



SEW
EURODRIVE

Systemhandbuch



Dezentrale Sicherheitssteuerung MOVISAFE® HM31



Inhaltsverzeichnis

1	Allgemeine Hinweise.....	9
1.1	Aufbau und Gebrauch der Dokumentation	9
1.2	Zielgruppe	10
1.3	Darstellungskonventionen	10
1.4	Aufbau der Warnhinweise	10
1.4.1	Bedeutung der Signalworte.....	10
1.4.2	Aufbau der abschnittsbezogenen Warnhinweise.....	10
1.4.3	Aufbau der eingebetteten Warnhinweise	11
1.5	Mängelhaftungsansprüche	11
1.6	Haftungsausschluss	11
1.7	Mitgelte Unterlagen	11
1.8	Urheberrechtsvermerk	12
1.9	Produktnamen und Marken	12
2	Systemeigenschaften.....	13
2.1	Überwachung der Betriebsspannung	13
2.2	Überwachung des Temperaturzustandes	13
2.2.1	Einstellung der Temperaturschwelle für Meldungen.....	14
2.2.2	Kurzschlussverhalten der Ausgangskanäle	14
2.3	Alarm- und Ereignisaufzeichnung	14
2.3.1	Alarmer und Ereignisse.....	14
2.3.2	Bildung von Ereignissen	15
2.3.3	Aufzeichnung von Ereignissen.....	15
2.3.4	Weitergabe von Ereignissen	15
3	Kommunikation	16
3.1	Ethernet	17
3.2	Kommunikation mit dem Programmierwerkzeug	17
3.3	Registrierung und Aktivierung der Protokolle	18
3.4	Lastbegrenzung	19
4	safeethernet	20
4.1	Was ist safeethernet?	21
4.2	safeethernet-Editor	23
4.3	Detailansicht des safeethernet-Editors	25
4.3.1	Register: Systemvariablen.....	25
4.4	safeethernet Parameter	28
4.4.1	Maximale Zykluszeit der Sicherheitssteuerung.....	28
4.4.2	Receive Timeout	28
4.4.3	Response Time.....	29
4.4.4	Sync/Async	29
4.4.5	Resend Timeout.....	30
4.4.6	Acknowledge Timeout.....	30
4.4.7	Production Rate	30
4.4.8	Speicher.....	30
4.5	Maximale Reaktionszeit für safeethernet	31

4.5.1	Berechnung der maximalen Reaktionszeit	32
4.5.2	safeethernet-Profil	32
4.5.3	Profil I (Fast & Cleanroom)	33
4.5.4	Profil II (Fast & Noisy)	34
4.5.5	Profil III (Medium & Cleanroom)	34
4.5.6	Profil IV (Medium & Noisy)	35
4.5.7	Profil V (Slow & Cleanroom)	35
4.5.8	Profil VI (Slow & Noisy)	36
4.6	Projektübergreifende Kommunikation	36
4.6.1	Varianten zur projektübergreifenden Kommunikation	37
4.7	Control Panel (safeethernet)	38
4.7.1	Anzeigefeld (safeethernet-Verbindung)	39
4.8	Maximale Kommunikationszeitscheibe	40
4.9	Anschlüsse für safeethernet/Ethernet	40
5	PROFINET IO	41
5.1	Steuerung des Consumer/Provider Status (IOxS)	41
5.1.1	Steuervariablen im Controller	41
5.1.2	Steuervariablen im DO-Device	42
5.1.3	Steuervariablen im DI-Device	42
5.2	PROFIsafe	42
5.2.1	PROFIsafe Control-Byte und Status-Byte	43
5.2.2	PROFIsafe Watchdog-Zeit (F_WD_Time)	43
5.2.3	Safety Function Response Time (SFRT)	45
5.3	Auflagen für den sicheren Betrieb von PROFIsafe	46
5.3.1	Adressierung	46
5.3.2	Netzwerkaspekte	47
5.4	PROFINET IO-Controller und PROFIsafe F-Host	48
5.5	PROFINET-IO / PROFIsafe-Beispiel (Controller)	49
5.5.1	Anlegen des PROFINET-IO-Controllers in SILworX®	49
5.6	Menüfunktionen des PROFINET-IO-Controllers	54
5.6.1	Beispiel für einen Strukturbaum des PROFINET-IO-Controllers	54
5.6.2	PROFINET-IO Controller	54
5.6.3	PROFINET-IO Device (im Controller)	56
5.6.4	DAP-Modul (Device Access Point Modul)	56
5.6.5	Input/Output PROFINET-IO Module	58
5.6.6	Submodul Input	59
5.6.7	Submodul Output	66
5.6.8	Submodul Inputs und Outputs	68
5.6.9	Applikationsbeziehung (Eigenschaften)	70
5.6.10	Alarm CR (Eigenschaften)	71
5.6.11	Input CR (Eigenschaften)	72
5.7	PROFINET-IO-Device	75
5.7.1	Systemanforderung	75
5.8	PROFINET-IO / PROFIsafe-Beispiel (Device)	75
5.8.1	Konfiguration des PROFINET-IO-Device in SILworX®	76
5.9	Menüfunktionen des PROFINET-IO-Device	78

