



Handbuch

Protokolle

Send/Receive TCP



Alle in diesem Handbuch genannten HIMA Produkte sind mit dem Warenzeichen geschützt. Dies gilt ebenfalls, soweit nicht anders vermerkt, für weitere genannte Hersteller und deren Produkte.

HIQuad®, HIQuad®X, HIMax®, HIMatrix®, SILworX®, XMR®, HICore® und FlexSILon® sind eingetragene Warenzeichen der HIMA Paul Hildebrandt GmbH.

Alle technischen Angaben und Hinweise in diesem Handbuch wurden mit größter Sorgfalt erarbeitet und unter Einschaltung wirksamer Kontrollmaßnahmen zusammengestellt. Bei Fragen bitte direkt an HIMA wenden. Für Anregungen, z. B. welche Informationen noch in das Handbuch aufgenommen werden sollen, ist HIMA dankbar.

Technische Änderungen vorbehalten. Ferner behält sich HIMA vor, Aktualisierungen des schriftlichen Materials ohne vorherige Ankündigungen vorzunehmen.

Alle aktuellen Handbücher können über die E-Mail-Adresse documentation@hima.com angefragt werden.

© Copyright 2019, HIMA Paul Hildebrandt GmbH

Alle Rechte vorbehalten.

Kontakt

HIMA Paul Hildebrandt GmbH

Postfach 1261

68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

E-Mail: info@hima.com

Revisions- index	Änderungen	Art der Änderung	
		technisch	redaktionell
11.00	Neu erstellt für SILworX V11	X	X

Inhaltsverzeichnis

1	Einleitung	5
1.1	Aufbau und Gebrauch des Handbuchs	5
1.2	Zielgruppe	6
1.3	Darstellungskonventionen	7
1.3.1	Sicherheitshinweise	7
1.3.2	Gebrauchshinweise	8
1.4	Safety Lifecycle Services	9
2	Sicherheit	10
2.1	Bestimmungsgemäßer Einsatz	10
2.2	Restrisiken	10
2.3	Sicherheitsvorkehrungen	10
2.4	Notfallinformationen	10
2.5	Automation Security bei HIMA Systemen	10
3	Send/Receive TCP	12
3.1	Systemanforderung	12
3.1.1	Anlegen eines S/R-TCP-Protokolls	12
3.2	Beispiel: S/R TCP Konfiguration	13
3.2.1	S/R TCP Konfiguration der Siemens Steuerung SIMATIC 300	15
3.2.2	S/R TCP Konfiguration der HIMax Steuerung	19
3.3	Menüfunktionen S/R-TCP-Protokoll	21
3.3.1	Edit	21
3.3.2	Eigenschaften	21
3.3.3	CPU/COM	23
3.4	Menüfunktionen TCP Verbindung	24
3.4.1	Edit	24
3.4.2	Systemvariablen	24
3.4.3	Eigenschaften	25
3.5	Datenaustausch	27
3.5.1	TCP-Verbindungen	27
3.5.2	Zyklischer Datenaustausch	28
3.5.3	Azyklischer Datenaustausch mit Funktionsbausteinen	28
3.5.4	Gleichzeitiger zyklischer und azyklischer Datenaustausch	28
3.5.5	Flusskontrolle	29
3.6	Fremdsysteme mit Pad Bytes	29
3.7	S/R-TCP-Funktionsbausteine	30
3.7.1	TCP_Reset	31
3.7.2	TCP_Send	34
3.7.3	TCP_Receive	37
3.7.4	TCP_ReceiveLine	41
3.7.5	TCP_ReceiveVar	45
3.8	Control-Panel (Send/Receive over TCP)	50
3.8.1	Anzeigefeld allgemeine Parameter	50
3.8.2	Anzeigefeld TCP Verbindungen	50
3.8.3	Fehlercode der TCP-Verbindung	51
3.8.4	Zusätzliche Fehlercodetabelle der Funktionsbausteine	52

3.8.5	Verbindungszustand	52
3.8.6	Partner Verbindungszustand	52
4	Allgemein	53
4.1	Maximale Kommunikationszeitscheibe	53
4.1.1	Ermitteln der maximalen Dauer der Kommunikationszeitscheibe	53
4.2	Lastbegrenzung	53
4.3	Konfiguration der Funktionsbausteine	54
4.3.1	Beschaffung der Funktionsbausteinbibliotheken	54
4.3.2	Konfiguration der Funktionsbausteine im Anwenderprogramm	54
4.3.3	Konfiguration der Funktionsbausteine im Strukturbaum von SILworX	55
	Anhang	56
	Glossar	56
	Abbildungsverzeichnis	57
	Tabellenverzeichnis	57

1 Einleitung

Das Send/Receive TCP Handbuch beschreibt die Eigenschaften und die Konfiguration des Send/Receive TCP Protokolls für die sicherheitsbezogenen HIMA Steuerungssysteme mit dem Programmierwerkzeug SILworX.

Die Kenntnis von Vorschriften und das technisch einwandfreie Umsetzen der in diesem Handbuch enthaltenen Hinweise durch qualifiziertes Personal sind Voraussetzung für die Planung, Projektierung, Programmierung, Installation, Inbetriebnahme, Betrieb und Instandhaltung der HIMA Steuerungen.

Bei nicht qualifizierten Eingriffen in die Geräte, bei Abschalten oder Umgehen (Bypass) von Sicherheitsfunktionen oder bei Nichtbeachtung von Hinweisen dieses Handbuchs (und dadurch verursachten Störungen oder Beeinträchtigungen von Sicherheitsfunktionen) können schwere Personen-, Sach- oder Umweltschäden eintreten, für die HIMA keine Haftung übernehmen kann.

HIMA Automatisierungsgeräte werden unter Beachtung der einschlägigen Sicherheitsnormen entwickelt, gefertigt und geprüft. Nur für die in den Beschreibungen vorgesehenen Einsatzfälle mit den spezifizierten Umgebungsbedingungen verwenden.

1.1 Aufbau und Gebrauch des Handbuchs

Das Handbuch enthält die folgenden Hauptkapitel:

- Einleitung
- Sicherheit
- Produktbeschreibung
- Send/Receive TCP

Zusätzlich sind die folgenden Dokumente zu beachten:

Name	Inhalt	Dokumenten-Nr.
HIMax Systemhandbuch	Hardware-Beschreibung HIMax System	HI 801 000 D
HIMax Sicherheitshandbuch	Sicherheitsfunktionen HIMax Systems	HI 801 002 D
HIMatrix Sicherheitshandbuch	Sicherheitsfunktionen HIMatrix Systems	HI 800 022 D
HIMatrix Kompakt Systemhandbuch	Hardware-Beschreibung HIMatrix Kompakt System	HI 800 140 D
HIMatrix Modular Systemhandbuch	Hardware-Beschreibung HIMatrix Modular System F 60	HI 800 190 D
HIQuad X Systemhandbuch	Hardware-Beschreibung HIQuad X System	HI 803 210 D
HIQuad X Sicherheitshandbuch	Sicherheitsfunktionen HIQuad X System	HI 803 208 D
Automation Security Handbuch	Beschreibung von Automation Security Aspekten bei HIMA Systemen	HI 801 372 D
SILworX Erste Schritte	Einführung in SILworX	HI 801 102 D

Tabelle 1: Zusätzlich geltende Handbücher

Die aktuellen Handbücher können über die E-Mail-Adresse documentation@hima.com angefragt werden. Für registrierte Kunden stellt HIMA die Dokumentationen im Download-Bereich <https://www.hima.com/de/downloads/> zur Verfügung.

1.2 Zielgruppe

Dieses Dokument wendet sich an Planer, Projektoren, Programmierer und Personen die zur Inbetriebnahme, zur Wartung und zum Betreiben von Automatisierungsanlagen berechtigt sind. Vorausgesetzt werden spezielle Kenntnisse auf dem Gebiet der sicherheitsbezogenen Automatisierungssysteme.

1.3 Darstellungskonventionen

Zur besseren Lesbarkeit und zur Verdeutlichung gelten in diesem Dokument folgende Schreibweisen:

Fett	Hervorhebung wichtiger Textteile. Bezeichnungen von Schaltflächen, Menüpunkten und Registern im Programmierwerkzeug, die angeklickt werden können.
<i>Kursiv</i>	Parameter und Systemvariablen, Referenzen.
<i>Courier</i>	Wörtliche Benutzereingaben.
RUN	Bezeichnungen von Betriebszuständen (Großbuchstaben).
Kap. 1.2.3	Querverweise sind Hyperlinks, auch wenn sie nicht besonders gekennzeichnet sind. Im elektronischen Dokument (PDF): Wird der Mauszeiger auf einen Hyperlink positioniert, verändert er seine Gestalt. Bei einem Klick springt das Dokument zur betreffenden Stelle.

Sicherheits- und Gebrauchshinweise sind besonders gekennzeichnet.

1.3.1 Sicherheitshinweise

Um ein möglichst geringes Risiko zu gewährleisten, sind die Sicherheitshinweise unbedingt zu befolgen.

Die Sicherheitshinweise im Dokument sind wie folgt dargestellt.

- Signalwort: Warnung, Vorsicht, Hinweis.
- Art und Quelle des Risikos.
- Folgen bei Nichtbeachtung.
- Vermeidung des Risikos.

Die Bedeutung der Signalworte ist:

- Warnung: Bei Missachtung droht schwere Körperverletzung bis Tod.
- Vorsicht: Bei Missachtung droht leichte Körperverletzung.
- Hinweis: Bei Missachtung droht Sachschaden.

SIGNALWORT



Art und Quelle des Risikos!
Folgen bei Nichtbeachtung.
Vermeidung des Risikos.

HINWEIS



Art und Quelle des Schadens!
Vermeidung des Schadens.

1.3.2 Gebrauchshinweise

Zusatzinformationen sind nach folgendem Beispiel aufgebaut:

i

An dieser Stelle steht der Text der Zusatzinformation.

Nützliche Tipps und Tricks erscheinen in der Form:

TIPP

An dieser Stelle steht der Text des Tipps.

1.4 Safety Lifecycle Services

HIMA unterstützt Sie in allen Phasen des Sicherheitslebenszyklus der Anlage: Von der Planung, der Projektierung, über die Inbetriebnahme, bis zur Aufrechterhaltung der Sicherheit.

Für Informationen und Fragen zu unseren Produkten, zu Funktionaler Sicherheit und zu Automation Security stehen Ihnen die Experten des HIMA Support zur Verfügung.

Für die geforderte Qualifizierung gemäß Sicherheitsstandards, führt HIMA produkt- oder kundenspezifische Seminare in eigenen Trainingszentren, oder bei Ihnen vor Ort durch. Das aktuelle Seminarangebot zu Funktionaler Sicherheit, Automation Security und zu HIMA Produkten finden Sie auf der HIMA Webseite.

Safety Lifecycle Services:

Onsite+ / Vor-Ort-Engineering	In enger Abstimmung mit Ihnen führt HIMA vor Ort Änderungen oder Erweiterungen durch.
Startup+ / Vorbeugende Wartung	HIMA ist verantwortlich für die Planung und Durchführung der vorbeugenden Wartung. Wartungsarbeiten erfolgen gemäß der Herstellervorgabe und werden für den Kunden dokumentiert.
Lifecycle+ / Lifecycle-Management	Im Rahmen des Lifecycle-Managements analysiert HIMA den aktuellen Status aller installierten Systeme und erstellt konkrete Empfehlungen zu Wartung, Upgrade und Migration.
Hotline+ / 24-h-Hotline	HIMA Sicherheitsingenieure stehen Ihnen für Problemlösung rund um die Uhr telefonisch zur Verfügung.
Standby+ / 24-h-Rufbereitschaft	Fehler, die nicht telefonisch gelöst werden können, werden von HIMA Spezialisten innerhalb vertraglich festgelegter Zeitfenster bearbeitet.
Logistic+/ 24-h-Ersatzteilservice	HIMA hält notwendige Ersatzteile vor und garantiert eine schnelle und langfristige Verfügbarkeit.

Ansprechpartner:

Safety Lifecycle Services	https://www.hima.com/de/unternehmen/ansprechpartner-weltweit/
Technischer Support	https://www.hima.com/de/produkte-services/support/
Seminarangebot	https://www.hima.com/de/produkte-services/seminarangebot/

2 Sicherheit

Sicherheitsinformationen, Hinweise und Anweisungen in diesem Dokument unbedingt lesen. Das Produkt nur unter Beachtung aller Richtlinien und Sicherheitsrichtlinien einsetzen.

Dieses Produkt wird mit SELV oder PELV betrieben. Vom Produkt selbst geht kein Risiko aus. Einsatz im Ex-Bereich nur mit zusätzlichen Maßnahmen erlaubt.

2.1 Bestimmungsgemäßer Einsatz

Für den Einsatz von HIMA Steuerungen, sind die jeweiligen Bedingungen einzuhalten, siehe zusätzlich geltende Handbücher 1.1.

2.2 Restrisiken

Von einem HIMA System selbst geht kein Risiko aus.

Restrisiken können ausgehen von:

- Fehlern in der Projektierung.
- Fehlern im Anwenderprogramm.
- Fehlern in der Verdrahtung.

2.3 Sicherheitsvorkehrungen

Am Einsatzort geltende Sicherheitsbestimmungen beachten und vorgeschriebene Schutzausrüstung tragen.

2.4 Notfallinformationen

Ein HIMA System ist Teil der Sicherheitstechnik einer Anlage. Der Ausfall einer Steuerung bringt die Anlage in den sicheren Zustand.

Im Notfall ist jeder Eingriff, der die Sicherheitsfunktion des HIMA Systems verhindert, verboten.

2.5 Automation Security bei HIMA Systemen

Automation Security hat die Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit von Daten. In Bezug auf Automation Security muss von gezielten Angriffen ausgegangen werden. Insbesondere an Schnittstellen, wie sie in diesem Handbuch beschrieben werden, ist von möglichen Angriffen auszugehen.

WARNUNG



Personenschaden durch unbefugte Manipulation an der Steuerung möglich!

Die Steuerung ist gegen unbefugte Zugriffe zu schützen!

Die für eine Anlage geeignete Umsetzung der benötigten Maßnahmen liegt in der Verantwortung des Anwenders!

Sorgfältige Planung sollte die zu ergreifenden Maßnahmen nennen. Nach erfolgter Risikoanalyse sind die benötigten Maßnahmen zu ergreifen. Solche Maßnahmen sind beispielsweise:

- Sinnvolle Einteilung von Benutzergruppen.
- Gepflegte Netzwerkpäne helfen sicherzustellen, dass secure Netzwerke dauerhaft von öffentlichen Netzwerken getrennt sind und, falls nötig, nur ein definierter Übergang (z. B. über eine Firewall oder eine DMZ) besteht.

- Verwendung geeigneter Passwörter.

Ein regelmäßiges Review (z. B. jährlich) der Security-Maßnahmen ist ratsam.

Weitere Einzelheiten siehe HIMA Automation Security Handbuch HI 801 372 D.

3 Send/Receive TCP

Send/Receive TCP (S/R TCP) ist ein herstellerunabhängiges Standardprotokoll für zyklischen und azyklischen Datenaustausch und verwendet außer TCP/IP kein spezielles Protokoll.

Mit dem S/R TCP-Protokoll unterstützt die HIMA Steuerung nahezu jedes Fremdsystem sowie PC's mit vorhandener Socket-Schnittstelle (z. B. Winsock.dll) zu TCP/IP.

S/R TCP ist kompatibel zu der Siemens SEND/RECEIVE Schnittstelle und erlaubt die Kommunikation mit Siemens Steuerungen über TCP/IP. Der Datenaustausch erfolgt über die S7-Funktionsbausteine AG_SEND (FC5) und AG_RECV (FC6).

