar ist.
Protocol	IN	INT	IP- Protokoll über welches der Modbus kommuniziert: 0 = Modbus_TCP 1 = Modbus_UDP
Close Timeout	IN	DINT	Zeit nach welcher eine Verbindung abgebaut wird, wenn keine Kommunikation stattfindet. (in s)
RespTimeout	IN	DINT	Zeit in welcher eine Antwort vom Server erwartet wird. (in ms)
Retries	IN	INT	Anzahl Sendewiederholungen im Falle von Response Timeout. Bereich: 1...15
Status	OUT	WORD	Rückgabewert des FB's. Bei ein Wert 0x8... wurde der FB fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.

8

### 8.3.2 SendReadReq (FB492)

Der FB 492 erlaubt das Lesen von Daten von einem Modbus Server.

Aufruf:

```
CALL FB 492 , DB492
Req           :=
ChannelID    :=
IPAddr       :=
UID          :=
FunctionCode :=
StartAddr    :=
Length       :=
Options      :=
NDR          :=
Error        :=
ExceptionCode :=
Status       :=
Data         :=
```

## Parameter Beschreibung:

Name	Typ	Format	Beschreibung								
Req	IN	BOOL	Eine positive Flanke auf dem Req bit löst einen Read Request aus.								
Channel ID	IN	INT	Ein Modbus Client kann mehrere logische Kanäle besitzen. Die Kanal ID wurde beim Erzeugen des Kanals (FB490) definiert. Erlaubter Bereich: 1...10								
IPAddr	IN	DWORD	IP Adresse des Modbus Servers. Es ist in einem Doppelwort (in Hex) Codiert. Jedes Byte entspricht einem Teil der IP-Adresse. Beispiel: Für die Adresse 172.23.2.156 wird der Wert DW#16#AC17029C angegeben. 0xAC = 172, 0x17 = 23, 0x02=2, 0x9C = 156.								
UID	IN	INT	Der Unit Identifier (UID) wird ins Modbus Telegramm eingefügt. Wenn der Server ein Telegramm empfängt, so wird über die UID definiert, wie er die Daten verarbeiten muss. Auf einem Server können mehrere UIDs definiert werden. Auf dem Server müssen die UID's definiert sein (FB InitUID). Die UID 0 und 255 sind identische UIDs. Es wird empfohlen die UID 0 nicht zu benutzen. Bereich 0...255.								
FunctionCode	IN	INT	Dieser Parameter definiert den Request Typ: <table style="margin-left: 20px; border: none;"> <tr> <td>READ_COILS</td> <td>0x01</td> </tr> <tr> <td>READ_DISC_INPUT</td> <td>0x02</td> </tr> <tr> <td>READ_HOLD_REG</td> <td>0x03</td> </tr> <tr> <td>READ_INPUT_REG</td> <td>0x04</td> </tr> </table>	READ_COILS	0x01	READ_DISC_INPUT	0x02	READ_HOLD_REG	0x03	READ_INPUT_REG	0x04
READ_COILS	0x01										
READ_DISC_INPUT	0x02										
READ_HOLD_REG	0x03										
READ_INPUT_REG	0x04										
StartAddr	IN	DINT	Modbus Adresse des ersten zu lesenden Elementes.								
Length	IN	INT	Anzahl zu lesender Modbus Elemente. Maximale Anzahl von Coils / DI: 2000 Maximale Anzahl von HR / IR: 125								
Options	IN	DWORD	Optionen für den Send Request: Bit 0: Swap Words. Die 2 Worte eines Doppelwortes werden vor dem Senden gewappt. Das Swapping ist nur auf Register anwendbar. Die Anzahl Register muss geradzahlig sein. Bit 1...31 reserved								
NDR	OUT	BOOL	Dieses Bit bleibt solange auf 0, wie der Read Request abgearbeitet wird, oder der Request nicht gestartet werden kann. Sobald der Request erfolgreich beendet wurde (eine korrekte Antwort wurde vom Server Empfangen) wird dieses Bit auf 1 gesetzt.								
Error	OUT	BOOL	Wird dieses Bit auf 1 gesetzt, so ist beim Ausführen des Requests ein Fehler aufgetreten. Entweder durch falsche Parameter oder ein Fehler wurde vom Server gesendet.								
Exception Code	OUT	BYTE	Im Exception Code steht der Fehler Code, welcher der Modbus Server zurückgesendet hat. Falls kein Fehler vom Server gesendet wurde, ist dieser Wert auf 0 gesetzt.								