5.9.1	Menü Eigenschaften	78
5.9.2	PROFINET-IO-Module	79
5.9.3	PROFIsafe-Module	80
5.9.4	PROFINET-IO- und PROFIsafe-Modul	82
5.10	PROFINET-IO-Funktionsbausteine	85
5.10.1	Funktionsbaustein MSTAT	86
5.10.2	Funktionsbaustein RALRM	89
5.10.3	Funktionsbaustein RDREC	93
5.10.4	Funktionsbaustein SLACT	96
5.10.5	Funktionsbaustein WRREC	99
5.11	Konfiguration der Funktionsbausteine	102
5.11.1	Hinzufügen der Funktionsbausteinbibliotheken	102
5.11.2	Konfiguration der Funktionsbausteine im Anwenderprogramm	102
5.11.3	Konfiguration der Funktionsbausteine im Strukturbaum von SILworX®	103
5.12	PROFINET-IO-Hilfsfunktionsbausteine	104
5.12.1	Hilfsfunktionsbaustein ACTIVE	104
5.12.2	Hilfsfunktionsbaustein ALARM	105
5.12.3	Hilfsfunktionsbaustein DEID	107
5.12.4	Hilfsfunktionsbaustein ID	107
5.12.5	Hilfsfunktionsbaustein NSLOT	108
5.12.6	Hilfsfunktionsbaustein SLOT	108
5.12.7	Hilfsfunktionsbaustein STDDIAG	109
5.13	Fehlercodes der Funktionsbausteine	110
6	Modbus TCP/UDP	112
6.1	Modbus-Master	112
6.1.1	Modbus-Beispiel	113
6.1.2	Beispiel zur alternativen Register/Bit-Adressierung	117
6.1.3	Menüfunktionen des Modbus-Master	118
6.1.4	Modbus Funktionscodes des Masters	121
6.1.5	Format der Request und Response Header	123
6.1.6	Anforderungstelegramme zum Lesen	124
6.1.7	Anforderungstelegramm zum Lesen und Schreiben	125
6.1.8	Anforderungstelegramm zum Schreiben	127
6.1.9	Ethernet-Slaves (TCP/UDP-Slaves)	128
6.1.10	Control-Panel (Modbus-Master)	131
6.1.11	Control-Panel (Modbus-Master → Slave)	132
6.2	Modbus-Slave	132
6.2.1	Konfiguration des Modbus-TCP-Slave	133
6.2.2	Menüfunktionen des Modbus-Slave-Set	133
6.2.3	Senden-/Empfangsvariablen zuweisen	136
6.2.4	Systemvariablen Modbus-Slave-Set	136
6.2.5	Modbus-Funktionscodes des Modbus-Slaves	138
6.2.6	SEW-spezifische Funktionscodes	140
6.2.7	Modbus-Adressierung durch Bit und Register	142
6.2.8	Offsets für alternative Modbus-Adressierung	145
6.2.9	Control-Panel (Modbus-Slave)	148

	6.2.10 Fehlercodes der Modbus TCP/IP-Verbindung	149
7	Send & Receive TCP	151
7.1	Systemanforderungen	151
7.1.1	Anlegen eines Send & Receive-TCP-Protokolls	151
7.2	Beispiel Send & Receive TCP-Konfiguration	152
7.2.1	Send & Receive TCP-Konfiguration der Steuerung SIMATIC 300	154
7.2.2	Send & Receive TCP-Konfiguration von MOVISAFE® HM31	157
7.3	Menüfunktionen im Send & Receive TCP-Protokoll	159
7.3.1	Edit.....	159
7.3.2	Eigenschaften	159
7.4	Menüfunktionen TCP-Verbindung	161
7.4.1	Edit.....	161
7.4.2	Systemvariablen	161
7.4.3	Eigenschaften	162
7.5	Datenaustausch	164
7.5.1	TCP-Verbindungen	165
7.5.2	Zyklischer Datenaustausch.....	165
7.5.3	Azyklischer Datenaustausch mit Funktionsbausteinen.....	166
7.5.4	Gleichzeitiger zyklischer und azyklischer Datenaustausch.....	166
7.5.5	Flusskontrolle	166
7.6	Fremdsysteme mit Pad-Bytes	167
7.7	Send & Receive TCP-Funktionsbausteine	167
7.7.1	TCP_Reset	168
7.7.2	TCP_Send	171
7.7.3	TCP_Receive.....	174
7.7.4	TCP_ReceiveLine	177
7.7.5	TCP_ReceiveVar	181
7.8	Control-Panel (Send/Receive over TCP)	185
7.8.1	Anzeigefeld allgemeine Parameter	185
7.8.2	Anzeigefeld TCP-Verbindungen	185
7.8.3	Fehlercodes der TCP-Verbindung	186
7.8.4	Zusätzliche Fehlercodetabelle der Funktionsbausteine.....	187
7.8.5	Verbindungszustand	188
7.8.6	Partner Verbindungszustand	188
8	Simple Network Time Protokoll (SNTP).....	189
8.1	SNTP-Client	189
8.1.1	Neuen SNTP-Client anlegen.....	189
8.2	SNTP-Client (Server Info)	190
8.2.1	Neue SNTP-Server Info anlegen	190
8.3	SNTP-Server	191
8.3.1	Neuen SNTP-Server anlegen	191
9	Com-User Task (CUT)	193
9.1	Eigenschaften der CUT	193
9.2	Voraussetzung	193