3.1 Systemanforderung

Benötigte Ausstattung und Systemanforderung:

Element	Beschreibung
Steuerung	HIMax mit COM-Modul HIMatrix F*03 HIMatrix F*01/02
CPU-Modul	Die Ethernet-Schnittstellen des Prozessormoduls können für S/R TCP nicht verwendet werden.
COM-Modul	Ethernet 10/100BaseT Es kann für jedes COM-Modul ein S/R-TCP-Protokoll konfiguriert werden.
Aktivierung	Die Freischaltung erfolgt per Software-Freischaltcode, siehe Kommunikationshandbuch HI 801 100 D.

Tabelle 2: Systemanforderung und Ausstattung S/R TCP

Eigenschaften des S/R-TCP-Protokolls:

Element	Beschreibung
Sicherheitsbezogen	Nein
Datenaustausch	Zyklischer und azyklischer Datenaustausch über TCP/IP.
Funktionsbausteine	Die S/R-TCP-Funktionsbausteine müssen beim azyklischen Datenaustausch verwendet werden.
TCP-Verbindungen	Bis zu 32 TCP-Verbindungen können in einer Steuerung konfiguriert werden, sofern nicht die maximale Größe der Sendedaten oder Empfangsdaten überschritten wird.
Max. Größe der Sendedaten	Siehe Kommunikationshandbuch HI 801 100 D.
Max. Größe der Empfangsdaten	<p>i Um die maximale Anzahl der Nutzdaten zu ermitteln, müssen alle Statusvariablen der verwendeten TCP-Verbindungen und der TCP/SR Funktionsbausteine von der maximalen Anzahl Sendedaten abgezogen werden. Die Aufteilung auf die einzelnen TCP-Verbindungen ist beliebig.</p>

Tabelle 3: Eigenschaften S/R TCP

3.1.1 Anlegen eines S/R-TCP-Protokolls

Ein neues S/R-TCP-Protokoll anlegen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle** öffnen.
2. Im Kontextmenü von Protokolle **Neu, Send/Receive over TCP** wählen, um ein neues S/R-TCP-Protokoll hinzuzufügen.
3. Im Kontextmenü vom Send/Receive-over-TCP-Protokoll **Eigenschaften, Allgemein** das **COM-Modul** auswählen.

3.2 Beispiel: S/R TCP Konfiguration

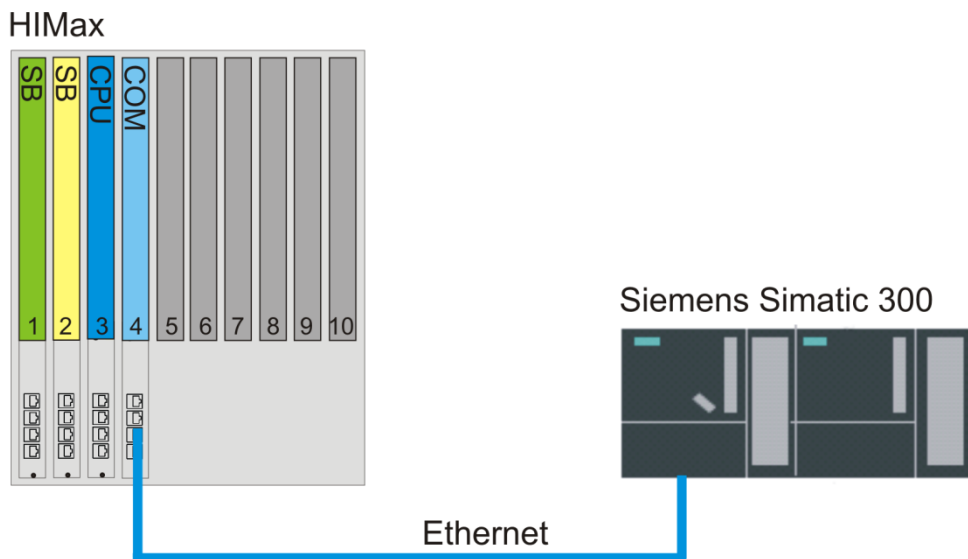


Bild 1: Verschaltung HIMA und Siemens Steuerung

In diesem Beispiel wird das Protokoll Send/Receive over TCP in einer HIMA Steuerung eingerichtet. Die HIMA soll zyklisch über S/R TCP mit einer Siemens Steuerung (z. B. SIMATIC 300) kommunizieren.

Dabei ist die HIMA (Client) die aktive Station, welche die TCP-Verbindung zur passiven Siemens SIMATIC 300 (Server) aufbaut. Nach dem Verbindungsaufbau sind aber beide Steuerungen gleichberechtigt und können jederzeit senden und empfangen.

Bei der Zusammenschaltung der HIMA und der Siemens SIMATIC 300 ist folgendes zu beachten:

- Für die HIMA gelten die im Abschnitt 3.1 (Systemanforderungen) beschriebenen Anforderungen.
- Die HIMA und die Siemens SIMATIC 300 werden über ihre Ethernet Schnittstellen miteinander verbunden.
- Die HIMA und die Siemens SIMATIC 300 müssen sich im gleichen Subnet befinden oder bei Verwendung eines Routers die entsprechenden Routing Einträge besitzen.

In diesem Beispiel sollen zwei BYTES und ein WORD von der HIMA Steuerung zur Siemens SIMATIC 300 gesendet werden. Die Variablen werden in der SIMATIC 300 vom Baustein AG_RECV (FC 6) empfangen und intern an den Baustein AG_SEND (FC 5) übergeben. Über den Baustein AG_SEND (FC 5) sendet die SIMATIC 300 die Variablen unverändert an die HIMA Steuerung zurück.

Die Übertragung der Variablen kann der Anwender nach der Konfiguration mit dem HIMA Force-Editor prüfen.

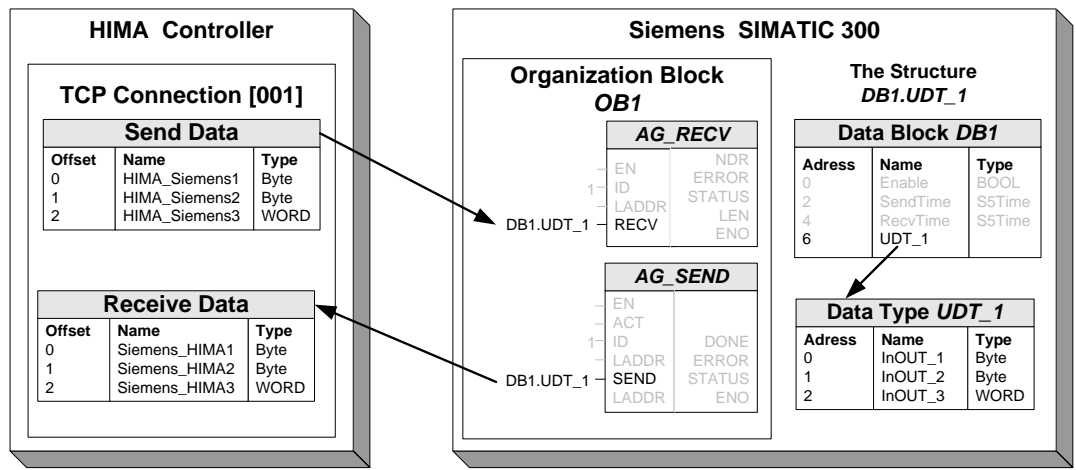


Bild 2: Datenübertragung zwischen HIMax und Siemens Steuerung

Konfigurationsbeschreibung HIMax Steuerung:

Element	Beschreibung
TCP-Verbindung [001]	In diesem Dialog stehen alle Parameter, die für die Kommunikation mit dem Kommunikationspartner (Siemens SIMATIC 300) notwendig sind.
Sendedaten	Die Offsets und Typen der Variablen in der Steuerung müssen mit der Adresse und den Typen der Variablen im Datentyp <i>UDT_1</i> der SIMATIC 300 übereinstimmen.
Empfangsdaten	Die Offsets und Typen der Variablen in der HIMax Steuerung müssen mit der Adresse und den Typen der Variablen im Datentyp <i>UDT_1</i> der SIMATIC 300 übereinstimmen.

Tabelle 4: Konfiguration HIMax Steuerung

Konfigurationsbeschreibung Siemens SIMATIC 300:

Element	Beschreibung
Organisationsbaustein OB1	Die Funktionsbausteine <i>AG_RECV</i> (FC6) und <i>AG_SEND</i> (FC 5) müssen im Organisationsbaustein <i>OB1</i> angelegt und konfiguriert werden.
<i>AG_RECV</i> (FC 6)	Der Funktionsbaustein <i>AG_RECV</i> (FC 6) übernimmt die empfangenen Daten vom Kommunikationspartner in den Datentyp <i>DB1.UDT_1</i> . Die Eingänge <i>ID</i> und <i>LADDR</i> müssen für die Kommunikation mit dem Kommunikationspartner entsprechend konfiguriert werden.
<i>AG_SEND</i> (FC 5)	Der Funktionsbaustein <i>AG_SEND</i> (FC 5) überträgt die Daten aus dem Datentyp <i>DB1.UDT_1</i> zum Kommunikationspartner. Die Eingänge <i>ID</i> und <i>LADDR</i> müssen für die Kommunikation mit dem Kommunikationspartner entsprechend konfiguriert werden.
Datenbaustein <i>DB1</i>	Der Datentyp <i>UDT_1</i> wird im Datenbaustein <i>DB1</i> definiert.
Datentyp <i>UDT_1</i>	Die Adressen und Typen der Variablen in der SIMATIC 300 müssen mit den Offsets und den Typen der Steuerung übereinstimmen. Der Datentyp <i>UDT_1</i> übernimmt die empfangenen Nutzdaten und speichert diese bis zur Übertragung an den Kommunikationspartner.

Tabelle 5: Konfiguration Siemens SIMATIC 300

3.2.1 S/R TCP Konfiguration der Siemens Steuerung SIMATIC 300

i

Die folgende Schrittanleitung zur Konfiguration der Siemens-Steuerung erhebt keinen Anspruch auf Vollständigkeit.

Alle Angaben sind ohne Gewähr, maßgebend zur Projektierung der Siemens-Steuerung ist die Dokumentation von Siemens.

Erstellen des S/R-TCP-Server im Projekt der SIMATIC 300:

1. SIMATIC Manager starten.
2. Im SIMATIC Manager das Projekt der SIMATIC 300 Steuerung öffnen.
3. In diesem Projekt das *Industrial Ethernet* und die *MPI* Verbindungen Erstellen und Konfigurieren.

Erstellen des Datentyp UDT1 mit den folgenden Variablen:

1. Ordner *Bausteine* im Siemens SIMATIC Manager öffnen.
2. Im Hauptmenü **Einfügen, S7 Baustein, Datentyp** öffnen und einen Datentyp erstellen.
3. Datentyp den Namen **UDT1** zuweisen
4. Datentyp den Symbolischen Namen **UDT_1** zuweisen.
5. Im Datentyp *UDT_1* die drei Variablen **InOut_x** wie im Bild unten erstellen.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	InOut_1	BYTE	B#16#0	
+1.0	InOut_2	BYTE	B#16#0	
+2.0	InOut_3	WORD	W#16#0	
=4.0		END_STRUCT		

Bild 3: Variablenliste im Siemens-Baustein UDT1

i

Beim zyklischem und azyklischen Datenaustausch ist zu beachten, dass manche Steuerungen (z. B. SIMATIC 300) so genannte *Pad Bytes* einfügen. Damit wird sichergestellt, dass alle Datentypen, die größer als ein Byte sind, immer an einem geraden Offset beginnen und dass die Gesamtlänge aller definierten Variablen ebenfalls immer gerade ist.

In diesen Fällen müssen an den entsprechenden Stellen Dummy-Bytes in der HiMax Steuerung eingefügt werden (siehe Kapitel 3.6, Fremdsysteme mit Pad Bytes).

Erstellen des Datenbaustein DB1 für die Funktionsbausteine FC 5 und FC 6:

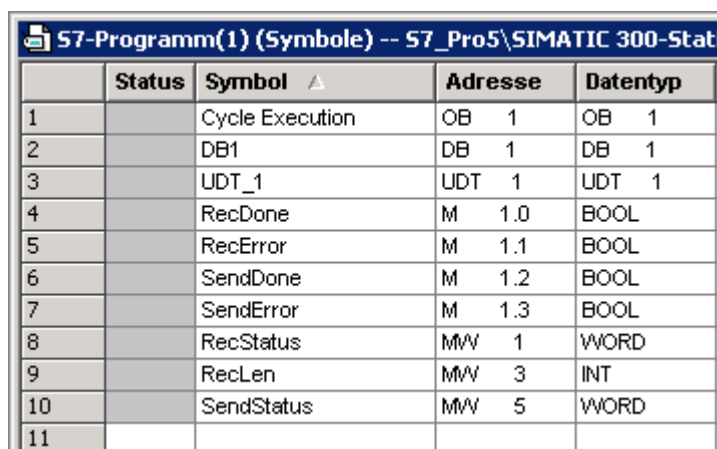
1. Im Hauptmenü **Einfügen, S7 Baustein, Datenbaustein** wählen und Datenbaustein erstellen.
2. Datenbaustein den Namen **DB1** zuweisen.
3. Datenbaustein den Symbolischen Namen **DB1** zuweisen.
4. Dem Datenbaustein **DB1** den Datentyp **UDT_1** zuweisen.
5. Im Datenbaustein **DB1** die Datentypen wie im Bild unten dargestellt parametrieren.

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	Enable	BOOL	FALSE
+2.0	SendTime	S5TIME	S5T#100MS
+4.0	RecvTime	S5TIME	S5T#10MS
+6.0	UDT_1	"UDT_1"	
=10.0		END_STRUCT	

Bild 4: Variablenliste im Siemens-Baustein DB1

Im Symboleditor die folgenden Symbole erstellen

1. Dialogfenster **KOP/AWL/FUP** mit einem Doppelklick auf den Organisationsbaustein **OB1** öffnen.
2. Im Hauptmenü **Extras, Symboltabelle** den Symboleditor öffnen.
3. Den **Symboleditor** mit den Variablen **M 1.0...MW 5** wie im Bild unten dargestellt ergänzen.



	Status	Symbol ▲	Adresse	Datentyp
1		Cycle Execution	OB 1	OB 1
2		DB1	DB 1	DB 1
3		UDT_1	UDT 1	UDT 1
4		RecDone	M 1.0	BOOL
5		RecError	M 1.1	BOOL
6		SendDone	M 1.2	BOOL
7		SendError	M 1.3	BOOL
8		RecStatus	MW 1	WORD
9		RecLen	MW 3	INT
10		SendStatus	MW 5	WORD
11				

Bild 5: SIMATIC-Symboleditor

Erstellen des FC-Baustein AG_RECV (FC 6):

1. Dialogfenster *KOP/AWL/FUP* öffnen.
2. Folgende FC-Bausteine nacheinander aus der Struktur im linken Teil des Symatic-Managers auswählen:
 - 1 *Oder-Gatter*
 - 1 *S_VIMP*
 - 1 *AG_RECV (FC 6)*.
3. Diese *Funktionsbausteine* per Drag&Drop in den Organisationsbaustein *OB1* ziehen.
4. *Funktionsbausteine* wie im Bild unten dargestellt Verbinden und Konfigurieren.
5. Rechtsklick auf den FC-Baustein **AG_RECV (FC 6)** und **Eigenschaften** wählen.
6. **Aktiver Verbindungsaufbau** deaktivieren und Ports Konfigurieren.
7. Bausteinparameter *LADDR* notieren und diese im Funktionsplan am Baustein *AG_RECV (FC 6)* eintragen.