Name	Typ	Format	Beschreibung
Status	OUT	WORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.
Data	IN / OUT	ANY	Der any pointer gibt an wohin die empfangenen Daten kopiert werden sollen. Er kann auf DB's, Merker, Eingangs- und Ausgangs- Bereiche zeigen. Der Typ Bool ist nur erlaubt beim Lesen von coils, oder digitalem Eingang Beispiele: p#DB10.DBX0.0 BYTE 20 Es werden 20 Byte in den DB10 ab Byte 0 kopiert. p#E150.1 BOOL 9 Es werden 9 bits in den Eingangsbereich kopiert. Das erste Bit wird auf E150.1 kopiert.



Wird ein Read Request angestossen, so darf sich der Parameter Channel ID nicht ändern, bis der Lesezugriff abgeschlossen wurde. (Error oder NDR Bit wurde gesetzt). Falls mehrere Kanäle parallel gelesen werden müssen, so muss für jeden SendReadRequest Aufruf mit einer unterschiedlichen ChannelID mit einem separaten Instanz DB aufgerufen werden.

Es kann kein neuer Request angestossen werden, solange der aktuelle Request noch nicht abgeschlossen ist (NDR oder ERROR NDR Bit wurde gesetzt).

8

### 8.3.3 SendWriteReq (FB493)

Der FB 493 erlaubt das Schreiben von Daten in einem Modbus Server.

Aufruf:

```
CALL FB 493 , DB493
Req           := [IN]      [BOOL]
ChannelID    := [IN]      [INT]
IPAddr       := [IN]      [DWORD]
UID          := [IN]      [INT]
FunctionCode := [IN]      [INT]
StartAddr    := [IN]      [DINT]
Length       := [IN]      [INT]
Options      := [IN]      [DWORD]
Done         := [OUT]     [BOOL]
Error        := [OUT]     [BOOL]
ExceptionCode := [OUT]    [BYTE]
Status       := [OUT]     [WORD]
Data        := [I/O]     [ANY]
```

## Parameter Beschreibung:

Name	Typ	Format	Beschreibung
Req	IN	BOOL	Eine positive Flanke auf dem Req bit löst einen Write Request aus.
Channel ID	IN	INT	Ein Modbus Client kann mehrere logische Kanäle besitzen. Die Kanal ID wurde beim Erzeugen des Kanals (FB490) definiert. Erlaubter Bereich: 1 ... 10
IPAddr	IN	DWORD	IP Adresse des Modbus Servers. Es ist in einem Doppelwort (in Hex) Codiert. Jedes Byte entspricht einem Teil der IP-Adresse. Beispiel: Für die Adresse 172.23.2.156 wird der Wert DW#16#AC17029C angegeben. 0xAC = 172, 0x17 = 23, 0x02=2, 0x9C = 156.
UID	IN	INT	Der Unit Identifier (UID) wird ins Modbus Telegramm eingefügt. Wenn der Server ein Telegramm empfängt, so wird über die UID definiert, wie er die Daten verarbeiten muss. Auf einem Server können mehrere UIDs definiert werden. Auf dem Server müssen die UID's definiert sein (FB Init UID). Die UID 0 und 255 sind identische UIDs. Es wird empfohlen die UID 0 nicht zu benutzen. Bereich 0...255.
FunctionCode	IN	INT	Dieser Parameter definiert den Request Typ: WRITE_SINGLE_COIL            0x01 WRITE_SINGLE_REG            0x02 WRITE_MULTIPLE_COILS        0x03 WRITE_MULTIPLE_REGS        0x04
StartAddr	IN	DINT	Modbus Adresse des ersten zu schreibenden Elementes.
Length	IN	INT	Anzahl zu schreibender Modbus Elemente. Maximale Anzahl von Coils / DI:        1968 Maximale Anzahl von HR / IR:        123
Options	IN	DWORD	Optionen für den Write Request: Bit 0: Swap Words. Die 2 Worte eines Doppelwortes werden vor dem Senden gewappt. Das Swapping ist nur auf Register anwendbar. Die Anzahl Register muss geradzahlig sein. Bit 1...31 reserved
NDR	OUT	BOOL	Dieses Bit bleibt solange auf 0, wie der Write Request abgearbeitet wird, oder der Request nicht gestartet werden kann. Sobald der Request erfolgreich beendet wurde (eine korrekte Antwort wurde vom Server empfangen) wird dieses Bit auf 1 gesetzt.
Error	OUT	BOOL	Wird dieses Bit auf 1 gesetzt, so ist beim Ausführen des Requests ein Fehler aufgetreten. Entweder durch falsche Parameter oder ein Fehler wurde vom Server gesendet.
Exception Code	OUT	BYTE	Im Exception Code steht der Fehler Code, welcher der Modbus Server zurückgesendet hat. Falls kein Fehler vom Server gesendet wurde, ist dieser Wert auf 0 gesetzt.