10	Betriebssystem.....	194
10.1	Funktionen des Prozessor-Betriebssystems	194
10.2	Verhalten bei Auftreten von Fehlern	194
10.2.1	Permanente Fehler bei Eingängen und Ausgängen	195
10.2.2	Vorübergehende Fehler bei Eingängen und Ausgängen.....	195
10.2.3	Interne Fehler.....	195
10.3	Das Prozessorsystem	195
10.3.1	Betriebszustände des Prozessorsystems	196
10.3.2	Programmierung	196
11	Anwenderprogramm	197
11.1	Betriebsarten des Anwenderprogramms	198
11.2	Multitasking	198
11.2.1	CPU-Zyklus ohne Multitasking	198
11.2.2	CPU-Zyklus mit Multitasking	198
11.2.3	Multitasking-Mode	203
11.3	Reload	206
11.3.1	Bedingungen für die Verwendung von Reload	207
11.4	Allgemeines zum Forcen	208
11.5	Forcen	209
11.5.1	Zeitbegrenzung	210
11.5.2	Force-Editor	210
11.5.3	Einschränkung des Forcens	210
12	Inbetriebnahme.....	211
12.1	Checkliste zur Projektierung, Programmierung und Inbetriebnahme	211
12.2	Konfiguration mit SILworX®	211
12.2.1	Prozessormodul	211
12.2.2	Kommunikationsmodul.....	217
12.2.3	Konfiguration der Ressource	217
12.2.4	Konfiguration der Ein- und Ausgänge	227
12.2.5	Generierung der Ressourcenkonfiguration	228
12.2.6	System-ID und Verbindungsparameter konfigurieren.....	229
12.2.7	Ressourcenkonfiguration vom Programmiergerät laden.....	229
12.2.8	Ressourcenkonfiguration aus dem Flash-Speicher des Kommunikationssystems laden	230
12.2.9	Ressourcenkonfiguration im Flash-Speicher des Kommunikationssystems bereinigen	230
12.2.10	Datum und Uhrzeit setzen	231
12.3	Benutzerverwaltung mit SILworX®	231
12.3.1	Benutzerverwaltung für ein SILworX®-Projekt.....	231
12.3.2	Benutzerverwaltung für die Steuerung.....	232
12.4	Konfiguration der Kommunikation mit SILworX®	235
12.4.1	Konfiguration der Ethernet-Schnittstellen	235
12.5	Konfigurieren von Alarmen und Ereignissen	236
12.6	Umgang mit dem Anwenderprogramm	240
12.6.1	Setzen der Parameter und Schalter.....	240
12.6.2	Starten des Programms von STOPP/GÜLTIGE KONFIGURATION	240

12.6.3	Neustart des Programms nach Fehler	240
12.6.4	Stoppen des Programms	240
12.6.5	Testmodus des Programms	240
13	Betrieb	242
13.1	Bedienung	242
13.2	Diagnose	242
13.2.1	LED-Anzeige	242
13.2.2	Diagnosehistorie	244
13.2.3	Diagnose in SILworX®	245
13.3	Parameter und Fehlercodes der Ein- und Ausgänge	246
13.3.1	Digitale Eingänge MOVISAFE® HM31	246
13.3.2	Digitale Ausgänge MOVISAFE® HM31	249
13.3.3	Zähler MOVISAFE® HM31	251
14	Service	255
14.1	Inspektion / Wartung	255
14.2	Gerätetausch	255
14.2.1	Voraussetzungen	255
14.2.2	Verbindung zur Sicherheitssteuerung	255
14.2.3	Verifizierung der Systemdaten	257
14.2.4	Diagnosedaten sichern (CPU und COM)	260
14.2.5	Inbetriebnahme MOVISAFE® HM31 mit Werkseinstellungen	262
14.2.6	Inbetriebnahme MOVISAFE® HM31 ohne Werkseinstellungen	264
14.2.7	Ressource (MOVISAFE® HM31) laden und starten	267
14.2.8	Elektrische Installation	270
14.2.9	Verifikation	270
14.3	Störungsinformation	270
14.4	Laden von Betriebssystemen	271
14.4.1	Laden von Betriebssystemen mit SILworX®	271
14.5	Außerbetriebnahme	272
15	Anhang	273
15.1	Glossar	273
	Stichwortverzeichnis	275

1 Allgemeine Hinweise

Dieses Handbuch enthält Informationen für den bestimmungsgemäßen Gebrauch der Sicherheitssteuerung.