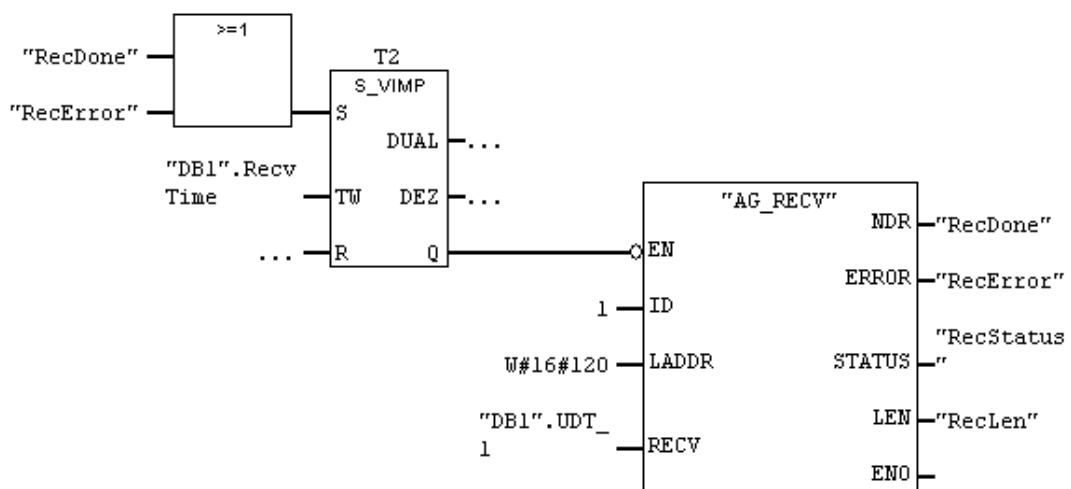


Bild 6: Funktionsplan zum Empfangen

Erstellen des FC-Baustein AG_SEND (FC 5):

1. Dialogfenster *KOP/AWL/FUP* öffnen.
2. Folgende FC-Bausteine nacheinander aus der Struktur im linken Teil des Symatic-Managers auswählen:
 1 *Oder-Gatter*
 1 *S_VIMP*
 1 *AG_SEND (FC 5)*.
3. Diese *Funktionsbausteine* per Drag&Drop in den Organisationsbaustein *OB1* ziehen.
4. *Funktionsbausteine* wie im Bild unten dargestellt verbinden und konfigurieren.
5. Rechtsklick auf den FC-Baustein **AG_SEND (FC 5)** und **Eigenschaften** wählen.
6. **Aktiver Verbindungsaufbau** deaktivieren und Ports Konfigurieren.
7. Bausteinparameter *LADDR* notieren und diese im Funktionsplan am Baustein *AG_SEND (FC 5)* eintragen.

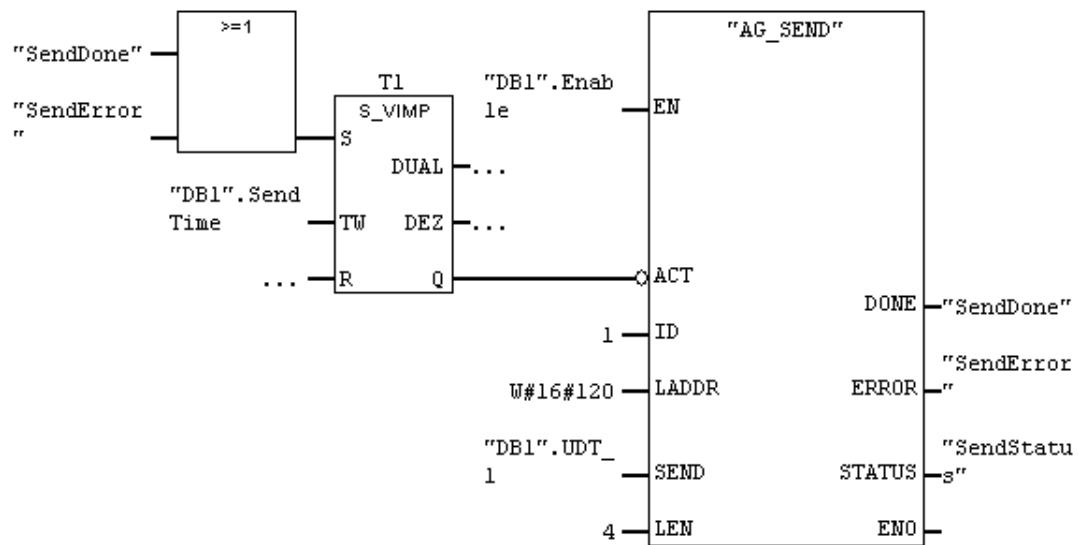


Bild 7: Funktionsplan zum Senden

Code in die SIMATIC 300 Steuerung laden:

1. **Code Generator** für das Programm starten.
2. Sicherstellen, dass die Codes fehlerfrei generiert wurden.
3. Code in die SIMATIC 300 Steuerung laden.

3.2.2 S/R TCP Konfiguration der HIMax Steuerung

Für die Konfiguration der HIMax Steuerungen und den Umgang mit dem Programmierwerkzeug SILworX wird das Handbuch "Erste Schritte SILworX" empfohlen.

Folgende Globalen Variablen im Variableneditor erstellen:

1. Im Strukturbaum **Konfiguration, Globale Variablen** wählen.
2. Rechtsklick auf **Globale Variablen** und **Edit** wählen.
3. Globale Variablen wie in der Tabelle 6 erstellen.

Name	Typ
Siemens_HIMA1	Byte
Siemens_HIMA2	Byte
Siemens_HIMA3	WORD
HIMA_Siemens1	Byte
HIMA_Siemens2	Byte
HIMA_Siemens3	WORD

Tabelle 6: Globale Variablen

Erstellen des S/R-TCP-Protokolls in der Ressource:

1. Im Strukturbaum **Konfiguration, Ressource** öffnen.
2. Rechtsklick auf **Protokolle** und im Kontextmenü **Neu** wählen.
3. **Send/Receive over TCP** wählen und einen Namen für das Protokoll eintragen.
4. Mit OK bestätigen, um ein neues Protokoll zu erstellen.
5. Rechtsklick auf **Send/Receive over TCP** und im Kontextmenü **Eigenschaften** wählen.
6. **COM Modul** auswählen. Die restlichen Parameter behalten die Standardwerte.

Erstellen der TCP Verbindung:

1. Rechtsklick auf **Send/Receive over TCP** und im Kontextmenü **Neu** wählen.
2. Rechtsklick auf **TCP Verbindung** und im Kontextmenü **Eigenschaften** wählen.
3. Eigenschaften wie im Bild unten konfigurieren.

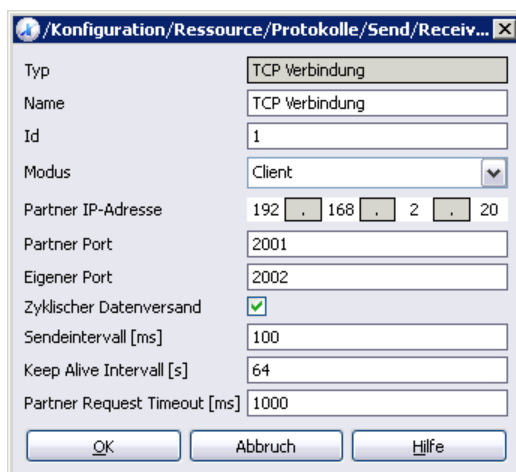


Bild 8: Eigenschaften der TCP-Verbindung in SILworX



Wenn ein zyklischer Datenaustausch zwischen zwei Steuerungen parametrieren soll, muss im Dialog *Eigenschaften* der TCP-Verbindung die Option *Zyklischer Datenversand* aktiviert sein.

Empfangsdaten der HIMA Steuerung konfigurieren.

1. Rechtsklick auf **TCP-Verbindung** und im Kontextmenü **Edit** wählen.
2. Register **Prozessvariablen** wählen.
3. In der **Objektauswahl** die folgenden Globalen Variablen auswählen und diese per Drag&Drop in den Bereich **Eingangssignale** ziehen.

Globale Variable	Typ
Siemens_HIMA1	Byte
Siemens_HIMA2	Byte
Siemens_HIMA3	WORD

Tabelle 7: Variablen für Empfangsdaten

4. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Register Eingänge** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.



Beachten Sie, dass die Offsets der Variablen in der HIMA Steuerung mit den Adressen der Variablen im Datentyp *UDT_1* der SIMATIC 300 übereinstimmen müssen.

Sendedaten der HIMA Steuerung konfigurieren.

1. Rechtsklick auf **TCP-Verbindung** und im Kontextmenü **Edit** wählen.
2. Register **Prozessvariablen** wählen.
3. In der **Objektauswahl** die folgenden Globalen Variablen auswählen und diese per Drag&Drop in den Bereich **Eingangssignale** ziehen.

Globale Variable	Typ
HIMA_Siemens1	Byte
HIMA_Siemens2	Byte
HIMA_Siemens3	WORD

Tabelle 8: Variablen für Sendedaten

4. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Register Eingänge** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.



Beachten Sie, dass die Offsets der Variablen in der HIMA Steuerung mit den Adressen der Variablen im Datentyp *UDT_1* der SIMATIC 300 übereinstimmen müssen.

S/R TCP Konfiguration verifizieren:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Send/Receive over TCP** selektieren.
2. Rechtsklick und im Kontextmenü **Verifikation** wählen.
3. Einträge im Logbuch sorgfältig überprüfen, gegebenenfalls korrigieren.

3.3 Menüfunktionen S/R-TCP-Protokoll

3.3.1 Edit

Das Dialogfenster **Edit** des S/R-TCP-Protokolls enthält das Register **Systemvariablen**.

Mit den *Systemvariablen* kann der Zustand vom S/R-TCP-Protokoll im Anwenderprogramm ausgewertet werden.

Element	Beschreibung
Anzahl aktive Verbindungen	Systemvariable, welche die Anzahl aktiver (ungestörter) Verbindungen liefert.
Anzahl gestörte Verbindungen	Systemvariable, welche die Anzahl gestörter Verbindungen liefert. Gestört bedeutet, dass die TCP Verbindung durch einen Timeout oder Fehler unterbrochen wurde.
Status	Keine Funktion.

Tabelle 9: Systemvariablen S/R TCP

3.3.2 Eigenschaften

Der Datenaustausch über eine TCP-Verbindung erfolgt entweder zyklisch oder azyklisch. Für den azyklischen Datenaustausch werden die S/R-TCP-Funktionsbausteine benötigt.

Zyklischer und nicht zyklischer Datenaustausch können nicht zusammen auf einer Verbindung verwendet werden.

3.3.2.1 Allgemein

Name	Beschreibung
Typ	Send/Receive over TCP.
Name	Name für das Send/Receive-over-TCP-Protokoll.
Modul	Auswahl des COM-Moduls auf dem dieses Protokoll abgearbeitet wird.
Max. μ P-Budget aktivieren	Aktiviert: Limit des μ P-Budget aus dem Feld <i>Max. μP-Budget in [%]</i> übernehmen. Deaktiviert: Kein Limit des μ P-Budget, für dieses Protokoll verwenden.
Max. μ P-Budget in [%]	Maximales μ P-Budget des Moduls, welches bei der Abarbeitung des Protokolls produziert werden darf. Wertebereich: 1 ... 100% Standardwert: 30 %
Verhalten bei CPU/COM Verbindungsverlust	Bei Verbindungsverlust des Prozessormoduls zum Kommunikationsmodul werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert im Prozessormodul verwendet. (z. B. wenn Kommunikationsmodul bei laufender Kommunikation gezogen wird). Soll ein Projekt von kleiner SILworX V3 konvertiert werden, muss dieser Wert auf "Letzten Wert beibehalten" gesetzt sein, um den CRC nicht zu ändern. Für HiMatrix Steuerungen kleiner CPU BS V8 muss dieser Wert immer auf "Letzten Wert beibehalten" gesetzt sein. Initialdaten annehmen: Eingangsvariablen werden auf die Initialwerte zurückgesetzt. Letzten Wert beibehalten: Eingangsvariablen behalten den letzten Wert.

Tabelle 10: Allgemeine Eigenschaften S/R TCP

3.3.3 CPU/COM

Die vorgegebenen Parameter sorgen für den schnellstmöglichen Datenaustausch der S/R-TCP-Daten zwischen dem COM-Modul (COM) und dem CPU-Modul (CPU) in der Steuerung. Diese Parameter sollten nur dann geändert werden, wenn eine Reduzierung der COM- und/oder CPU-Auslastung für eine Anwendung erforderlich sind und der Prozess dies zulässt.

i

Die Änderung der Parameter wird nur dem erfahrenen Programmierer empfohlen. Eine Erhöhung der COM und CPU Aktualisierungszeit bedeutet auch, dass die tatsächliche Aktualisierungszeit der S/R TCP-Daten erhöht wird. Die Zeitanforderungen der Anlage sind zu prüfen.

Name	Beschreibung
Aktualisierungsintervall der Prozessdaten [ms]	Aktualisierungszeit in Millisekunden, mit der die Daten im S/R-TCP-Protokoll zwischen COM und CPU ausgetauscht werden. Ist die Refresh Rate Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich. Wertebereich: 0 ... (2 ³¹ -1) Standardwert: 0
Prozessdaten-Konsistenz erzwingen	Aktiviert: Transfer der S/R TCP-Daten von der CPU zur COM innerhalb eines Zyklus der CPU. Deaktiviert: Transfer der S/R-TCP-Daten (maximal 1100 Byte pro Datenrichtung) von der CPU zur COM über mehrere Zyklen der CPU.

Tabelle 11: Parameter COM/CPU

3.4 Menüfunktionen TCP Verbindung

3.4.1 Edit

Über die Menüfunktion **Edit** sind die Register **Prozessvariablen** und **Systemvariablen** zu erreichen.

3.4.1.1 Prozessvariablen

Eingangssignale

Die Variablen für den zyklischen Datenaustausch, die von dieser Steuerung empfangen werden sollen, werden im Bereich *Eingangssignale* eingetragen.

Im Register *Eingangssignale* können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen allerdings identisch mit den Offsets und den Typen der Variablen (Sendedaten) des Kommunikationspartners sein.

Ausgangssignale

Die Variablen für den zyklischen Datenaustausch, die von dieser Steuerung gesendet werden sollen, werden im Bereich *Ausgangssignale* eingetragen.

Im Register *Ausgangssignale* können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen allerdings identisch mit den Offsets und den Typen der Variablen (Empfangsdaten) des Kommunikationspartners sein.

3.4.2 Systemvariablen

Mit den Variablen im Register **Systemvariablen** kann der Zustand der TCP-Verbindung im Anwenderprogramm ausgewertet werden.

Name	Datentyp	R/W	Beschreibung
Bytes empfangen	UDINT	W	Anzahl Bytes, die bisher empfangen wurden.
Bytes versenden	UDINT	W	Anzahl Bytes, die bisher gesendet wurden.
Errorcode	UDINT	W	Fehlercode der TCP-Verbindung, siehe Kapitel 3.8.3.
Errorcode Zeitstempel [ms]	UDINT	W	Millisekunden-Anteil des Zeitstempels. Zeitpunkt des Fehlereintritts.
Errorcode Zeitstempel [s]	UDINT	W	Sekunden-Anteil des Zeitstempels. Zeitpunkt des Fehlereintritts.
Partner Request Timeout	UDINT	W	Bei zyklischer Datenübertragung: Timeoutzeit, innerhalb der nach Datenversand mindestens eine Datensendung vom Kommunikationspartner empfangen werden muss. 0 = Aus 1 ... (2 ³¹ -1) [ms]
Partner Verbindungszustand	BYTE	W	Wird keine Datensendung innerhalb der Timeoutzeit empfangen, so wird der Status Partner Verbindungszustand auf <i>nicht verbunden</i> gesetzt und die Verbindung neu aufgesetzt. 0 = Keine Verbindung. 1 = Verbindung OK.
Status	DWORD	W	Verbindungsstatus der TCP-Verbindung, siehe Kapitel 3.8.5.