Name	Typ	Format	Beschreibung
Status	OUT	WORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.
Data	IN / OUT	ANY	Der any pointer gibt an welche Daten gesendet werden sollen. Er kann auf DB's, Merker, Eingangs- und Ausgangsbereiche zeigen. Der Typ Bool ist nur erlaubt beim Lesen von coils, oder digitalem Eingang Beispiele: p#DB10.DBX0.0 BYTE 20 Es werden 20 Byte in den DB10 ab Byte 0 gesendet. p#E150.1 BOOL 9 Es werden 9 bits in den Eingangsbereich gesendet. Das erste Bit wird auf E150.1 kopiert.



Wird ein Write Request angestossen, so darf sich der Parameter Channel ID nicht ändern, bis der Schreib-Zugriff abgeschlossen wurde. (Error oder NDR Bit wurde gesetzt). Falls mehrerer Kanäle parallel geschrieben werden müssen, so ist für jeden SendWriteRequest Aufruf mit einer unterschiedlichen ChannelID mit einem separaten Instanz DB aufgerufen werden.

Es kann kein neuer Request angestossen werden, solange der aktuelle Request noch nicht abgeschlossen ist (NDR oder ERROR Bit wurde gesetzt).

### Asynchrone Funktionen

Die Send Read/Write Request FB's arbeiten asynchron zum Benutzerprogramm. Dies bedeutet, mit dem FB wird ein Hintergrund Prozess gestartet, und das Anwenderprogramm kann weiterarbeiten ohne auf das Ende des FBs zu warten. Dies hat zur folgen, dass der FB zyklisch mit den gleichen Parametern aufgerufen werden muss, bis der Hintergrund Prozess beendet wurde. Solange der Hintergrund Prozess noch aktiv ist, gibt der FB eine entsprechende Meldung im Status zurück (0x7001/0x7002). Falls in dieser Zeit versucht wird ein zweiter Prozess zu starten liefert der FB den Status Wert 0x7000 zurück.

In der nachfolgenden Tabelle sind die verschiedenen Kombinationen der Rückgabewerte aufgeführt:

Done / NDR	Error	Status	Description
0	0	0x7000	Ein Hintergrund Prozess ist bereits aktiv. Der neue Aufruf wird nicht ausgeführt.
0	0	0x7001 0x7002	Der aktuelle Hintergrund Prozess ist noch nicht beendet.
1	0	0	Der aktuelle Hintergrund Prozess ist ohne Fehler beendet.
0	1	= 0x 8 x x x	Der aktuelle Hintergrund Prozess wurde mit einem Fehler beendet.
0	0	0x8023	Die Response Timeout ist abgelaufen, der Prozess versucht ein Retry.

## 8.4 System Funktionen für Modbus Server

### 8.4.1 InitServer (FB500)

Ein Aufruf der Funktion InitServer erzeugt eine TCP/UDP Server Instanz. Eine Serverinstanz (Port/Protokoll) darf nicht mehrfach verwendet werden.