Voraussetzung für die risikolose Installation, Inbetriebnahme und für die Sicherheit bei Betrieb und Instandhaltung sind:

- Kenntnis von Vorschriften
- Technisch einwandfreie Umsetzung der in diesem Handbuch enthaltenen Sicherheitshinweise durch qualifiziertes Personal

In folgenden Fällen können durch Störungen oder Beeinträchtigungen von Sicherheitsfunktionen schwere Personen-, Sach- oder Umweltschäden eintreten, für die SEW-EURODRIVE keine Haftung übernehmen kann:

- Bei nicht qualifizierten Eingriffen in die Geräte
- Bei Abschalten oder Umgehen (Bypass) von Sicherheitsfunktionen
- Bei Nichtbeachtung von Hinweisen dieses Handbuchs

SEW-EURODRIVE entwickelt, fertigt und prüft Sicherheitssteuerungen unter Beachtung der einschlägigen Sicherheitsnormen. Die Verwendung der Geräte ist nur zulässig, wenn alle folgenden Voraussetzungen erfüllt sind:

- Nur die in den Beschreibungen vorgesehenen Einsatzfälle
- Nur die spezifizierten Umgebungsbedingungen
- Nur in Verbindung mit zugelassenen Fremdgeräten

1.1 Aufbau und Gebrauch der Dokumentation

Dieses Systemhandbuch enthält folgende Themen:

- Allgemeine Hinweise
- Systemeigenschaften
- Kommunikation
- Safeethernet
- PROFINET IO
- Modbus TCP/UDP
- Send & Receive TCP
- SNTP-Protokoll
- Com-User Task (CUT)
- Betriebssystem
- Anwenderprogramm
- Inbetriebnahme
- Betrieb
- Instandhaltung

Das Handbuch beschreibt folgende Variante:

Programmierwerkzeug	Prozessor-Betriebssystem	Kommunikations-Betriebssystem
SILworX®	Ab CPU-BS V.8	Ab COM-BS V.13

1.2 Zielgruppe

Dieses Dokument wendet sich an Planer, Projektoren und Programmierer von Automatisierungsanlagen sowie Personen, die zu Inbetriebnahme, Betrieb und Wartung der Geräte und Systeme berechtigt sind. Vorausgesetzt werden spezielle Kenntnisse auf dem Gebiet der sicherheitsgerichteten Automatisierungssysteme.

1.3 Darstellungskonventionen

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

Schreibweise	Bedeutung
Fett	Hervorhebung wichtiger Textteile.
[...]	Bezeichnungen von Schaltflächen und Menübefehlen im Programmierwerkzeug, auf die Sie klicken können.
<i>Kursiv</i>	Parameter und Systemvariablen.
<code>Courier</code>	Wörtliche Benutzereingaben.
RUN	Bezeichnungen von Betriebszuständen in Großbuchstaben.

1.4 Aufbau der Warnhinweise

1.4.1 Bedeutung der Signalworte

Die folgende Tabelle zeigt die Abstufung und Bedeutung der Signalworte der Warnhinweise.

Signalwort	Bedeutung	Folgen bei Missachtung
▲ GEFAHR	Unmittelbar drohende Gefahr	Tod oder schwere Verletzungen
▲ WARNUNG	Mögliche, gefährliche Situation	Tod oder schwere Verletzungen
▲ VORSICHT	Mögliche, gefährliche Situation	Leichte Verletzungen
ACHTUNG	Mögliche Sachschäden	Beschädigung des Antriebssystems oder seiner Umgebung
HINWEIS	Nützlicher Hinweis oder Tipp: Erleichtert die Handhabung des Antriebssystems.	

1.4.2 Aufbau der abschnittsbezogenen Warnhinweise

Die abschnittsbezogenen Warnhinweise gelten nicht nur für eine spezielle Handlung, sondern für mehrere Handlungen innerhalb eines Themas. Die verwendeten Gefahrensymbole weisen entweder auf eine allgemeine oder spezifische Gefahr hin.



Hier sehen Sie den formalen Aufbau eines abschnittsbezogenen Warnhinweises:

SIGNALWORT!

Art der Gefahr und ihre Quelle.

Mögliche Folge(n) der Missachtung.

- Maßnahme(n) zur Abwendung der Gefahr.

1.4.3 Aufbau der eingebetteten Warnhinweise

Die eingebetteten Warnhinweise sind direkt in die Handlungsanleitung vor dem gefährlichen Handlungsschritt integriert.

Hier sehen Sie den formalen Aufbau eines eingebetteten Warnhinweises:

- **▲ SIGNALWORT!** Art der Gefahr und ihre Quelle.
Mögliche Folge(n) der Missachtung.
 - Maßnahme(n) zur Abwendung der Gefahr.

1.5 Mängelhaftungsansprüche

Die Einhaltung der Dokumentation ist die Voraussetzung für den störungsfreien Betrieb und die Erfüllung eventueller Mängelhaftungsansprüche. Lesen Sie deshalb zuerst die Dokumentation, bevor Sie mit dem Gerät arbeiten!