Tabelle 12: Systemvariablen

3.4.3 Eigenschaften

Der Datenaustausch über eine TCP-Verbindung erfolgt entweder zyklisch oder azyklisch. Für den azyklischen Datenaustausch werden die S/R-TCP-Funktionsbausteine benötigt. Beim zyklischen Datenverkehr ist der Betrieb von S/R-TCP-Funktionsbausteinen nicht möglich.

Name	Beschreibung
Typ	TCP-Verbindung.
Name	Beliebiger, eindeutiger Name für eine TCP-Verbindung.
Id	Beliebige, aber eindeutige Identifikationsnummer ID für jede TCP-Verbindung. Die ID wird auch als Referenz in den S/R-TCP-Funktionsbausteinen benötigt. Wertebereich 0 ... 255 Standardwert: 0
Modus	Server (Standardwert): Diese Station arbeitet als Server, d. h. im passiven Modus. Der Verbindungsaufbau muss durch den Kommunikationspartner (Client) initiiert werden. Nach dem ersten Verbindungsaufbau sind aber beide Stationen gleichberechtigt und können zu jedem Zeitpunkt Daten senden. Benötigt wird die Angabe des eigenen Ports.
	Server mit definiertem Partner: Diese Station arbeitet als Server, d. h. im passiven Modus. Der Verbindungsaufbau muss durch den Kommunikationspartner (Client) initiiert werden. Nach dem ersten Verbindungsaufbau sind aber beide Stationen gleichberechtigt und können zu jedem Zeitpunkt Daten senden. Wird hier die IP-Adresse und/oder Port des Kommunikationspartners eingetragen, dann kann nur der definierte Kommunikationspartner eine Verbindung aufnehmen. Alle anderen Stationen werden ignoriert. Wird einer der Parameter (IP-Adresse oder Port) auf Null gesetzt, findet für diesen Parameter keine Überprüfung statt.
	Client: Diese Station arbeitet als Client, d. h. die Station initiiert den Verbindungsaufbau mit dem Kommunikationspartner. Benötigt die Angabe von IP-Adresse und Port des Kommunikationspartners. Optional kann auch ein eigener Port angegeben werden.
Partner IP-Adresse	IP-Adresse des Kommunikationspartners. 0.0.0.0: beliebige IP-Adresse ist erlaubt. Gültiger Bereich: 1.0.0.0 ... 223.255.255.255, außer: 127.x.x.x Standardwert: 0
Partner Port	Port des Kommunikationspartners. 0: beliebiger Port. Reservierte oder bereits belegte Ports (1 ... 1024) werden von dem COM-BS abgelehnt. Wertebereich 0 ... 65535 Standardwert: 0
Eigener Port	Eigener Port. 0: beliebiger Port. Reservierte oder bereits belegte Ports (1...1024) werden vom COM-BS abgelehnt. Wertebereich 0 ... 65535 Standardwert: 0

Name	Beschreibung
Zyklischer Datenversand	Deaktiviert (Standardwert): Zyklischer Datenversand ist deaktiviert. Der Datenaustausch über diese TCP-Verbindung muss mit Funktionsbausteinen programmiert werden. Es dürfen auf dieser Verbindung keine zyklischen E/A-Daten definiert sein.
	Aktiviert: Zyklischer Datenversand ist aktiv. Die Daten werden im Dialog Prozessvariablen der TCP-Verbindung definiert. Es müssen Empfangsdaten definiert sein. Es können auf dieser Verbindung keine Funktionsbausteine betrieben werden.
Sendeintervall [ms]	Nur editierbar bei zyklischem Datenversand. Hier wird das Sendeintervall eingestellt. Wertebereich 10 ... 2 147 483 647 ms (Kleinere Werte werden auf 10 ms aufgerundet). Standardwert: 0
KeepAlive Intervall [s]	Ist die Zeit, bis die von TCP bereitgestellte Verbindungsüberwachung aktiv wird. Null deaktiviert die Verbindungsüberwachung. Werden innerhalb des eingestellten KeepAlive-Intervalls keine Daten ausgetauscht, werden KeepAlive-Proben an den Kommunikationspartner geschickt. Besteht die Verbindung noch, werden die KeepAlive-Proben vom Kommunikationspartner bestätigt. Findet in einer Zeit > 10 KeepAlive Intervall kein Datenaustausch zwischen den Partnern statt, wird die Verbindung geschlossen. Wird nach dem Senden eines Datenpakets keine Antwort empfangen so wird in bestimmten Intervallen das Datenpaket wiederholt. Die Verbindung wird nach 12 erfolglosen Wiederholungen abgebrochen (ca. 7 Minuten). Wertebereich 1 ... 65535s Standardwert: 0 = deaktiviert
Partner Request Timeout [ms]	Bei zyklischer Datenübertragung: Timeoutzeit, innerhalb der nach Datenversand mindestens eine Datensendung vom Kommunikationspartner empfangen werden muss. Wird keine Datensendung innerhalb der Timeoutzeit empfangen, so wird der <i>Partner Verbindungszustand</i> auf <i>nicht verbunden</i> gesetzt und die Verbindung neu aufgesetzt. Nach dem Schließen der Verbindung durch ein Timeout oder einen anderen Fehler baut die aktive Seite die Verbindung mit einer Verzögerung von 10 x PartnerRequestTimeout bzw. 10 s, wenn PartnerRequestTimeout = 0 ist, neu auf. Die passive Seite öffnet den Port bereits nach der halben Zeit. 0 = Aus Wertebereich 1 ... (2 ³¹ -1) [ms] Standardwert: 0

Tabelle 13: Eigenschaften S/R TCP-Verbindung

3.5 Datenaustausch

S/R TCP arbeitet gemäß dem Client/Server Prinzip. Der Verbindungsaufbau muss durch den Kommunikationspartner initiiert werden, der als Client konfiguriert ist. Nach dem ersten Verbindungsaufbau sind aber beide Kommunikationspartner gleichberechtigt und können zu jedem Zeitpunkt Daten senden.

S/R TCP besitzt kein eigenes Protokoll zur Datensicherung, sondern benutzt dafür direkt das TCP/IP Protokoll. Da TCP die Daten in einem „Daten-Stream“ sendet, muss sichergestellt sein, dass die Offsets und die Typen der auszutauschenden Variablen auf der Empfangsseite und auf der Sendeseite identisch sind.

S/R TCP ist kompatibel zu der Siemens SEND/RECEIVE-Schnittstelle und erlaubt den zyklischen Datenaustausch mit den Siemens S7-Funktionsbausteinen AG_SEND (FC5) und AG_RECV (FC6) (siehe Kapitel 3.2, Beispiel S/R TCP Konfiguration).

Zudem stellt HIMA fünf S/R-TCP-Funktionsbausteine bereit, mit denen die Kommunikation über das Anwenderprogramm gesteuert und individuell angepasst werden kann. Mit den S/R-TCP-Funktionsbausteinen können beliebige Protokolle (z. B. Modbus), die über TCP übertragen werden, gesendet und empfangen werden.

3.5.1 TCP-Verbindungen

Für jede Verbindung über S/R TCP mit einem Kommunikationspartner muss mindestens eine TCP-Verbindung in der HIMax Steuerung erstellt werden.

In den *Eigenschaften* der TCP-Verbindung müssen die Identifikationsnummer der TCP-Verbindung und die Adressen/Ports der eigenen Steuerung und des Kommunikationspartners eingetragen werden.

Maximal 32 TCP-Verbindungen können in einer HIMax Steuerung erstellt werden.

Die erstellten TCP-Verbindungen müssen unterschiedliche Identifikationsnummern und unterschiedliche Adressen/Ports besitzen.

Erstellen einer neuen TCP-Verbindung:

1. Im Strukturbaum **Konfiguration, Ressource** öffnen.
2. Rechtsklick auf **Protokolle** und im Kontextmenü **Neu** wählen.
3. **Send/Receive over TCP** wählen und einen Namen für das Protokoll eintragen.
4. Mit OK bestätigen, um ein neues Protokoll zu erstellen.
5. Rechtsklick auf **Send/Receive over TCP** und im Kontextmenü **Eigenschaften** wählen.
6. **COM Modul** auswählen. Die restlichen Parameter behalten die Standardwerte.

TIPP Die HIMA Steuerung und das Fremdsystem müssen sich im gleichen Subnet befinden oder bei Verwendung eines Routers die entsprechenden Routinginträge besitzen.

3.5.2 Zyklischer Datenaustausch

Wird zyklischer Datenaustausch verwendet, dann muss ein Sendeintervall in der HIMA Steuerung und im Kommunikationspartner festgelegt werden.

Das Sendeintervall legt das zyklische Intervall fest, innerhalb dessen der sendende Kommunikationspartner seine Variablen an den empfangenden Kommunikationspartner sendet.

- Um einen kontinuierlichen Datenaustausch zu gewährleisten, sollte bei beiden Kommunikationspartnern ungefähr das gleiche Sendeintervall festgelegt werden (siehe Kapitel 3.5.5, Flusskontrolle).
- Die Option *Zyklischer Datenversand* muss in der verwendeten TCP-Verbindung für den zyklischen Datenaustausch aktiviert sein.
- In einer TCP-Verbindung, in der die Option *zyklischer Datenversand* aktiviert ist, dürfen keine Funktionsbausteine verwendet werden.
- Die zu sendenden und zu empfangenden Variablen werden im Dialogfenster *Prozessvariablen* der TCP-Verbindung zugewiesen. Empfangsdaten **müssen** vorhanden sein, Sendedaten sind optional.

i

Die gleichen Variablen (gleiche Offsets und Typen), die in der einen Station als Sendedaten definiert sind, müssen in der anderen Station als Empfangsdaten definiert werden.

3.5.3 Azyklischer Datenaustausch mit Funktionsbausteinen

Der azyklische Datenaustausch wird in der HIMA Steuerung vom Anwenderprogramm über die S/R-TCP-Funktionsbausteine gesteuert.

Somit ist es möglich, mit einem Timer oder einen mechanischen Schalter an einem physikalischen Eingang der HIMA Steuerung, den Datenaustausch zu steuern.

- Die Option *Zyklischer Datenversand* muss in der verwendeten TCP-Verbindung deaktiviert sein.
- Zu einem Zeitpunkt darf immer nur ein TCP S/R Funktionsbaustein senden.
- Die zu sendenden oder zu empfangenden Variablen werden im Dialog *Prozessvariablen* der S/R-TCP-Funktionsbausteine (alle außer *Reset*) zugewiesen.

i

Die gleichen Variablen (gleiche Offsets und Typen), die in der einen Station als Sendedaten definiert sind, müssen in der anderen Station als Empfangsdaten definiert werden.

3.5.4 Gleichzeitiger zyklischer und azyklischer Datenaustausch

Hierzu muss eine TCP-Verbindung für zyklische Daten und eine zweite für azyklische Daten konfiguriert werden. Beide TCP-Verbindungen müssen unterschiedliche *Partner IP-Adressen* und *Partner Ports* verwenden.

Eine einzelne TCP-Verbindung kann nicht für zyklischen und azyklischen Datenaustausch gemeinsam verwendet werden.

3.5.5 Flusskontrolle

Die Flusskontrolle ist ein Bestandteil von TCP und überwacht den kontinuierlichen Datenverkehr zwischen zwei Kommunikationspartnern.

Bei zyklischer Datenübertragung muss nach maximal 3 bis 5 gesendeten Paketen mindestens ein Paket empfangen werden, sonst blockiert das Senden, bis wieder ein Paket empfangen wird oder die Verbindungsüberwachung die Verbindung schließt.

Die Anzahl (3 ... 5) der möglichen Sendungen ohne Paketerhalt ist abhängig von der Größe der zu sendenden Pakete.

Anzahl = 5 für kleine Pakete < 4 kB.

Anzahl = 3 für große Pakete \geq 4 kB.

- Bei der Projektierung ist darauf zu achten, dass keine der beiden Stationen mehr Daten sendet, als die andere synchron verarbeiten kann.
- Für den zyklischen Datenaustausch muss bei beiden Kommunikationspartnern ungefähr das gleiche Sendeintervall eingestellt werden.

3.6 Fremdsysteme mit Pad Bytes

Beim zyklischen und azyklischen Datenaustausch ist zu beachten, dass manche Steuerungen (z. B. SIMATIC 300) sogenannte *Pad Bytes* einfügen. Damit wird sichergestellt, dass alle Datentypen die größer als ein Byte sind, immer an einem geraden Offset beginnen und dass die Gesamtlänge der Pakete (in Byte) ebenfalls immer gerade ist.

In der HIMA Steuerung müssen für die *Pad Bytes*, *Dummy-Bytes* an den entsprechenden Stellen eingefügt werden.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	InOut_1	BYTE	B#16#0
+2.0	InOut_3	WORD	W#16#0
=4.0		END_STRUCT	

Bild 9: Variablenliste Siemens

In der Siemens Steuerung wird ein *Pad-Byte* (nicht sichtbar) eingefügt, damit die Variable *InOut_3* an einem geraden Offset beginnt.

Ausgangssignale					
F	Name	Datentyp	ffs	Transfer-Operation	Globale Variable
1	InOut_1	BYTE	0	5	InOut_1
2	Dummy	BYTE	1	5	Dummy
3	InOut_3	WORD	2	5	InOut_3

Bild 10: Variablenliste HIMA

In der HIMA Steuerung muss ein *Dummy-Byte* eingefügt werden, damit die Variable *InOut_3* den gleichen Offset wie in der Siemens Steuerung hat.

3.7 S/R-TCP-Funktionsbausteine

Wenn die zyklische Datenübertragung zu unflexibel ist, können Daten auch mittels der S/R-TCP-Funktionsbausteine gesendet und empfangen werden. Die Option *Zyklischer Datenversand* muss in der verwendeten TCP-Verbindung deaktiviert werden.

Mit den S/R-TCP-Funktionsbausteinen kann der Anwender die Datenübertragung über TCP/IP den Erfordernissen seines Projekts optimal anpassen.

Die Funktionsbausteine werden im Anwenderprogramm parametriert. So können die Funktionen (Senden, Empfangen, Reset) der HIMA Steuerung im Anwenderprogramm gesetzt und ausgewertet werden.

S/R-TCP-Funktionsbausteine werden nur für den azyklischen Datenaustausch benötigt. Für den zyklischen Datenaustausch zwischen Server und Client sind diese Funktionsbausteine nicht erforderlich!

i

Die Konfiguration der S/R-TCP-Funktionsbausteine wird im Kapitel 4.3 beschrieben.

Es stehen die folgenden Funktionsbausteine zur Verfügung:

Funktionsbaustein	Beschreibung der Funktion
TCP_Reset (siehe Kapitel 3.7.1)	Rücksetzen einer TCP-Verbindung.
TCP_Send (siehe Kapitel 3.7.2)	Senden von Daten.
TCP_Receive (siehe Kapitel 3.7.3)	Empfangen von Datenpaketen fester Länge.
TCP_ReceiveLine (siehe Kapitel 3.7.4)	Empfang einer ASCII-Zeile.
TCP_ReceiveVar (siehe Kapitel 3.7.5)	Empfangen von Datenpaketen variabler Länge (mit Längenfeld).
LATCH	Wird nur innerhalb anderer Funktionsbausteine verwendet.
PIG	Wird nur innerhalb anderer Funktionsbausteine verwendet.
PIGII	Wird nur innerhalb anderer Funktionsbausteine verwendet.