Aufruf:

```
CALL FB 500 , DB500
      Protocol   :=
      Port       :=
      Status     :=
```

Parameter Beschreibung:

Name	Typ	Format	Beschreibung
Protocol	IN	INT	IP- Protokoll über welches der Modbus kommuniziert: 0 = Modbus_TCP 1 = Modbus_UDP
Port	IN	INT	IP- Port Nummer auf welcher der Modbus Server öffnet. (Diese Nummer wird vom Client benötigt)
Status	OUT	WORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.

### 8.4.2 InitUID (FB502)

Die Funktion „InitUID“ erzeugt oder ändert eine UnitIdentifier (UID) auf der PCD Station.

Alle Server Instanzen können auf diese UID zugreifen. Die UID definiert wie der Server die Empfangenen Daten verarbeiten muss. Auf einer PCD können mehrere unterschiedliche UID's definiert werden.

Wird diese Funktion mit einer bereits bestehenden UID aufgerufen, wird die bestehende Konfiguration mit der Neuen überschrieben.

Aufruf:

```
CALL FB 502 , DB502
      UID       :=
      Options   :=
      Status    :=
```

Parameter Beschreibung:

Name	Typ	Format	Beschreibung
UID	IN	INT	Der Unit Identifier (UID) wird ins Modbus Telegramm eingefügt. Wenn der Server ein Telegram empfängt, so wird über die UID definiert, wie er die Daten verarbeiten muss. Auf einem Server können mehrere UIDs definiert werden. Auf dem Server müssen die UID's definiert sein (FB InitUID). Die UID 0 und 255 sind identische UIDs. Es wird empfohlen die UID 0 nicht zu benutzen. Bereich 0...255.
Options	IN	DWORD	Definiert wie der Server die Empfangenen Daten verarbeitet. <b>Bit 0: DEFMAP</b> Bestimmt in welchen Mapping Tabellen die Modbus Adresse gesucht werden. 0 = Es wird nur in den Benutzer-definierten Mapping Tabellen gesucht. 1 = Es wird zuerst in den Benutzer-definierten Mapping Tabellen gesucht und falls die Modbus Adresse nicht gefunden wird zusätzlich in der Default Mapping Tabelle. <b>Bit 1: OFFSET</b> 0 = Für Modbus Clients bei welchen die Adressierung bei 0 beginnt. (z.B. Saia PCD) 1 = Für Modbus Clients bei welchen die Adressierung bei 1 beginnt. (z.B. Modicon)
Status	OUT	DWORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB Fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.

### 8.4.3 InitMap (FB503)

Mit der Funktion InitMap wird das Mapping der Modbus Medien auf die PCD Medien definiert. Es können mehrere Mapping Bereiche definiert werden. Werden mehrere Mapping Bereiche definiert, so dürfen sich die Bereiche nicht überlappen.

Für Coils oder DI wird das Mapping Bitweise durchgeführt:

Coil oder DI Adresse	PCD Adresse (Merker, Input, Output)
0	X0.0
1	X0.1
8	X1.0

Für Register gelten folgend Mapping Regeln: Bei den PCD Medien Merker, Inputs und Outputs wird der PCD Range in Byte angegeben. PCD Start entspricht dem ersten Wort (MW, IW oder OW). Der Zugriff erfolgt Wortweise.

Bei Zeiten und Zählern entspricht PCDStart dem ersten Zähler oder Zeitglied und PCDRange entspricht der Anzahl Zeitglieder oder Zähler. Der Zugriff auf das ‚n-te‘ Register entspricht dem ‚n-ten‘ Zähler oder Zeitglied im Bereich.