1.6 Haftungsausschluss

Die Beachtung der Dokumentation ist Grundvoraussetzung für den sicheren Betrieb und für das Erreichen der angegebenen Produkteigenschaften und Leistungsmerkmale. Für Personen-, Sach- oder Vermögensschäden, die wegen Nichtbeachtung der Betriebsanleitung entstehen, übernimmt SEW-EURODRIVE keine Haftung. Die Sachmängelhaftung ist in solchen Fällen ausgeschlossen.

1.7 Mitgeltende Unterlagen

Beachten Sie die folgenden mitgeltenden Unterlagen:

- Betriebsanleitung "Dezentrale Sicherheitssteuerung MOVISAFE® HM31"
- Sicherheitshandbuch "Dezentrale Sicherheitssteuerung MOVISAFE® HM31"
- Praxis der Antriebstechnik - EMV in der Antriebstechnik

Wenn Sie die CUT-Funktionalität nutzen möchten, beachten Sie zusätzlich die folgenden mitgeltenden Unterlagen:

- Handbuch "Com-User Task für MOVISAFE® HM31"
- Handbuch "MOVIVISION® Parameter- und Diagnosetool Version 2.0"

Sie benötigen Software, die **nicht** im Lieferumfang ist. Sie können die Software zusammen mit der Dokumentation auf einem Datenträger (CD/DVD) von SEW-EURODRIVE unter folgenden Bestellangaben beziehen:

Bezeichnung	Sachnummer
SILworX® für MOVISAFE® HM31 <ul style="list-style-type: none"> Hardware: SILworX® Lizenz Dongle Software: SILworX® ab 4.64.0 	19500114
Motion Library MOVISAFE® HM31 Bausteinbibliothek für die sichere Wegmessung /Function block library for safety related position detection	17106400

Beachten Sie darüber hinaus die mitgeltenden Unterlagen in Abhängigkeit zu der angeschlossenen Antriebstechnik

Sie finden die jeweils aktuelle Version der Dokumentation auf der SEW-Homepage (www.sew-eurodrive.de) in der Rubrik "Dokumentationen".

1.8 Urheberrechtsvermerk

© 2014 SEW-EURODRIVE. Alle Rechte vorbehalten.

Jegliche – auch auszugsweise – Vervielfältigung, Bearbeitung, Verbreitung und sonstige Verwertung sind verboten.

1.9 Produktnamen und Marken

Die in dieser Dokumentation genannten Produktnamen sind Marken oder eingetragene Marken der jeweiligen Titelhälter.

2 Systemeigenschaften

Die Sicherheitssteuerung enthält in einem Gehäuse ein sicherheitsgerichtetes Prozessorsystem, eine Anzahl Eingänge und Ausgänge sowie Kommunikationsanschlüsse.

Details entnehmen Sie der Betriebsanleitung "Dezentrale Sicherheitssteuerung MOVISAFE® HM31".

2.1 Überwachung der Betriebsspannung

Das Gerät überwacht die Spannung DC 24 V während des Betriebs. Reaktionen erfolgen entsprechend der aufgelisteten Spannungspegel.

Spannungspegel	Reaktion der Geräte
DC 24 V -20% / +25% (19.2 V – 30 V)	Normalbetrieb
< 18.0 V (softwareseitig ausgelesene Spannung auf der Platine)	Alarmzustand (interne Variable werden beschrieben und an die Eingänge und Ausgänge gegeben)
< 12.0 V (softwareseitig ausgelesene Spannung auf der Platine)	Abschaltung der Eingänge und Ausgänge

Die Systemvariable *Stromversorgungszustand* erlaubt es, den Zustand der Betriebsspannung mit dem Programmierwerkzeug oder im Anwenderprogramm auszuwerten.

2.2 Überwachung des Temperaturzustandes

Die Temperatur wird durch Sensoren an relevanten Stellen im Innern des Gerätes oder des Systems gemessen und softwareseitig ausgegeben.

Diese hat einen Delta-Betrag zu der Umgebungstemperatur, welcher von vielen Faktoren abhängt.

Überschreitet die geräteintern gemessene Temperatur die definierten Schaltschwellen, ändert sich der Wert der Systemvariable "Temperaturzustand" wie folgt:

Temperatur (geräteintern)	Temperaturbereich	Temperaturzustand [BYTE]
< 60 °C	Normal	0x00
60 °C bis 70 °C	Hohe Temperatur	0x01
> 70 °C	Sehr hohe Temperatur	0x03
Rückkehr auf 64 °C bis 54 °C ¹⁾	Hohe Temperatur	0x01
Rückkehr auf < 54 °C ¹⁾	Normal	0x00

1) Die Sensoren haben eine Hysterese von 6 °C.

Bei mangelnder oder fehlender Luftzirkulation und nicht ausreichender Eigenkonvektion kann die Schwelle zum Bereich "Hohe Temperatur" in der Sicherheitssteuerung schon bei Umgebungstemperaturen < 35 °C ansprechen. Ursachen können lokale Erwärmungen oder eine ungünstige Wärmeableitung sein. Insbesondere bei digitalen Ausgängen ist die Erwärmung stark von der Belastung abhängig. Die Systemvariable

„Temperaturzustand“ ermöglicht dem Anwender die interne Temperatur auszulesen. Um die hohe Lebensdauer der Sicherheitssteuerung zu erhalten, empfiehlt SEW-EURODRIVE, bei häufigem Auftreten des Zustands „sehr hohe Temperatur“ die Wärmeabfuhr des Systems zu verbessern, z. B. durch zusätzliche Kühlung oder Lüftung.