Tabelle 14: Funktionsbausteine für S/R TCP Verbindungen

3.7.1 TCP_Reset

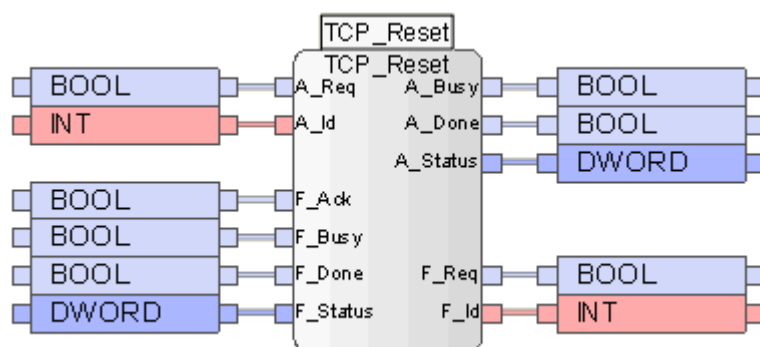


Bild 11: Funktionsbaustein TCP_Reset

Mit dem Funktionsbaustein **TCP Reset** kann eine gestörte Verbindung wiederhergestellt werden, wenn sich ein Send- oder Receive-Funktionsbaustein mit einem TIMEOUT-Fehler meldet (16#8A).

i

Zur Konfiguration den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm ziehen, siehe auch Kapitel 4.3.

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix A:

Über diese Ein- und Ausgänge kann der Funktionsbaustein mit Hilfe des Anwenderprogramms gesteuert und ausgewertet werden. Das Präfix "A" steht für "Application".

A-Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet den Baustein.	BOOL
A_Id	Identifikationsnummer <i>ID</i> der gestörten TCP-Verbindung, die zurückgesetzt werden soll.	INT

Tabelle 15: A-Eingänge Funktionsbaustein TCP_Reset

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Reset des Funktionsbausteins ist noch nicht beendet.	BOOL
A_Done	TRUE: Der Sendevorgang wurde fehlerfrei beendet.	BOOL
A_Status	Am Ausgang <i>A_Status</i> wird Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD

Tabelle 16: A-Ausgänge Funktionsbaustein TCP_Reset

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F:

Diese Ein- und Ausgänge des Funktionsbausteins stellen die Verbindung zum Funktionsbaustein **Reset** im Strukturbaum her. Das Präfix "F" steht für "Field".

i

Die Verbindung des Funktionsbausteins **Reset** im Strukturbaum (im Ordner Funktionsbausteine) mit dem Funktionsbaustein **TCP_Reset** (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor vom Anwender im Variableneditor erstellt werden.

Die *F-Eingänge* des Funktionsbausteins **TCP_Reset** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Ausgänge des Funktionsbausteins **Reset** im Strukturbaum verbunden werden.

F-Eingänge	Typ
F_Ack	BOOL
F_Busy	BOOL
F_Done	BOOL
F_Status	DWORD

Tabelle 17: F-Eingänge Funktionsbaustein TCP_Reset

Die *F-Ausgänge* des Funktionsbausteins **TCP_Reset** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Eingänge des Funktionsbausteins **Reset** im Strukturbaum verbunden werden.

F-Ausgänge	Typ
F_Req	BOOL
F_Id	DWORD

Tabelle 18: F-Ausgänge Funktionsbaustein TCP_Reset

Zugehörigen Funktionsbaustein Reset im Strukturbaum erstellen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Send Receive over TCP, Funktionsbausteine, Neu** öffnen.
2. Funktionsbaustein **Reset** wählen.
3. Rechtsklick auf Funktionsbaustein **Reset** und **Edit** wählen
☒ Variablenzuweisung zum Funktionsbaustein wird geöffnet.

Die Eingänge des Funktionsbausteins **Reset** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Ausgänge* des Funktionsbausteins **TCP_Reset** im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	DWORD
REQ	BOOL

Tabelle 19: Eingangs-Systemvariablen

Die Ausgänge des Funktionsbausteins **Reset** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Eingänge* des Funktionsbausteins **TCP_Reset** im Anwenderprogramm verbunden wurden.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Tabelle 20: Ausgangs-Systemvariablen

Für die Bedienung des Funktionsbaustein TCP_Reset sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm am Eingang *A_Id* die Identifikationsnummer der gestörten TCP-Verbindung setzen.
2. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen.

i

Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis ein Reset an die definierte TCP-Verbindung gesendet wurde. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Done* auf TRUE.

3.7.2 TCP_Send

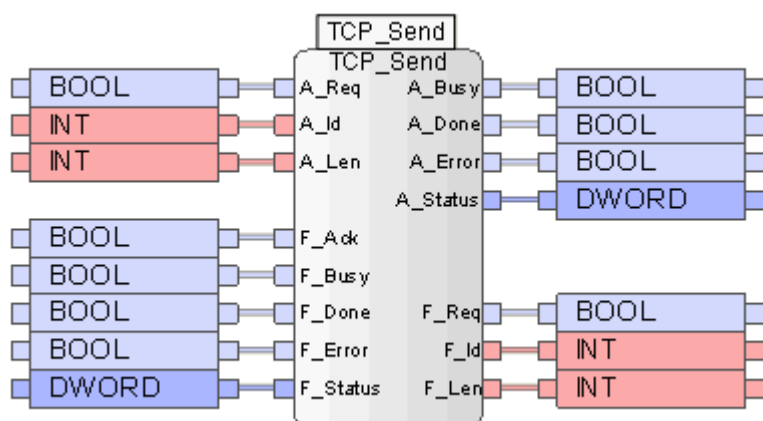


Bild 12: Funktionsbaustein TCP_Send

Der Funktionsbaustein **TCP_Send** dient zum azyklischen Senden von Variablen zu einem Kommunikationspartner. Im Kommunikationspartner muss ein Funktionsbaustein z. B. *Empfangen* mit den gleichen Variablen und Offsets konfiguriert werden.

i

Zur Konfiguration den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm ziehen, siehe auch Kapitel 4.3.

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix A:

Über diese Ein- und Ausgänge kann der Funktionsbaustein mit Hilfe des Anwenderprogramms gesteuert und ausgewertet werden. Das Präfix "A" steht für "Application".

A-Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet den Baustein.	BOOL
A_Id	Identifikationsnummer der konfigurierten TCP-Verbindung zu dem Kommunikationspartner, zu welchem die Daten gesendet werden sollen.	INT
A_Len	Anzahl der zu sendenden Variablen in Bytes. A_Len muss größer als Null sein und darf nicht innerhalb einer Variablen enden.	INT

Tabelle 21: A-Eingänge Funktionsbaustein TCP_Send

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Sendevorgang ist noch nicht beendet.	BOOL
A_Done	TRUE: Der Sendevorgang wurde fehlerfrei beendet.	BOOL
A_Error	TRUE: Ein Fehler ist aufgetreten. FALSE: Kein Fehler.	BOOL
A_Status	Am Ausgang A_Status wird Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD

Tabelle 22: A-Ausgänge Funktionsbaustein TCP_Send

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F:

Diese Ein- und Ausgänge des Funktionsbausteins stellen die Verbindung zum Funktionsbaustein **Senden** im Strukturbaum her. Das Präfix "F" steht für "Field".

i

Die Verbindung des Funktionsbausteins **Senden** im Strukturbaum (im Ordner Funktionsbausteine) mit dem Funktionsbaustein **TCP_Send** (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor vom Anwender im Variableneditor erstellt werden.

Die *F-Eingänge* des Funktionsbausteins **TCP_Send** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Ausgänge des Funktionsbausteins **Senden** im Strukturbaum verbunden werden.

F-Eingänge	Typ
F_Ack	BOOL
F_Busy	BOOL
F_Done	BOOL
F_Error	BOOL
F_Status	DWORD

Tabelle 23: F-Eingänge Funktionsbaustein TCP_Send

Die *F-Ausgänge* des Funktionsbausteins **TCP_Send** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Eingänge des Funktionsbausteins **Senden** im Strukturbaum verbunden werden.

F-Ausgänge	Typ
F_Id	DWORD
F_Len	INT
F_Req	BOOL

Tabelle 24: F-Ausgänge Funktionsbaustein TCP_Send

Zugehörigen Funktionsbaustein Senden im Strukturbaum erstellen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Send Receive over TCP, Funktionsbausteine, Neu** öffnen.
2. Funktionsbaustein **Senden** wählen.
3. Rechtsklick auf Funktionsbaustein **Senden** und **Edit** wählen
 - ☒ Variablenzuweisung zum Funktionsbaustein wird geöffnet.

Die Eingänge des Funktionsbausteins **Senden** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Ausgänge* des Funktionsbausteins **TCP_Send** im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	DWORD
LEN	INT
REQ	BOOL

Tabelle 25: Eingangs-Systemvariablen

Die Ausgänge des Funktionsbausteins **Senden** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Eingänge* des Funktionsbausteins **TCP_Send** im Anwenderprogramm verbunden wurden.

Ausgänge	Typ
Ack	BOOL
Busy	BOOL
Done	BOOL
ERROR	BOOL
STATUS	DWORD

Tabelle 26: Ausgangs-Systemvariablen

Daten	Beschreibung
Sende-Daten	Im Register <i>Prozessvariablen</i> können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen allerdings identisch mit den Offsets und den Typen der Variablen des Kommunikationspartners sein.

Tabelle 27: Sende-Daten

Für die Bedienung des Funktionsbaustein TCP_Send sind die folgenden Schritte erforderlich:

-
- i** Die zu sendenden Variablen müssen im Register *Prozessvariablen* des Dialogs *Senden* angelegt werden. Die Offsets und Typen der Variablen müssen identisch mit den Offsets und den Typen der Variablen des Kommunikationspartners sein.
-
1. Im Anwenderprogramm die TCP-Verbindungs ID am Eingang *A_Id* setzen.
 2. Im Anwenderprogramm die Länge der zu sendenden Variablen am Eingang *A_Len* in Byte setzen.
 3. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen.
-

- i** Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.
-

Der Ausgang *A_Busy* geht solange auf TRUE, bis die Variablen gesendet wurden. Danach gehen die Ausgänge *A_Busy* auf FALSE und *A_Done* auf TRUE.

Konnte der Sendevorgang nicht erfolgreich ausgeführt werden, geht der Ausgang *A_Error* auf TRUE und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

3.7.3 TCP_Receive

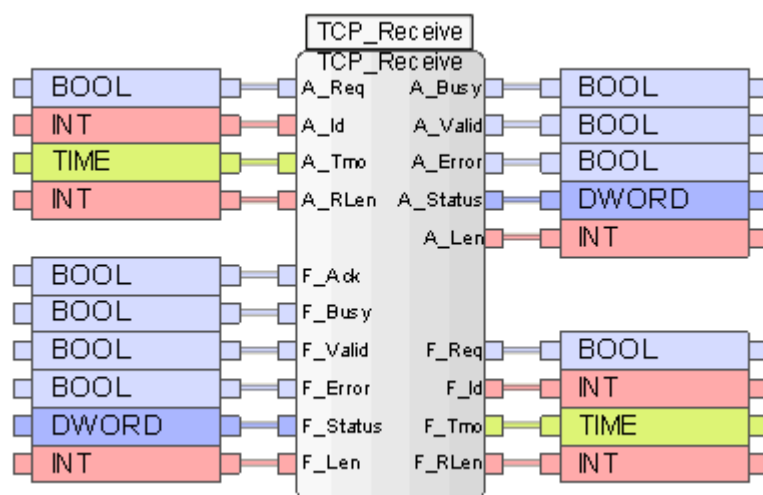


Bild 13: Funktionsbaustein TCP_Receive

Mit dem Funktionsbaustein **TCP_Receive** können definierte Variablen von einem Kommunikationspartner empfangen werden.

Im Kommunikationspartner muss ein Funktionsbaustein z. B. TCP_Send mit den gleichen Variablen und Offsets konfiguriert werden.

i

Zur Konfiguration den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm ziehen, siehe auch Kapitel 4.3.

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix A:

Über diese Ein- und Ausgänge kann der Funktionsbaustein mit Hilfe des Anwenderprogramms gesteuert und ausgewertet werden. Das Präfix "A" steht für "Application".

A-Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet den Funktionsbaustein.	BOOL
A_Id	Identifikationsnummer der konfigurierten TCP-Verbindung zu dem Kommunikationspartner, von welchem die Daten empfangen werden sollen.	INT
A_Tmo	Empfangs-Timeout. Wenn innerhalb dieser Zeit keine Daten empfangen wurden, wird der Baustein mit einer Fehlermeldung beendet. Wird der Eingang A_Tmo nicht belegt, oder null angelegt, ist der Timeout deaktiviert.	TIME
A_RLen	A_RLen ist die erwartete Länge der zu empfangenden Variablen in Bytes. A_RLen muss größer als null sein und darf nicht innerhalb einer Variablen enden.	INT

Tabelle 28: A-Eingänge Funktionsbaustein TCP_Receive

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Empfang der Daten ist noch nicht beendet.	BOOL
A_Valid	TRUE: Der Empfang der Daten wurde fehlerfrei beendet.	BOOL
A_Error	TRUE: Ein Fehler ist aufgetreten. FALSE: Kein Fehler.	BOOL
A_Status	Am Ausgang <i>A_Status</i> werden Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD
A_Len	Anzahl der empfangenen Bytes.	INT

Tabelle 29: A-Ausgänge Funktionsbaustein TCP_Receive

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F:

Diese Ein- und Ausgänge des Funktionsbausteins stellen die Verbindung zum Funktionsbaustein **Empfangen** im Strukturbaum her. Das Präfix "F" steht für "Field".

i

Die Verbindung des Funktionsbausteins **Empfangen** im Strukturbaum (im Ordner Funktionsbausteine) mit dem Funktionsbaustein **TCP_Receive** (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor vom Anwender im Variableneditor erstellt werden.

Die *F-Eingänge* des Funktionsbausteins **TCP_Receive** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Ausgänge des Funktionsbausteins **Empfangen** im Strukturbaum verbunden werden.

F-Eingänge	Typ
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Tabelle 30: A-Eingänge Funktionsbaustein TCP_Receive

Die *F-Ausgänge* des Funktionsbausteins **TCP_Receive** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Eingänge des Funktionsbausteins **Empfangen** im Strukturbaum verbunden werden.

F-Ausgänge	Typ
F_Req	BOOL
F_Id	DWORD
F_Tmo	INT
F_RLen	INT

Tabelle 31: F-Ausgänge Funktionsbaustein TCP_Receive

Zugehörigen Funktionsbaustein Empfangen im Strukturbaum erstellen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Send Receive over TCP, Funktionsbausteine, Neu** öffnen.
2. Funktionsbaustein **Empfangen** wählen.
3. Rechtsklick auf Funktionsbaustein **Empfangen** und **Edit** wählen
☒ Variablenzuweisung zum Funktionsbaustein wird geöffnet.

Die Eingänge des Funktionsbausteins **Empfangen** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Ausgänge* des Funktionsbausteins **TCP_Receive** im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	INT
REQ	BOOL
RLEN	INT
TIMEOUT	TIME

Tabelle 32: Eingangs-Systemvariablen

Die Ausgänge des Funktionsbausteins **Empfangen** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Eingänge* des Funktionsbausteins **TCP_Receive** im Anwenderprogramm verbunden wurden.