**Aufruf:**

```
CALL FB 503 , DB503
    PartnerMedia      :=
    PartnerStart      :=
    PartnerRange      :=
    PCDMedia          :=
    PCDStart          :=
    PCDRange          :=
    AccessType        :=
    UID               :=
    Options           :=
    Status            :=
```

**Parameter Beschreibung:**

Name	Typ	Format	Beschreibung
PartnerMedia	IN	INT	Dieser Typ definiert den Typ der Modbus Daten auf welcher der Client zugreifen möchte. MB_NO_MEDIA                    0 MB_COILS_MEDIA                1 MB_DISCR_INPUT_MEDIA        2 MB_HOLDING_REG_MEDIA        3 MB_INPUT_REG_MEDIA          4
PartnerStart	IN	DINT	Modbus Adresse des ersten Elementes der Mapping Tabelle.
PartnerRange	IN	DINT	Anzahl Modbus Elemente der Mapping Tabelle
PCDMedia	IN	INT	PCD Medien Typ auf welche die Modbus Medien gemappt werden. XX7_NO_MEDIA                    0 XX7_MERKER                      1 XX7_TIMER                        2 XX7_COUNTER                    3 XX7_INPUT                        4 XX7_OUTPUT                      5
PCDStart	IN	DINT	Start Adresse der korrespondierenden PCD Medien
PCDRange	IN	DINT	Anzahl Elemente der PCD Medien. Es können maximale folgende Anzahl PCD Medien definiert werden: 4096 MerkerBytes 512 Timer 512 Counter 256 InputBytes 256 OutputBytes
AccessType	IN	INT	Modbus Zugriffs Typ: READ_WRITE                      0 READ_ONLY                        1 WRITE_ONLY                       2 NO_ACCESS                        3
UID	IN	INT	UID's die mit dem FB InitUID definiert wurden.
Options	IN	DWORD	<b>Bit 0:</b> Swap Words. Die 2 Worte eines Doppelwortes werden vor dem senden gewappt, bzw bei Empfangen gewappt. Das Swapping ist nur auf Register anwendbar. Die Anzahl Register muss geradzahlig sein. <b>Bit 1..31 reserved</b>

Name	Typ	Format	Beschreibung
Status	OUT	DWORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB Fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.

### 8.5 System Funktionen für Modbus Diagnose

Das Diagnose Register wird benützt um den Zustand der UID bzw. des Modbus Kanals zu erfahren. Es wird ebenfalls benützt um Server spezifische Fehler anzuzeigen.

Im Diagnose Register hat jedes Bit Einfluss auf den Retry Counter und die Response Timeout, d.h. die Bits sind Kumulativ. Die Bits müssen vom Anwender wieder Rückgesetzt werden (FB504 / FB505).

#### 8.5.1 Diagnose Register

	Bit	Designation	Description	Protocol	Used for
R E C E P T I O N	0	<i>Overrun Error</i>	Overrun of the internal reception buffer	Serial	Server
	1	<i>Parity Error</i>	Parity error in the received character	Serial	Server
	2	<i>Framing Error</i>	Incorrect baudrate	Serial	Server
	3	<i>Break Error</i>	Break on the serial line	Serial	Server
	4	<i>CRC Error</i>	CRC of the frame is incorrect	Serial	Server
	5	-	-		
	6	-	-		
	7	-	-		
	8	<i>Length Error</i>	Telegram with invalid length received	Serial	Server
	9	<i>Media Access Error</i>	Error while accessing PCD Media	All	UID
	10	-	-		
	11	<i>Server Start Error</i>	Server could not start	All	Server
	12	<i>Range Error</i>	Error in the address or number of Modbus elements to access (see below).	All	UID
	13	<i>Value Error</i>	A value in the telegram is invalid (Media type, ...)	All	UID
14	<i>Area access error</i>	Area not accessible (UID busy, area access not allowed)	All	UID	
T R A N S M I S S I O N	15	-	-		
	16	-	-	-	-
	17				
	18				
	19				
	20	-	-	-	-
	21	-	-	-	-
	22	-	-	-	-
	23	-	-	-	-
	24	-	-	-	-
	25	-	-	-	-
	26	-	-	-	-
	27	-	-	-	-
	28	-	-	-	-
	29	-	-	-	-
	30	-	-	-	-
	31	-	-	-	-

Der Fehler „Range Error“ kann mehrere Ursachen haben:

- kein korrespondierendes Mapping gefunden, kein Mappingbereich stimmt mit der Start Adresse und der Anzahl Modbus Elemente überein. (Sowohl in den Anwenderdefinierten Mappings als auch im Default Mapping)
- die Anzahl PCD Medien ist grösser als der erlaubte Bereich für den Zugriff auf die PCD Medien.  
Falls dieser Fehler gesetzt wird, sollte man die Adresse der Modbus Elemente und die Mapping Bereiche überprüft werden.