HINWEIS



Der Übergang in den Zustand „hohe Temperatur“ oder „sehr hohe Temperatur“ bedeutet nicht, dass die Sicherheit des Systems beeinträchtigt ist.

2.2.1 Einstellung der Temperatururschwelle für Meldungen

Für die Sicherheitssteuerung ist einstellbar, welche Schwelle bei einer Temperaturüberschreitung zu einer Meldung führt. Die Parametrierung erfolgt im SiLworX®-Hardware-Editor in der „Detailansicht der Steuerung“.

2.2.2 Kurzschlussverhalten der Ausgangskanäle

Die Sicherheitssteuerung schaltet bei einem Kurzschluss in einem Ausgangskanal den betroffenen Kanal ab. Bei mehreren Kurzschlüssen werden die Kanäle einzeln entsprechend ihrer Stromaufnahme abgeschaltet. Wird der maximal zulässige Gesamtstrom für alle Ausgänge überschritten, werden alle Ausgänge abgeschaltet und zyklisch wieder eingeschaltet.

HINWEIS



Die Steckverbinder für Ausgangskreise dürfen nicht mit angeschlossener Last gesteckt werden. Bei vorhandenen Kurzschlüssen kann der dabei auftretende Kurzschlussstrom die Klemmen beschädigen.

2.3 Alarm- und Ereignisaufzeichnung

Die Sicherheitssteuerung verfügt über die Fähigkeit, Alarme und Ereignisse aufzuzeichnen (Sequence of Events Recording, SER).

2.3.1 Alarme und Ereignisse

Ereignisse sind Änderungen des Zustands von Anlage oder Steuerung, die mit einem Zeitstempel versehen sind.

Alarme sind solche Ereignisse, die eine Erhöhung des Gefahrenpotenzials signalisieren.

Die Sicherheitssteuerung zeichnet als Ereignisse die Zustandsänderungen zusammen mit dem Zeitpunkt ihres Auftretens auf.

Die Sicherheitssteuerung unterscheidet boolesche und skalare Ereignisse.

Boolesche Ereignisse:

- Änderungen von Booleschen Variablen, z. B. von digitalen Eingängen.
- Alarm- und Normalzustand, diese sind den Zuständen der Variablen beliebig zuzuordnen

Skalare Ereignisse:

- Übergänge über Grenzwerte, die für eine skalare Variable definiert sind.
- Skalare Variable haben einen numerischen Datentyp, z. B. INT, REAL.

- Es sind zwei obere und zwei untere Grenzen möglich.
- Für die Grenzwerte muss gelten:
Oberste Grenze = obere Grenze = Normalbereich = untere Grenze = unterste Grenze.
- Eine Hysterese kann in folgenden Fällen wirken:
 - Bei Unterschreitung einer oberen Grenze.
 - Bei Überschreitung einer unteren Grenze.

Die Angabe einer Hysterese vermeidet eine unnötig große Menge an Ereignissen, wenn die globale Variable stark um einen Grenzwert schwankt.

Die Sicherheitssteuerung kann nur dann Ereignisse bilden, wenn diese in SILworX® definiert sind, siehe Kapitel "Konfigurieren von Alarmen und Ereignissen". Bis zu 4 000 Alarme und Ereignisse sind definierbar.

2.3.2 Bildung von Ereignissen

Das Prozessorsystem ist in der Lage, Ereignisse zu bilden. Das Prozessorsystem bildet die Ereignisse aus globalen Variablen und legt sie im Puffer ab, siehe "Aufzeichnung von Ereignissen". Die Ereignisbildung findet im Zyklus des Anwenderprogramms statt. Jedes gelesene Ereignis kann durch ein neu aufgetretenes Ereignis überschrieben werden.

Systemereignisse:

Außer den Ereignissen, die Änderungen von globalen Variablen oder Eingangssignalen registrieren, bilden die Prozessorsysteme folgende Arten von Systemereignissen:

- Überlauf: Es sind infolge von Pufferüberlauf Ereignisse nicht gespeichert worden. Der Zeitstempel des Überlauf-Ereignisses entspricht dem des Ereignisses, das den Überlauf erzeugt hat.
- Init: Der Ereignispuffer wurde initialisiert.

Systemereignisse enthalten die SRS-Identifikation des Geräts, das sie ausgelöst hat. Statusvariable stellen dem Anwenderprogramm den Ereigniszustand skalarer Ereignisse zur Verfügung. Jedem der folgenden Zustände kann als Statusvariable eine globale Variable vom Typ BOOL zugeordnet sein:

- Normal.
- Untere Grenze unterschritten.
- Unterste Grenze unterschritten. Obere Grenze überschritten.
- Oberste Grenze überschritten.

Die zugeordnete Statusvariable wird TRUE, wenn der betreffende Zustand erreicht ist.