Ausgänge	Typ
Ack	BOOL
Busy	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabelle 33: Ausgangs-Systemvariablen

Daten	Beschreibung
Empfangsvariablen	Im Register <i>Prozessvariablen</i> können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen allerdings identisch mit den Offsets und den Typen der Variablen des Kommunikationspartners sein.

Tabelle 34: Empfangsvariablen

Für die Bedienung des Funktionsbaustein TCP_Receive sind die folgenden Schritte erforderlich:

i

Die Empfangsvariablen müssen im Register *Prozessvariablen* des Dialogs *Empfangen* angelegt werden. Die Offsets und Typen der Empfangsvariablen müssen identisch mit den Offsets und den Typen der Sendeveriablen des Kommunikationspartners sein.

1. Im Anwenderprogramm die Identifikationsnummer der TCP-Verbindung am Eingang *A_Id* setzen.
 2. Im Anwenderprogramm die Empfangs-Timeout am Eingang *A_Tmo* setzen.
 3. Im Anwenderprogramm die erwartete Länge der zu empfangenden Variablen am Eingang *A_RLen* setzen.
 4. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen.
-

i

Der Funktionsbaustein startet mit einem positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis die Variablen empfangen wurden, oder der Empfangs-Timeout abgelaufen ist. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Valid* oder *A_Error* auf TRUE.

Ist der Empfang der Variablen fehlerfrei, wechselt der Ausgang *A_Valid* auf TRUE. Die Variablen, die im Register *Daten* definiert wurden, können ausgewertet werden.

Ist der Empfang der Variablen nicht fehlerfrei, wechselt der Ausgang *A_Error* auf TRUE, und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

3.7.4 TCP_ReceiveLine

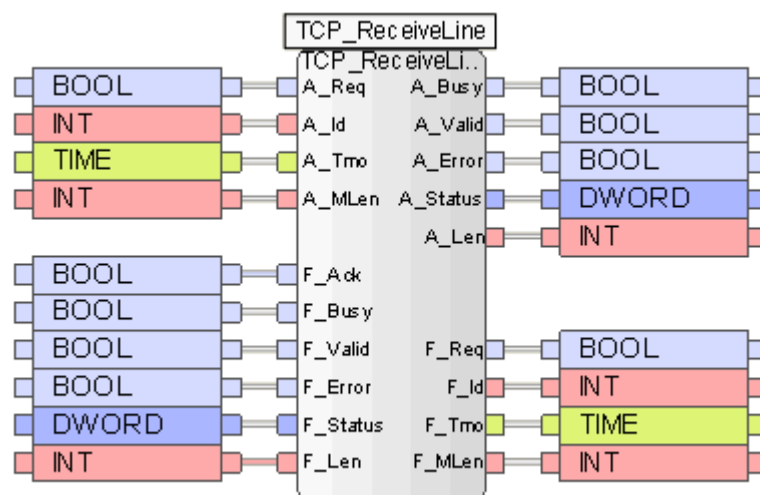


Bild 14: Funktionsbaustein TCP_ReceiveLine

Der Funktionsbaustein **TCP_ReceiveLine** dient zum Empfang einer ASCII Zeichenkette inklusive LineFeed (16#0A) eines Kommunikationspartners.

i

Zur Konfiguration den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm ziehen, siehe auch Kapitel 4.3.

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix A:

Über diese Ein- und Ausgänge kann der Funktionsbaustein mit Hilfe des Anwenderprogramms gesteuert und ausgewertet werden. Das Präfix "A" steht für "Application".

A-Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet den Baustein und setzt die Verbindung empfangsbereit.	BOOL
A_Id	Identifikationsnummer der konfigurierten TCP-Verbindung zu dem Kommunikationspartner, von welchem die Daten empfangen werden sollen.	INT
A_Tmo	Empfangs-Timeout Wenn innerhalb dieser Zeit keine Daten empfangen wurden, wird der Baustein mit einer Fehlermeldung beendet. Wird der Eingang offen gelassen, oder Null angelegt, ist der Timeout ausgeschaltet.	TIME
A_MLen	<i>A_Mlen</i> ist die maximale Länge einer zu empfangenden Zeile in Bytes. Die Empfangsvariablen müssen im Register <i>Daten</i> im Com-Funktionsbaustein angelegt werden. Übertragene Bytes = Min (<i>A_MLen</i> , Zeilenlänge, Länge des Datenbereichs).	INT

Tabelle 35: A-Eingänge Funktionsbaustein TCP_ReceiveLine

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Empfang der Daten ist noch nicht beendet.	BOOL
A_Valid	TRUE: Der Empfang der Daten wurde fehlerfrei beendet.	BOOL
A_Error	TRUE: Ein Fehler ist aufgetreten. FALSE: Kein Fehler.	BOOL
A_Status	Am Ausgang A_Status wird Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD
A_Len	Anzahl der empfangenen Bytes.	INT

Tabelle 36: A-Ausgänge Funktionsbaustein TCP_ReceiveLine

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F:

Diese Ein- und Ausgänge des Funktionsbausteins stellen die Verbindung zum Funktionsbaustein **Zeilenweises Empfangen** im Strukturbaum her. Das Präfix "F" steht für "Field".

-
- i** Die Verbindung des Funktionsbausteins **Zeilenweises Empfangen** im Strukturbaum (im Ordner Funktionsbausteine) mit dem Funktionsbaustein **TCP_ReceiveLine** (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor vom Anwender im Variableneditor erstellt werden.
-

Die *F-Eingänge* des Funktionsbausteins **TCP_ReceiveLine** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Ausgänge des Funktionsbausteins **Zeilenweises Empfangen** im Strukturbaum verbunden werden.

F-Eingänge	Typ
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Tabelle 37: F-Eingänge Funktionsbaustein TCP_ReceiveLine

Die *F-Ausgänge* des Funktionsbausteins **TCP_ReceiveLine** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Eingänge des Funktionsbausteins **Zeilenweises Empfangen** im Strukturbaum verbunden werden.

F-Ausgänge	Typ
A_Req	BOOL
A_Id	INT
A_Tmo	TIME
A_MLen	INT

Tabelle 38: F-Ausgänge Funktionsbaustein TCP_ReceiveLine

Zugehörigen Funktionsbaustein Zeilenweises Empfangen im Strukturbaum erstellen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Send Receive over TCP, Funktionsbausteine, Neu** öffnen.
2. Funktionsbaustein **Zeilenweises Empfangen** wählen.
3. Rechtsklick auf Funktionsbaustein **Zeilenweises Empfangen** und **Edit** wählen
☒ Variablenzuweisung zum Funktionsbaustein wird geöffnet.

Die Eingänge des Funktionsbausteins **Zeilenweises Empfangen** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Ausgänge* des Funktionsbausteins **TCP_ReceiveLine** im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	INT
MLEN	INT
REQ	BOOL
TIMEOUT	TIME

Tabelle 39: Eingangs-Systemvariablen

Die Ausgänge des Funktionsbausteins **Zeilenweises Empfangen** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Eingänge* des Funktionsbausteins **TCP_ReceiveLine** im Anwenderprogramm verbunden wurden.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabelle 40: Ausgangs-Systemvariablen

Daten	Beschreibung
Empfangsvariablen	Im Register <i>Prozessvariablen</i> sollten sinnvollerweise nur Variablen vom Typ BYTE angelegt werden. Die Offsets der Variablen müssen identisch mit den Offsets der Variablen des Kommunikationspartners sein.

Tabelle 41: Empfangsvariablen

Für die Bedienung des Funktionsbaustein TCP_ReceiveLine sind die folgenden Schritte erforderlich:

i

Die Empfangsvariablen vom Typ Byte müssen im Register *Prozessvariablen* des Dialogs *Zeilenweises Empfangen* angelegt werden. Die Offsets der Empfangsvariablen müssen identisch mit den Offsets der Sendeveriablen des Kommunikationspartners sein.

1. Im Anwenderprogramm die Identifikationsnummer der TCP-Verbindung am Eingang *A_Id* setzen.
 2. Im Anwenderprogramm die Empfangs-Timeout am Eingang *A_Tmo* setzen.
 3. Im Anwenderprogramm die maximale Länge der zu empfangenden Zeile am Eingang *A_MLen* setzen.
-

i

A_MLen muss größer als Null sein und bestimmt die Größe des Empfangspuffers in Byte. Wenn der Empfangspuffer gefüllt ist, und noch kein Zeilenende aufgetreten ist, wird der Lesevorgang ohne Fehlermeldung beendet.

Am Ausgang *A_Len* wird die Anzahl der empfangenen Bytes zur Verfügung gestellt:
Empfangene Bytes = Min (*A_MLen*, Zeilenlänge, Länge des Datenbereichs).

4. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen.
-

i

Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis der Empfangspuffer voll ist oder das Zeilenende *LineFeed* empfangen wurde oder der Empfangs-Timeout abgelaufen ist. Danach gehen die Ausgänge *A_Busy* auf FALSE und *A_Valid* oder *A_Error* auf TRUE.

Ist der Empfang der Zeile fehlerfrei, wechselt der Ausgang *A_Valid* auf TRUE. Die Variablen, die im Register Daten definiert wurden, können ausgewertet werden.

Ist der Empfang der Zeile nicht fehlerfrei, wechselt der Ausgang *A_Error* auf TRUE, und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

3.7.5 TCP_ReceiveVar

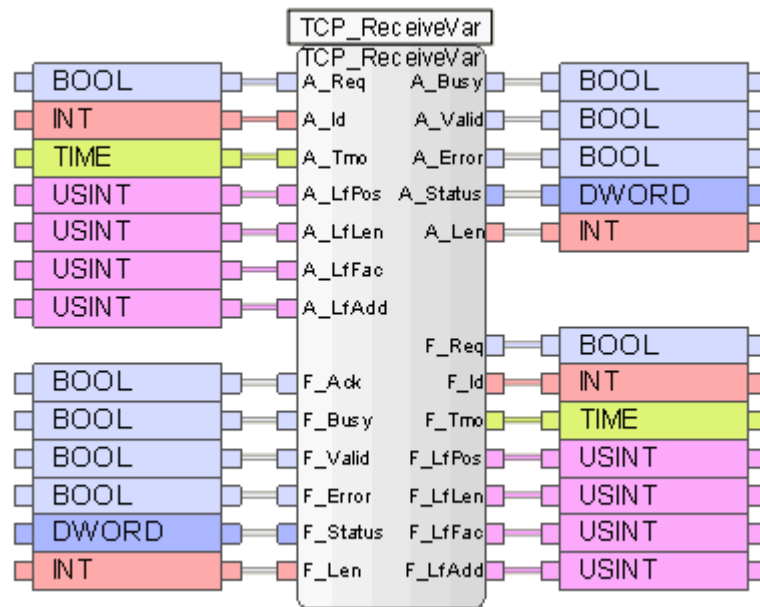


Bild 15: Funktionsbaustein TCP_ReceiveVar

Mit dem Funktionsbaustein **TCP_ReceiveVar** können Datenpakete variabler Länge, die mit einem Längenfeld ausgestattet sind, ausgewertet werden.

i

Zur Konfiguration den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm ziehen, siehe auch Kapitel 4.3.

Funktionsbeschreibung

Die empfangenen Datenpakete müssen den im Bild unten dargestellten Aufbau besitzen (z. B. Modbus-Protokoll). Eine Anpassung an ein beliebiges Protokoll-Format erfolgt über die Einstellung der Eingabeparameter **A_LfPos**, **A_LfLen**, **A_LfFac**, **A_LfLen**.

Das empfangene Datenpaket besteht aus einem Kopf- und einem Nutzdatenbereich. Der Kopfbereich enthält Daten wie Teilnehmer-Adresse, Telegrammfunktion, Längenfeld usw., die für die Kommunikationsverbindung erforderlich sind. Um den Nutzdatenbereich auszuwerten muss der Kopfbereich abgetrennt und das Längenfeld ausgelesen werden.

Die Größe des Kopfbereichs wird im Parameter **A_LfAdd** eingetragen.

Die Länge des Nutzdatenbereichs muss aus dem Längenfeld des aktuell gelesenen Datenpakets ausgelesen werden. Die Position des Längenfeldes wird im Parameter **A_LfPos** eingetragen. Die Größe des Längenfeldes wird in **LfLen** in Byte eingetragen. Falls die Länge nicht als Byte angegeben ist, muss der Umrechnungsfaktor hierfür in **A_LfFac** eingetragen werden (z. B. 2 für Word oder 4 für Double Word).

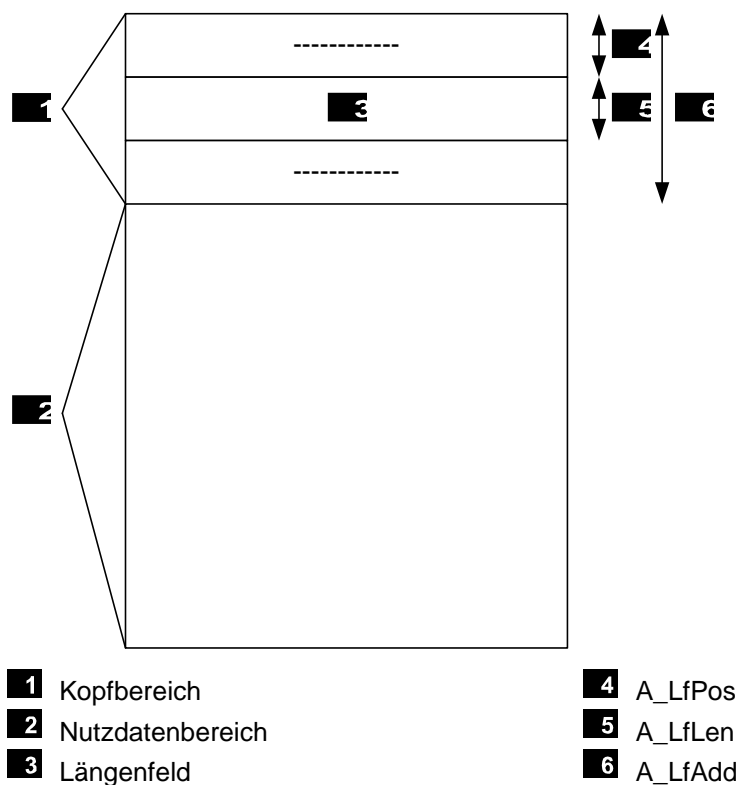


Bild 16: Aufbau der Datenpakete

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix A

Über diese Ein- und Ausgänge kann der Funktionsbaustein mit Hilfe des Anwenderprogramms gesteuert und ausgewertet werden. Das Präfix "A" steht für "Application".