Der Fehler „Media Error“ wird gesetzt, wenn die Anzahl PCD Medien auf die zugegriffen werden soll die maximale Anzahl übersteigt, oder falls die PCD einen Fehler beim Zugriff feststellt.

### 8.5.2 GetUID\_Diag (FB504)

Die Funktion GetUID\_Diag liefert die Diagnose Informationen einer UID Instanz zurück.

Aufruf:

```
CALL FB 504 , DB504
    UID      :=
    ClearDiag :=
    DiagReg  :=
    ReqCounter :=
    Status   :=
```

Parameter Beschreibung:

Name	Typ	Format	Beschreibung
UID	IN	INT	Auf dem Server definierte UID (FB InitUID).
ClearDiag	IN	BYTE	Bit 0: setzt das Diagnose Register zurück Bit 1: setzt den Request Counter zurück
DiagReg	OUT	DWORD	Diagnose Register
ReqCounter	OUT	DWORD	Anzahl Requests auf diese UID
Status	OUT	DWORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.

### 8.5.3 GetServer\_Diag (FB505)

Die Funktion GetServer\_Diag liefert die Diagnose Informationen der Server Instanz zurück.

Aufruf:

```
CALL FB 505 , DB505
    Port      :=
    Protocol  :=
    ClearDiag :=
    DiagReg   :=
    Status    :=
```

Parameter Beschreibung:

Name	Typ	Format	Beschreibung
Port	IN	INT	IP- Port Nummer auf welcher der Modbus Server öffnet. (Diese Nummer wird vom Client benötigt)
Protocol	IN	INT	IP- Protokoll über welches der Modbus kommuniziert:
	0 = Modbus_TCP		
	1 = Modbus_UDP		
ClearDiag	IN	BYTE	Bit 0: setzt das Diagnose Register zurück
DiagReg	OUT	DWORD	Diagnose Register
Status	OUT	DWORD	Rückgabewert des FB's. Bei einem Wert 0x8... wurde der FB Fehlerhaft ausgeführt. (Fehlermeldungen siehe Kapitel 8.5) Ein Wert >= 0 zeigt die korrekte Ausführung an.

## 8.6 Fehler Meldungen

Im Falle einer fehlerlosen Ausführung eines Modbus FB's liefern diese den Status Wert 0 oder einen Wert grösser 0 zurück. Ein negativer Statuswert zeigt an, dass der FB mit einem Fehler beendet wurde.

8

### 8.6.1 Fehler Rückgabe von FB Aufrufen

Diese Fehler treten auf, wenn einer oder mehrere FB Parameter fehlerhaft sind.

Code	Designation	Description
0x8064	-	
0x8063	DATA_AREA_TOO_LONG	Area length (last element of the area) exceeds las media available
0x8062	DATA_AREA_TOO_SHORT	Area too short for the length specified (SendRequest)
0x8061	DATA_AREA_ERROR	Data area is not valid
0x8060	-	-
0x805F	-	-
0x805E	CHANNEL_NOT_DEFINED	Channel is not defined (Send Request)
0x805D	CHANNEL_BUSY	Tries to send a request on a channel already used for another request
0x805C	SERVER_NOT_CREATED	Server cannot be created (Init Server): max number of servers already reached or server already defined.
0x805B	SERVER_NOT_STARTED	Server cannot start (Init Server)
0x805A	INVALID_FUNCTION_CODE	Function code does not match request type (Read / Write)
0x8059	-	-
0x8058	INVALID_PROTOCOL	Protocol is invalid or does not correspond with FB.
0x8057	INVALID_PARTNER_TYPE	Modbus Media is not a valid type(Init Map)
0x8056	INVALID_PCDMEDIA_TYPE	PCD Media is not a valid type(Init Map)
0x8055	INVALID_PARTNER_ADDR	Partner start address is invalid (exceeds 0xFFFF = 65535)
0x8054	INVALID_PCD_MEDIA_ADDR	PCD Media address is invalid (Init Map): last element of the PCD media defined exceeds last media available