2.3.3 Aufzeichnung von Ereignissen

Das Prozessorsystem sammelt die Ereignisse: Das Prozessorsystem speichert alle Ereignisse in seinem Puffer. Der Puffer ist im nichtflüchtigen Speicher angelegt und fasst 1000 Ereignisse. Ist der Puffer voll, werden keine neuen Ereignisse gespeichert, bis weitere Ereignisse gelesen und dadurch zum Überschreiben markiert wurden.

2.3.4 Weitergabe von Ereignissen

Die Ereignisse können über das MODBUS-Protokoll zur Antriebssteuerung (Beck-PC) oder über safeethernet zur übergeordneten Sicherheitssteuerung übertragen werden. Hierfür müssen zuvor im Anwenderprogramm die entsprechenden Variablen verknüpft werden. Die erweiterte Diagnose erfolgt über das PADT (SILworX®).

3 Kommunikation

Die Sicherheitssteuerung kommuniziert unter Nutzung folgender Protokolle:

- safeethernet
Sicherheitsgerichtetes Protokoll für die Kommunikation der Steuerungen untereinander
- Feldbusprotokoll Modbus TCP/UDP für den Anschluss externer Geräte oder Systeme
- Kommunikation mit dem Programmiergerät

Das Kommunikationssystem ist an das sicherheitsgerichtete Prozessorsystem angeschlossen.

Es ist mit den Feldbus-Schnittstellen über ein Dual-Port-RAM an das sichere Mikroprozessorsystem angebunden. An die Schnittstellen dürfen nur Geräte angeschlossen werden, die eine sichere elektrische Trennung gewährleisten.

Das Kommunikationssystem steuert die Kommunikation der Steuerung mit anderen Systemen über leistungsfähige Schnittstellen:

Verfügbare Protokolle

Die folgenden Protokolle sind für die Steuerung MOVISAFE® HM31 verfügbar und können folgendermaßen aktiviert werden.

Protokoll	Schnittstellen	Aktivierung
safeethernet	Ethernet	Funktion ist bei MOVISAFE® HM31 standardmäßig freigeschaltet.
SNTP Server/Client		
Modbus TCP/UDP Master		
COM-User Task	CAN (X4111_1/2) RS485 (X4011)	
OPC Server (läuft auf Host-PC)	Ethernet	Auf Anfrage wird bei SEW-EURODRIVE eine Lizenz generiert, mit der das gewünschte Protokoll freigeschaltet werden kann.
PROFINET IO Controller		
PROFINET IO Device		
Modbus TCP/UDP Slave		
TCP Send/Receive		
PROFIsafe Host		
PROFIsafe Device		

HINWEIS



Die optionalen Protokolle können in MOVISAFE® HM31 ohne Aktivierung für 5000 Betriebsstunden zu Testzwecken verwendet werden. Bei der Verwendung der nicht aktivierten Protokollen leuchtet die System-LED "ERROR" dauerhaft rot.

Nach Ablauf der 5000 Betriebsstunden läuft die Steuerung nicht mehr an.

- Bestellen Sie rechtzeitig die Lizenzen zur Freischaltung der benötigten Protokolle.

3.1 Ethernet

Die Sicherheitssteuerung enthält einen Ethernet-Switch mit Anschlüssen. Über diese Anschlüsse können mithilfe von Ethernet-Kabeln andere Geräte mit der Steuerung verbunden werden.

Es stehen die folgenden Schnittstellen zur Verfügung:

- **2 Ethernet-Schnittstellen:** X4233_1 und X4233_2

Die beiden Schnittstellen befinden sich auf der Anschlussleiste des Geräts

- **1 Ethernet-Service-Schnittstelle:** X4223

Zum Anschluss eines Programmiergeräts (PADT)

Ethernet-Switch

- Ein Ethernet-Switch ist im Gegensatz zu einem Hub in der Lage, Datenpakete zu analysieren und kurzfristig zu speichern, um dann eine zeitweilige gezielte Verbindung zwischen zwei Kommunikationspartnern (Sender/Empfänger) für die Übertragung der Daten aufzubauen. Dadurch werden die bei einem Hub üblichen Kollisionen vermieden und das Netzwerk entlastet. Zur gezielten Weiterleitung der Daten benötigt jeder Ethernet-Switch eine Adress-/Port-Zuordnungstabelle. Diese Tabelle wird in einem Selbstlernprozess vom Switch automatisch generiert. In ihr sind MAC-Adressen einem bestimmten Port im Ethernet-Switch zugeordnet. Eingehende Datenpakete werden anhand dieser Tabelle an den entsprechenden Port direkt weitergeleitet.
- Der Ethernet-Switch schaltet automatisch um, sowohl zwischen den Übertragungsraten 10 und 100 MBit/s, als auch zwischen Voll- und Halbduplex-Verbindungen. Damit steht in jeder Richtung der Datenübertragung die volle Bandbreite zur Verfügung (Vollduplexbetrieb).
- Ein Ethernet-Switch regelt die Kommunikation zwischen verschiedenen Endgeräten. Der Ethernet-Switch kann dabei bis zu 1000 absolute MAC-Adressen ansprechen.
- Autocrossing erkennt den Anschluss von Kabeln mit gekreuzten Adern, und der Ethernet-Switch stellt sich automatisch darauf ein.