A-Eingänge	Beschreibung	Typ
A_Req	Mit der positiven Flanke wird der CPU-Funktionsbaustein gestartet.	BOOL
A_Id	Identifikationsnummer <i>ID</i> der konfigurierten TCP-Verbindung zu einem Kommunikationspartner, von dem das Datenpaket empfangen werden soll.	DWORD
A_Tmo	Empfangs-Timeout Wenn innerhalb dieser Zeit keine Daten empfangen wurden, wird der Baustein mit einer Fehlermeldung beendet. Wird der Eingang offen gelassen, oder null angelegt, ist der Timeout ausgeschaltet.	INT
A_LfPos	Startposition des Längenfeldes im Datenpaket; die Nummerierung beginnt mit null (gemessen in Bytes).	USINT
A_LfLen	Größe des Längenfeldes <i>A_LfLen</i> in Bytes. Erlaubt sind 1, 2 oder 4 Bytes.	USINT
A_LfFac	Umrechnungsfaktor in Bytes, falls der Eintrag im Längenfeld nicht in Bytes ist. Wird der Eingang offen gelassen, oder mit Null belegt, wird 1 als Standardwert genommen.	USINT
A_LfAdd	Größe des Kopffeldes in Bytes.	USINT

Tabelle 42: A-Eingänge Funktionsbaustein TCP_ReceiveVar

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Empfang der Daten ist noch nicht beendet.	BOOL
A_Valid	TRUE: Der Empfang der Daten wurde fehlerfrei beendet.	BOOL
A_Error	TRUE: Beim Lesen trat ein Fehler auf. FALSE: Kein Fehler.	BOOL
A_Status	Am Ausgang <i>A_Status</i> wird Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD
A_Len	Anzahl der empfangenen Bytes.	INT

Tabelle 43: A-Ausgänge Funktionsbaustein TCP_ReceiveVar

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F:

Diese Ein- und Ausgänge des Funktionsbausteins stellen die Verbindung zum Funktionsbaustein **Variabel Empfangen** im Strukturbaum her. Das Präfix "F" steht für "Field".

i

Die Verbindung des Funktionsbausteins **Variabel Empfangen** im Strukturbaum (im Ordner Funktionsbausteine) mit dem Funktionsbaustein **TCP_ReceiveVar** (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor vom Anwender im Variableneditor erstellt werden.

Die *F-Eingänge* des Funktionsbausteins **TCP_ReceiveVar** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Ausgänge des Funktionsbausteins **Variabel Empfangen** im Strukturbaum verbunden werden.

F-Eingänge	Typ
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
A_Status	DWORD
A_Len	INT

Tabelle 44: F-Eingänge Funktionsbaustein TCP_ReceiveVar

Die *F-Ausgänge* des Funktionsbausteins **TCP_ReceiveVar** im Anwenderprogramm mit den gleichen Variablen verbinden, mit denen später auch die Eingänge des Funktionsbausteins **Variabel Empfangen** im Strukturbaum verbunden werden.

F-Ausgänge	Typ
F_Req	BOOL
F_Id	INT
F_Tmo	TIME
F_LfPos	USINT
A_LfLen	USINT
A_LfFac	USINT
A_LfAdd	USINT

Tabelle 45: F-Ausgänge Funktionsbaustein TCP_ReceiveVar

Zugehörigen Funktionsbaustein Variabel Empfangen im Strukturbaum erstellen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Send Receive over TCP, Funktionsbausteine, Neu** öffnen.
2. Funktionsbaustein **Variabel Empfangen** wählen.
3. Rechtsklick auf Funktionsbaustein **Variabel Empfangen** und **Edit** wählen
☒ Variablenzuweisung zum Funktionsbaustein wird geöffnet.

Die Eingänge des Funktionsbausteins **Variabel Empfangen** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Ausgänge* des Funktionsbausteins **TCP_ReceiveVar** im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	INT
Lf Add	USINT
Lf Fac	USINT
Lf Len	USINT
Lf Pos	USINT
REQ	BOOL
TIMEOUT	TIME

Tabelle 46: Eingangs-Systemvariablen

Die Ausgänge des Funktionsbausteins **Variabel Empfangen** im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die *F-Eingänge* des Funktionsbausteins **TCP_ReceiveVar** im Anwenderprogramm verbunden wurden.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabelle 47: Ausgangs-Systemvariablen

Daten	Beschreibung
Empfangsvariablen	Im Register <i>Prozessvariablen</i> können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen allerdings identisch mit den Offsets und den Typen der Variablen des Kommunikationspartners sein.

Tabelle 48: Empfangsvariablen

Für die Bedienung des Funktionsbaustein TCP_ReceiveVar sind die folgenden Schritte erforderlich:

i

Die Empfangsvariablen müssen im Register *Prozessvariablen* des Dialogs *Variabel Empfangen* angelegt werden. Die Offsets und Typen der Empfangsvariablen müssen identisch mit den Offsets und den Typen der Sendevariablen des Kommunikationspartners sein.

-
1. Im Anwenderprogramm die Identifikationsnummer der TCP-Verbindung am Eingang *A_Id* setzen.
 2. Im Anwenderprogramm die Empfangs-Timeout am Eingang *A_Tmo* setzen.
 3. Im Anwenderprogramm die Parameter *A_LfPos*, *A_LfLen*, *A_LfFac* und *A_LfAdd* setzen.
 4. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen.

i

Der Funktionsbaustein startet mit einem positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis die Variablen empfangen wurden, oder der Empfangs-Timeout abgelaufen ist. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Valid* oder *A_Error* auf TRUE.

Ist der Empfang der Variablen fehlerfrei, wechselt der Ausgang *A_Valid* auf TRUE. Die Variablen, die im Register Daten definiert wurden, können ausgewertet werden. Der Ausgang *A_Len* enthält die Anzahl der Bytes, die tatsächlich ausgelesen wurden.

Ist der Empfang der Variablen nicht fehlerfrei, wechselt der Ausgang *A_Error* auf TRUE, und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

3.8 Control-Panel (Send/Receive over TCP)

Im Control-Panel kann der Anwender die Einstellungen des Send/Receive Protokolls überprüfen und steuern. Zudem werden aktuelle Statusinformationen (z. B. gestörte Verbindungen usw.) des Send/Receive Protokolls angezeigt.

Öffnen des Control Panels zur Überwachung des Send/Receive Protokolls:

1. Im Strukturbaum **Ressource** selektieren.
2. Rechtsklick und im Kontextmenü **Online** wählen.
3. Im **System-Login**, Zugangsdaten eingeben um das Control Panel der Ressource zu öffnen.
4. Im Strukturbaum des Control Panels **Send/Receive Protokoll** wählen.

3.8.1 Anzeigefeld allgemeine Parameter

In dem Anzeigefeld werden die folgenden Werte des Send/Receive Protokolls angezeigt.

Element	Beschreibung
Name	TCP SR Protokoll.
Projektiertes μ P-Budget [%]	Anzeige des projektierten maximalen μ P-Budget des COM Moduls, welches bei der Abarbeitung des Protokolls produziert werden darf.
Aktuelles μ P-Budget [%]	Anzeige des aktuellen μ P-Budget des COM Moduls, welches bei der Abarbeitung des Protokolls momentan produziert wird.
Ungestörte Verbindungen	Anzahl ungestörte Verbindungen.
Gestörte Verbindungen	Anzahl gestörte Verbindungen.

Tabelle 49: Anzeigefeld S/R-Protokoll

3.8.2 Anzeigefeld TCP Verbindungen

In dem Anzeigefeld werden die folgenden Werte der selektierten TCP Verbindungen angezeigt.

Element	Beschreibung
Name	TCP Verbindung.
Partner Timeout	Ja: Partner Request Timeout abgelaufen. Nein: Partner Request Timeout nicht abgelaufen.
Verbindungszustand	Aktueller Verbindungszustand dieser Verbindung: 0x00: Verbindung OK. 0x01: Verbindung geschlossen. 0x02: Server wartet auf Verbindungsaufnahme. 0x04: Client versucht Verbindungsaufbau. 0x08: Verbindung ist blockiert.
Peer-Adresse	IP-Adresse des Kommunikationspartners.
Peer-Port	Port des Kommunikationspartners.
Eigener Port	Port dieser Steuerung.
Watchdog-Zeit [ms]	Ist die aktuelle Partner Request Timeout, innerhalb der nach Datenversand mindestens eine Datensendung vom Kommunikationspartner empfangen wurde.
Fehlercode	Fehlercode, siehe Kapitel 3.8.3.
Zeitstempel Fehlercode [ms]	Zeitstempel des letzten gemeldeten Fehlers. Wertebereich: Sekunden seit 1.1.1970 in Millisekunden.
Empfangene Bytes [Byte]	Anzahl empfangener Bytes dieser TCP Verbindung.
Gesendete Bytes [Byte]	Anzahl gesendeter Bytes dieser TCP Verbindung.

Tabelle 50: Anzeigefeld Modbus Slaves

3.8.3 Fehlercode der TCP-Verbindung

Die Fehlercodes können aus der Variablen *Errorcode* gelesen werden.

Pro konfigurierter Verbindung: Der Verbindungsstatus setzt sich zusammen aus dem Verbindungszustand und dem Fehlercode der letzten Operation.

Fehlercode Dezimal	Fehlercode Hexadezimal	Beschreibung
0	16#00	OK.
4	16#04	Unterbrochener Systemaufruf.
5	16#05	I/O-Fehler.
6	16#06	Unbekanntes Gerät.
9	16#09	Ungültiger Socket-Deskriptor.
12	16#0C	Kein Speicher mehr vorhanden.
13	16#0D	Zugriff verweigert.
14	16#0E	Ungültige Adresse.
16	16#10	Gerät ist beschäftigt.
22	16#16	Ungültiger Wert (z. B. im Längenfeld).
23	16#17	Deskriptortabelle ist voll.
32	16#20	Verbindung unterbrochen.
35	16#23	Operation ist blockiert.
36	16#24	Operation jetzt in Arbeit.
37	16#25	Operation bereits in Arbeit.
38	16#27	Zieladresse erforderlich.
39	16#28	Message zu lang.
40	16#29	Falscher Protokolltyp für Socket.
42	16#2A	Protokoll nicht verfügbar.
43	16#2B	Protokoll nicht unterstützt.
45	16#2D	Operation auf Socket nicht unterstützt.
47	16#2F	Adresse von Protokoll nicht unterstützt.
48	16#30	Adresse ist bereits in Verwendung.
49	16#31	Adresse kann nicht zugewiesen werden.
50	16#32	Netzwerk läuft nicht.
53	16#35	Software hat Verbindung abgebrochen.
54	16#36	Verbindung wurde vom Partner zurückgesetzt.
55	16#37	Kein Pufferspeicher mehr verfügbar.
56	16#38	Socket ist bereits verbunden.
57	16#39	Socket ist nicht verbunden.
58	16#3A	Socket ist geschlossen.
60	16#3C	Zeit für Operation abgelaufen.
61	16#3D	Verbindung abgewiesen (durch Partner).
65	16#41	Kein Routingeintrag zum Partner vorhanden.
78	16#4E	Funktion nicht vorhanden.
254	16#FE	Timeout aufgetreten.
255	16#FF	Verbindung vom Partner geschlossen.

Tabelle 51: Fehlercodes der TCP-Verbindung

3.8.4 Zusätzliche Fehlercodetabelle der Funktionsbausteine

Die Fehlercodes der Funktionsbausteine werden nur an A_Status der S/R-TCP-Funktionsbausteinen ausgegeben.

Fehlercode Dezimal	Fehlercode Hexadezimal	Beschreibung
129	16#81	Es gibt keine Verbindung mit diesem Identifier.
130	16#82	Länge ist Null oder zu groß.
131	16#83	Auf dieser Verbindung sind nur zyklische Daten erlaubt.
132	16#84	Falscher Zustand.
133	16#85	Der Timeout-Wert ist zu groß.
134	16#86	Interner Programmfehler.
135	16#87	Konfigurationsfehler.
136	16#88	Übertragene Daten passen nicht auf den Datenbereich.
137	16#89	Funktionsbaustein gestoppt.
138	16#8A	Timeout aufgetreten oder Senden blockiert.
139	16#8B	Ein Funktionsbaustein dieser Art ist bereits auf dieser Verbindung aktiv.

Tabelle 52: Zusätzliche Fehlercodes

3.8.5 Verbindungszustand

Der Verbindungszustand wird über folgende Fehlercodes angezeigt:

Fehlercode Hexadezimal	Beschreibung
16#00	Verbindung OK.
16#01	Verbindung geschlossen.
16#02	Server wartet auf Verbindungsaufnahme.
16#04	Client versucht Verbindungsaufbau.
16#08	Verbindung ist blockiert.

Tabelle 53: Verbindungszustand

3.8.6 Partner Verbindungszustand

Der Partner Verbindungszustand wird wie folgt angezeigt:

Protokollzustand Dezimal	Beschreibung
0	Keine Verbindung.
1	Verbindung OK.

Tabelle 54: Partner Verbindungszustand

4 Allgemein

In diesem Kapitel sind Parameter gesammelt, die für alle Kommunikationsprotokolle relevant sind.

4.1 Maximale Kommunikationszeitscheibe

Die maximale Kommunikationszeitscheibe ist die zugeteilte Zeit in Millisekunden (ms) pro CPU-Zyklus, innerhalb der das Prozessormodul die Kommunikationsaufgaben abarbeitet. Wenn die Protokollverarbeitung innerhalb der Dauer einer Kommunikationszeitscheibe nicht beendet werden konnte, führt die CPU dennoch die sicherheitsrelevanten Überwachungen für alle Protokolle in einem CPU-Zyklus aus.

i

Wenn nicht alle in einem CPU-Zyklus anstehenden Kommunikationsaufgaben ausgeführt werden können, erfolgt die komplette Übertragung der Kommunikationsdaten über mehrere CPU-Zyklen. Die Anzahl der Kommunikationszeitscheiben ist dann größer 1.

Für die Berechnungen der zulässigen maximalen Reaktionszeiten gilt die Bedingung, dass die Anzahl der Kommunikationszeitscheiben genau 1 ist.

4.1.1 Ermitteln der maximalen Dauer der Kommunikationszeitscheibe

Für eine erste Abschätzung der maximalen Dauer der Kommunikationszeitscheibe müssen die folgenden Zeiten aufsummiert und das Ergebnis in den Systemparameter *Max. Kom.-Zeitscheibe [ms]* in den Eigenschaften der Ressource eingetragen werden:

- Pro COM-Modul 3 ms.
- Pro redundante **safeethernet** Verbindung 1 ms.
- Pro nicht redundante **safeethernet** Verbindung 0,5 ms.
- Pro KByte Nutzdaten bei nichtsicheren Protokollen (z. B. Modbus) 1 ms.

HIMA empfiehlt, den abgeschätzten Wert *Max. Kom.-Zeitscheibe [ms]* mit dem im Control Panel angezeigten Wert zu vergleichen und gegebenenfalls in den Eigenschaften der Ressource zu korrigieren. Dies kann z. B. in einem FAT (Factory Acceptance Test) oder SAT (Site Acceptance Test) durchgeführt werden.

Ermitteln der tatsächlichen Dauer der maximalen Kommunikationszeitscheibe

1. Das HIMA System unter voller Last betreiben (FAT, SAT):
Alle Kommunikationsprotokolle sind in Betrieb (**safeethernet** und Standardprotokolle).
2. Das **Control Panel** öffnen und im Strukturbaum das Verzeichnis **Kom.-Zeitscheibe** wählen.
3. Anzeige *Maximale Kom.-Zeitscheibe Dauer pro Zyklus [ms]* auszulesen.
4. Anzeige *Maximale Anzahl benötigter Kom.-Zeitscheibe Zyklen* auszulesen.

Die Dauer der Kommunikationszeitscheibe ist so hoch einzustellen, dass der CPU-Zyklus die vom Prozess vorgegebene Watchdog-Zeit nicht überschreiten kann, wenn er die eingestellte Kommunikationszeitscheibe ausnutzt.

4.2 Lastbegrenzung

Für jedes Kommunikationsprotokoll kann ein Rechenzeitbudget in % (*μP-Budget*) vorgegeben werden. So kann die verfügbare Rechenzeit zwischen den konfigurierten Protokollen verteilt werden. Die Summe der Rechenzeitbudgets aller parametrisierten Kommunikationsprotokolle eines CPU- oder COM-Moduls darf nicht größer als 100 % sein.