Code	Designation	Description
0x8053	INVALID_RANGE	Invalid media range (Init Map): Partner range < PCD range or range = 0.
0x8052	INVALID_AREA_ACCESS_TYPE	Access type is invalid (Init Map).
0x8051	INVALID_NB_OF_MEDIA	Number of media does not match function code (Send Request): > 1 for write single coil or single register, 0 for any request, swap set with an odd number of HR/IR...
0x8050	INVALID_UID	UID out of the range 0-255 (Init UID, Send Request) or UID = 0 for a read request in serial mode or UID not defined (GetUID_Diag)
0x804F	INVALID_CHANNEL	Channel out of the range 1-10 (Open Channel, Send Request)
0x804E	INCOMPATIBLE_PCDMEDIA	PCD media is incompatible with Modbus media (Init Map: Coils/DI □□Timers/Counters) or Function code (Send Request: tries to access HR or IR in a BOOL area).
0x804D	INVALID_PORT_CONF	Port configuration is invalid: check parameters (port number, databits, stopbits, baudrate, ...)
0x804C	-	-
0x804B	NO_PORT_TO_CLOSE	No serial port to close: for FB494, no channel or server related to the port passed as parameter.
0x804A	SRPORT_START_ERROR	SR port cannot be open (Init Server, Init SR Port). In serial, the port may be already used or port parameters are not correct for the port concerned.
0x8049	INVALID_PORT_NUMBER	SR Port Number is invalid or the port is not open
0x8048	INVALID_RETRY_NB	Number of retries invalid (must be < 16)

### 8.6.2 Modbus Shell Fehler Meldungen

Diese Fehler treten bei einem Fehler während der Ausführung auf:

Code	Designation	Description
0x8032	UID_TABLE_FULL	Maximum number of UIDs is already reached (Init UID)
0x8031	UID_NOT_DEFINED	UID is not defined (Init Map)
0x8030	UID_BUSY	UID is being used (Init UID), cannot modify it
0x802F	CHANNEL_TABLE_FULL	Channel table is full
0x802E	CHANNEL_ALREADY_DEFINED	A channel with same pair Port / Protocol is already defined (Open Channel)
0x802D	CHANNEL_BUSY	This channel is being used, cannot modify it
0x802C	MAP_TABLE_FULL	Maximum number of mappings reached for the UID (Init Map)
0x802B	MAP_AREA_OVERLAP	At least 2 map areas overlap (Init Map)
0x802A	CONNECTION_TABLE_FULL	Maximum number of "connections" is reached (Send Request). TCP/UDP: connection = {Port-Protocol-IPAddress} Serial: connection = {Port-Protocol}
0x8029	CONNECTION_NOT_READY	Connection is not ready for sending (Send Request). (internal error)
0x8028	DRIVER_ERROR	Error caused by Modbus driver (Internal error)
0x8027	CLIENT_START_ERROR	Client cannot start (no more socket available, ...)

Code	Designation	Description
0x8026	CLIENT_SEND_ERROR	Error while sending (in RTU/ASCII, check handshake)
0x8025	CLIENT_RESPONSE_ERROR	Error in response
0x8024	CLIENT_RESPONSE_EXCEPTION	Exception response received
0x8023	RESPONSE_TIMEOUT	Response not received after a timeout
0x8022	NO_RESPONSE_AFTER_RETRIES	Response not received after retries
0x8021	CLIENT_STATUS_ERROR	Invalid client status (internal error)
0x8020	SERIAL_ERROR	Error in serial communication (parity, framing, break, CRC,...)
0X801F	UNKNOWN_ERROR	Unknown error (internal error)
0X801E	MEDIA_ACCESS_ERROR	Error while accessing media (internal error)