Die festgelegten Rechenzeitbudgets der einzelnen Kommunikationsprotokolle werden überwacht. Hat ein Kommunikationsprotokoll sein Rechenzeitbudget erreicht oder überschritten und es steht keine zusätzliche Rechenzeit als Reserve zur Verfügung, so wird das Kommunikationsprotokoll nicht komplett abgearbeitet.

Wenn noch genügend zusätzliche Rechenzeit vorhanden ist, wird diese verwendet, um ein Kommunikationsprotokoll, das sein Rechenzeitbudget erreicht oder überschritten hat noch abzuarbeiten. Dadurch kann es vorkommen, dass ein Kommunikationsprotokoll tatsächlich ein höheres Rechenzeitbudget verwendet als ihm zugeteilt wurde.

Eventuell werden über 100 % Rechenzeitbudget online angezeigt. Dies ist kein Fehler, das Rechenzeitbudget über 100 % ist die zusätzlich verwendete Rechenzeit.

i

Das zusätzliche Rechenzeitbudget ist keinesfalls eine Zusicherung für ein bestimmtes Kommunikationsprotokoll und kann jederzeit vom System zurückgenommen werden.

4.3 Konfiguration der Funktionsbausteine

Die Feldbus Protokolle und die zugehörigen Funktionsbausteine laufen auf dem COM-Modul der HIMA Steuerung. Daher müssen diese Funktionsbausteine im SILworX Strukturbaum unter **Konfiguration, Ressource, Protokolle...** angelegt werden.

Um diese Funktionsbausteine auf dem COM-Modul zu steuern, können im Anwenderprogramm von SILworX Funktionsbausteine angelegt werden (siehe Kapitel 4.3.1), die wie Standard-Funktionsbausteine verwendet werden können.

Die Verbindung der Funktionsbausteine im Anwenderprogramm von SILworX mit den entsprechenden Funktionsbausteinen im Strukturbaum von SILworX erfolgt über gemeinsame Variablen. Diese müssen zuvor vom Anwender im Variableneditor erstellt werden.

4.3.1 Beschaffung der Funktionsbausteinbibliotheken

Die Funktionsbausteinbibliotheken für PROFIBUS DP und TCP Send/Receive muss über die Funktion *Wiederherstellen...* (Kontextmenü des Projekts) dem Projekt hinzugefügt werden.

Die Funktionsbausteinbibliothek ist auf Anfrage über den HIMA Support erhältlich, siehe Kapitel 1.4.

4.3.2 Konfiguration der Funktionsbausteine im Anwenderprogramm

Die benötigten Funktionsbausteine können per Drag&Drop in das Anwenderprogramm kopiert werden. Die Eingänge und Ausgänge nach der Beschreibung des jeweiligen Funktionsbausteins konfigurieren.

Oberer Teil des Funktionsbausteins

Der obere Teil des Funktionsbausteins entspricht der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

Hier werden die Variablen verbunden, die im Anwenderprogramm verwendet werden. Das Präfix "A" steht für "Applikation".

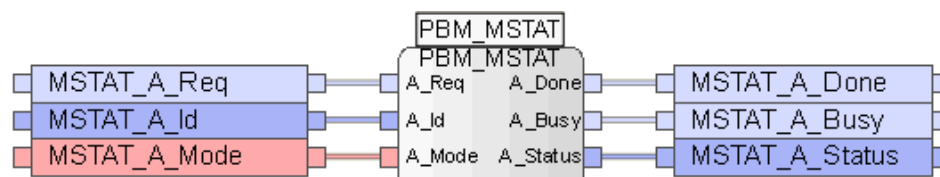


Bild 17: Funktionsbaustein PNM_MSTST (oberer Teil)

Unterer Teil des Funktionsbausteins

Der untere Teil des Funktionsbausteins stellt die Verbindung zum Funktionsbaustein (im Strukturbaum von SILworX) dar.

Hier werden die Variablen verbunden, die mit dem Funktionsbaustein im Strukturbaum von SILworX verbunden werden müssen. Das Präfix "F" steht für "Field".

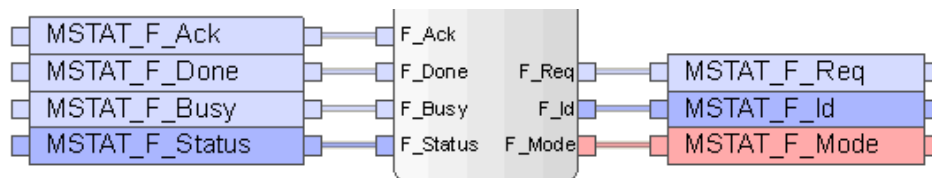


Bild 18: Funktionsbaustein PNM_MSTST (unterer Teil)

4.3.3 Konfiguration der Funktionsbausteine im Strukturbaum von SILworX

Den Funktionsbaustein im Strukturbaum von SILworX:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle**, z. B. **PROFIBUS Master** wählen.
2. Rechtsklick auf **Funktionsbausteine** und **Neu** wählen.
3. Den passenden Funktionsbaustein (im Strukturbaum von SILworX) auswählen.

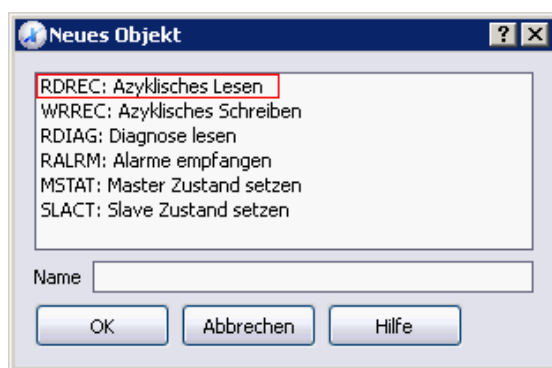


Bild 19: Auswahl Funktionsbausteine

Die Eingänge des Funktionsbausteins (Häkchen in Spalte Eingangsvariable) müssen mit den gleichen Variablen verbunden werden, die mit den *F-Ausgängen* des Funktionsbausteins im Anwenderprogramm verbunden sind.

Die Ausgänge des Funktionsbausteins (kein Häkchen in Spalte Eingangsvariable) müssen mit den gleichen Variablen verbunden werden, die mit den *F-Eingängen* des Funktionsbausteins im Anwenderprogramm verbunden sind.

Systemvariablen					
F	Name	Datentyp	Transfer-Operation	Eingangsvariable	Globale Variable
1	ACK	BOOL	5	<input checked="" type="checkbox"/>	MSTAT_F_Ack
2	BUSY	BOOL	5	<input checked="" type="checkbox"/>	MSTAT_F_Busy
3	DONE	BOOL	5	<input checked="" type="checkbox"/>	MSTAT_F_Done
4	REQ	BOOL	5	<input type="checkbox"/>	MSTAT_F_Req
5	M_ID	DWORD	5	<input type="checkbox"/>	MSTAT_F_Id
6	STATUS	DWORD	5	<input checked="" type="checkbox"/>	MSTAT_F_Status
7	MODE	INT	5	<input type="checkbox"/>	MSTAT_F_Mode

Bild 20: Systemvariablen des Funktionsbausteins MSTAT

Anhang

Glossar

Begriff	Beschreibung
ARP	Address Resolution Protocol: Netzwerkprotokoll zur Zuordnung von Netzwerkadressen zu Hardwareadressen.
Bit-Variable	Variable, die bitweise adressiert wird.
CENELEC	Comité Européen de Normalisation Électrotechnique (Europäisches Komitee für elektrotechnische Normung)
Connector Board	Anschlusskarte für HIMax Modul.
COM	Kommunikationsmodul
CPU	Prozessormodul
CRC	Cyclic Redundancy Check, Prüfsumme
Dataview	Einer Dataview sind die Globalen Variablen für Eingangs- und Ausgangsdaten für den Zugriff durch Modbus-Quellen zugeordnet.
EN	Europäische Normen
Exportbereich	Als Exportbereich wird die Prozessdatenmenge bezeichnet, die vom System (aus einem Anwenderprogramm, HW-Eingang oder einem anderen Protokoll) geschrieben und vom Modbus Master gelesen werden kann.
FB	Feldbus
FBS	Funktionsbausteinsprache
ICMP	Internet Control Message Protocol: Netzwerkprotokoll für Status- und Fehlermeldungen.
IEC	Internationale Normen für die Elektrotechnik.
Importbereich	Als Importbereich wird die Prozessdatenmenge bezeichnet, die vom Modbus-Master geschrieben wird und als Eingangsdaten für das System (in einem Anwenderprogramm, HW-Ausgang oder einem anderen Protokoll) verwendet werden kann.
KE	Kommunikationsendpunkt
MAC-Adresse	Hardware-Adresse eines Netzwerkanschlusses (Media Access Control).
NSIP	Nicht-sicherheitsbezogenes Protokoll.
PADT	Programming and Debugging Tool (nach IEC 61131-3), PC mit SILworX.
PE	Schutzerde
PELV	Protective Extra Low Voltage: Funktionskleinspannung mit sicherer Trennung.
PES	Programmierbares Elektronisches System
R	Read
Rack-ID	Identifikation eines Basisträgers (Nummer).
rückwirkungsfrei	Es seien zwei Eingangsschaltungen an dieselbe Quelle (z. B. Transmitter) angeschlossen. Dann wird eine Eingangsschaltung „rückwirkungsfrei“ genannt, wenn sie die Signale der anderen Eingangsschaltung nicht verfälscht.
R/W	Read/Write
Register-Variable	Variable, die wortweise adressiert wird.
SB	Systembusmodul
SFF	Safe Failure Fraction, Anteil der sicher beherrschbaren Fehler.
SIF	Sicherheitstechnische Funktion
SIL	Safety Integrity Level (nach IEC 61508)
SILworX	Programmiersoftware für HIMax, HIQuad X und HIMatrix.
SIP	Sicherheitsbezogenes Protokoll
SNTP	Simple Network Time Protocol (RFC 1769)
SRS	System.Rack.Slot
SW	Software

Begriff	Beschreibung
TMO	Timeout
W	Write
WD	Watchdog
WDZ	Watchdog-Zeit

Abbildungsverzeichnis

Bild 1:	Verschaltung HIMax und Siemens Steuerung	13
Bild 2:	Datenübertragung zwischen HIMax und Siemens Steuerung	14
Bild 3:	Variablenliste im Siemens-Baustein UDT1	15
Bild 4:	Variablenliste im Siemens-Baustein DB1	16
Bild 5:	SIMATIC-Symboleditor	16
Bild 6:	Funktionsplan zum Empfangen	17
Bild 7:	Funktionsplan zum Senden	18
Bild 8:	Eigenschaften der TCP-Verbindung in SILworX	19
Bild 9:	Variablenliste Siemens	29
Bild 10:	Variablenliste HIMA	29
Bild 11:	Funktionsbaustein TCP_Reset	31
Bild 12:	Funktionsbaustein TCP_Send	34
Bild 13:	Funktionsbaustein TCP_Receive	37
Bild 14:	Funktionsbaustein TCP_ReceiveLine	41
Bild 15:	Funktionsbaustein TCP_ReceiveVar	45
Bild 16:	Aufbau der Datenpakete	46
Bild 17:	Funktionsbaustein PNM_MSTST (oberer Teil)	54
Bild 18:	Funktionsbaustein PNM_MSTST (unterer Teil)	55
Bild 19:	Auswahl Funktionsbausteine	55
Bild 20:	Systemvariablen des Funktionsbausteins MSTAT	55

Tabellenverzeichnis

Tabelle 1:	Zusätzlich geltende Handbücher	5
Tabelle 2:	Systemanforderung und Ausstattung S/R TCP	12
Tabelle 3:	Eigenschaften S/R TCP	12
Tabelle 4:	Konfiguration HIMax Steuerung	14
Tabelle 5:	Konfiguration Siemens SIMATIC 300	14
Tabelle 6:	Globale Variablen	19
Tabelle 7:	Variablen für Empfangsdaten	20
Tabelle 8:	Variablen für Sendedaten	20
Tabelle 9:	Systemvariablen S/R TCP	21
Tabelle 10:	Allgemeine Eigenschaften S/R TCP	22

Tabelle 11: Parameter COM/CPU	23
Tabelle 12: Systemvariablen	24
Tabelle 13: Eigenschaften S/R TCP-Verbindung	26
Tabelle 14: Funktionsbausteine für S/R TCP Verbindungen	30
Tabelle 15: A-Eingänge Funktionsbaustein TCP_Reset	31
Tabelle 16: A-Ausgänge Funktionsbaustein TCP_Reset	31
Tabelle 17: F-Eingänge Funktionsbaustein TCP_Reset	32
Tabelle 18: F-Ausgänge Funktionsbaustein TCP_Reset	32
Tabelle 19: Eingangs-Systemvariablen	32
Tabelle 20: Ausgangs-Systemvariablen	33
Tabelle 21: A-Eingänge Funktionsbaustein TCP_Send	34
Tabelle 22: A-Ausgänge Funktionsbaustein TCP_Send	34
Tabelle 23: F-Eingänge Funktionsbaustein TCP_Send	35
Tabelle 24: F-Ausgänge Funktionsbaustein TCP_Send	35
Tabelle 25: Eingangs-Systemvariablen	35
Tabelle 26: Ausgangs-Systemvariablen	36
Tabelle 27: Sende-Daten	36
Tabelle 28: A-Eingänge Funktionsbaustein TCP_Receive	37
Tabelle 29: A-Ausgänge Funktionsbaustein TCP_Receive	38
Tabelle 30: A-Eingänge Funktionsbaustein TCP_Receive	38
Tabelle 31: F-Ausgänge Funktionsbaustein TCP_Receive	38
Tabelle 32: Eingangs-Systemvariablen	39
Tabelle 33: Ausgangs-Systemvariablen	39
Tabelle 34: Empfangsvariablen	39
Tabelle 35: A-Eingänge Funktionsbaustein TCP_ReceiveLine	41
Tabelle 36: A-Ausgänge Funktionsbaustein TCP_ReceiveLine	42
Tabelle 37: F-Eingänge Funktionsbaustein TCP_ReceiveLine	42
Tabelle 38: F-Ausgänge Funktionsbaustein TCP_ReceiveLine	42
Tabelle 39: Eingangs-Systemvariablen	43
Tabelle 40: Ausgangs-Systemvariablen	43
Tabelle 41: Empfangsvariablen	43
Tabelle 42: A-Eingänge Funktionsbaustein TCP_ReceiveVar	46
Tabelle 43: A-Ausgänge Funktionsbaustein TCP_ReceiveVar	47
Tabelle 44: F-Eingänge Funktionsbaustein TCP_ReceiveVar	47
Tabelle 45: F-Ausgänge Funktionsbaustein TCP_ReceiveVar	47
Tabelle 46: Eingangs-Systemvariablen	48
Tabelle 47: Ausgangs-Systemvariablen	48
Tabelle 48: Empfangsvariablen	48
Tabelle 49: Anzeigefeld S/R-Protokoll	50
Tabelle 50: Anzeigefeld Modbus Slaves	50

Tabelle 51: Fehlercodes der TCP-Verbindung	51
Tabelle 52: Zusätzliche Fehlercodes	52
Tabelle 53: Verbindungszustand	52
Tabelle 54: Partner Verbindungszustand	52

HANDBUCH
Send/Receive TCP

HI 801 516 D

Für weitere Informationen kontaktieren Sie:

HIMA Paul Hildebrandt GmbH

Albert-Bassermann-Str. 28
68782 Brühl, Germany

Telefon +49 6202 709-0
Fax +49 6202 709-107
E-Mail info@hima.com

Erfahren Sie online mehr über HIMA Lösungen:



www.hima.com/de/



www.hima.com