### 8.6.3 Modbus Treiber Fehler Meldungen

Code	Designation	Description
0x8FA0	-	-
0x8F9F	UID_TABLE_FULL	UID table is full
0x8F9E	UID_NOT_DEFINED	Trying to access a non existing UID
0x8F9D	UID_BUSY	UID is busy
0x8F9C	INVALID_PARAMETER	Call with invalid parameters
0x8F9B	SI_ALREADY_DEFINED	Trying to create an already defined server instance
0x8F9A	SI_TABLE_FULL	Server instance table full
0x8F99	SI_NOT_DEFINED	Server instance not found in table
0x8F98	CLII_ALREADY_DEFINED	Client/Connection instance already defined
0x8F97	CLII_TABLE_FULL	Client/Connection table full
0x8F96	CLII_DRIVER_NOT_READY	Driver is not ready (Send Request)
0x8F95	TOO_MANY_CI	Too many client instances for this server instance
0x8F94	ADDRESS_ERROR	Address error in media processing
0x8F93	QUANTITY_ERROR	Error in the quantity of modbus media to read / write (Send Request): exceeds limit or < 0
0x8F92	INVALID_MEDIA	Invalid media for this function
0x8F91	MEDIA_ACCESS_ERROR	Media cannot be accessed correctly
0x8F90	CLIENT_INSTANCE_NOT_FOUND	Client/Connection instance does not exist
0x8F8F	CLIENT_NO_RESSOURCE	No resource available (Send Request)
0x8F8E	PORT_ID_ERROR	Port already initialized or used
0x8F8D	INVALID_PORT_STATE	Port not ready (Serial) (Send Request)
0x8F8C	UNKNOWN_ERROR	Unknown error

### 8.6.4 Ausnahme Fehler Meldungen (vom Server gesendet)

ILLEGAL_FUNCTION_CODE	1	Function code not valid or does not correspond to Modbus media type
ILLEGAL_DATA_ADDRESS	2	Modbus media at received address cannot be accessed (check UID / see diagnostic)
ILLEGAL_DATA_VALUE	3	Number of data in the request is invalid
SERVER_FAILURE	4	PCD Media could not be accessed (see diagnostic) or UID not configured (in particular UID 255 in TCP/UDP)
ACKNOWLEDGE	5	
SERVER_BUSY	6	Server is already busy when trying to access it
GATEWAY_PATH_NOT_FOUND	10	UID not defined
GATEWAY_TARGET_PROBLEM	11	

## A Anhang

### A.1 Icons

	<p>Dieses Symbol verweist den Leser innerhalb eines Handbuchs auf weiterführende Informationen in diesem oder einem anderen Handbuch, oder in technischen Informationsbroschüren. In der Regel besteht kein direkter Link zu diesen Dokumenten.</p>
	<p>Dieses Symbol warnt den Leser vor dem Risiko elektrischer Entladung durch Berühren. <b>Empfehlung:</b> Bevor Sie in Kontakt mit elektronischen Bauteilen kommen, sollten Sie zumindest vorher den Minuspol des Systems (Gehäuse der PGU-Buchse) berühren. Besser ist es, permanent mit einer Erdungsglasche am Handgelenk mit dem Minuspol verbunden zu sein.</p>
	<p>Dieses Zeichen steht neben Anweisungen, die befolgt werden müssen.</p>
	<p>Erklärungen neben diesem Zeichen sind nur für die Saia PCD Classic Serie gültig.</p>
	<p>Erklärungen neben diesem Zeichen sind nur für die Saia PCD xx7 Serie gültig.</p>

**A.2 Adresse****Saia-Burgess Controls AG**

Bahnhofstrasse 18  
CH-3280 Murten | Schweiz

Telefon +41 26 / 672 71 11  
Telefax +41 26 / 672 74 43  
E-mail: [support@saia-pcd.com](mailto:support@saia-pcd.com)  
Homepage: [www.saia-pcd.com](http://www.saia-pcd.com)  
Technischer Support: [www.sbc-support.com](http://www.sbc-support.com)

**Postadresse für Rücksendungen  
von Kunden des Verkauf Schweiz:****Saia-Burgess Controls AG**

Service Après-Vente  
Bahnhofstrasse 18  
CH-3280 Murten / Schweiz