HINWEIS



Bei der Konfiguration der sicherheitsgerichteten Kommunikation sind die Hinweise im Sicherheitshandbuch zu beachten.

3.2 Kommunikation mit dem Programmierwerkzeug

Die Kommunikation der Sicherheitssteuerung mit einem PADT erfolgt über Ethernet. Ein PADT ist ein PC / Laptop, auf dem das Programmierwerkzeug SILworX® installiert ist.

Es ist möglich, dass eine Steuerung gleichzeitig mit bis zu 5 PADTs kommuniziert. Dabei kann jedoch nur ein Programmierwerkzeug schreibend auf die Steuerung zugreifen. Alle übrigen können nur Informationen lesen. Bei jedem weiteren Versuch, eine schreibende Verbindung aufzubauen, erteilt die Steuerung nur einen lesenden Zugriff.

3.3 Registrierung und Aktivierung der Protokolle

Die folgenden Protokolle sind für die Steuerung MOVISAFE® HM31 verfügbar und können folgendermaßen aktiviert werden:

Protokoll	Schnittstellen	Aktivierung
safeethernet	Ethernet	Funktion ist bei MOVISAFE® HM31 standardmäßig freigeschaltet
SNTP Server/Client		
Modbus TCP/UDP Master		
COM-User Task	CAN (X4111_1/2) RS485 (X4011)	
OPC Server (läuft auf Host-PC)	Ethernet	Auf Anfrage bei SEW-EURODRIVE
PROFINET IO Controller		
PROFINET IO Device		
Modbus TCP/UDP Slave		
TCP Send/Receive		
PROFIsafe Host		
PROFIsafe Device		

So tragen Sie den Software-Freischaltcode in SILworX® ein:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Lizenzverwaltung].
2. Markieren Sie den Eintrag "Lizenzverwaltung" und wählen Sie im Kontextmenü den Menüpunkt [Neu] / [Lizenzschlüssel]. Ein neuer Lizenzschlüssel wird hinzugefügt.
3. Markieren Sie "Lizenzschlüssel" und wählen Sie im Kontextmenü den Menüpunkt [Eigenschaften].
4. Tragen Sie im Feld "Freischaltcode" den generierten Software-Freischaltcode ein.

HINWEIS



Die Lizenz ist untrennbar mit dieser System-ID verbunden. Eine Lizenz kann nur einmalig für eine bestimmte System-ID genutzt werden. Deshalb sollte die Freischaltung erst durchgeführt werden, wenn die System-ID eindeutig feststeht. Ein Software-Freischaltcode kann maximal 32 Lizenzen enthalten. Es können auch mehrere Freischaltcodes in der Lizenzverwaltung eingetragen werden. In eine Steuerung können maximal 64 Lizenzen geladen werden.

HINWEIS



Die Ethernet-Protokolle können in der Geräteoption PFF-HM31A1-E61-I111-00/000/000 ohne Aktivierung für 5000 Betriebsstunden zu Testzwecken verwendet werden. Bei der Verwendung der nicht aktivierten Protokollen leuchtet die System-LED "ERROR" dauerhaft rot.

Nach Ablauf der 5000 Betriebsstunden läuft die Steuerung nicht mehr an.

- Bestellen Sie rechtzeitig die Geräteoption mit den benötigten Protokollen.

3.4 Lastbegrenzung

Für jedes Kommunikationsprotokoll kann ein Rechenzeitbudget in % (μ P-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, wird das Kommunikationsprotokoll nicht komplett abgearbeitet.

Ist noch genügend zusätzliche Rechenzeit vorhanden, 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.

HINWEIS



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

4 safeethernet

Die Sicherheitssteuerung ist safeethernet-fähig. Sie kann sicherheitsgerichtet gemäß SIL 3 über Ethernet (100 Mbit/s) kommunizieren.

Die Ethernet-Schnittstellen der Sicherheitssteuerung sind simultan auch für andere Protokolle nutzbar.

Die safeethernet-Kommunikation zwischen den Steuerungen kann über verschiedene Ethernet-Netzwerktopologien erfolgen. Passen Sie die Parameter des safeethernet-Protokolls an das verwendete Ethernet-Netzwerk an, um Geschwindigkeit und Effizienz des Datentransfers zu erhöhen.

Diese Parameter können mit Hilfe sogenannter Netzwerkprofile eingestellt werden. Die werkseitige Einstellung der Parameter stellt die Kommunikation sicher, ohne dass sich der Anwender zunächst in Details der Netzwerkkonfiguration einarbeiten muss.

HINWEIS



Das safeethernet Protokoll ist sicherheitsgerichtet und TÜV-zertifiziert bis SIL 3 gemäß IEC 61508.

safeethernet-Eigenschaften

Element	Eigenschaften	Beschreibung
Benötigtes Modul/Steuerung	Integriertes Prozessormodul der Steuerung	safeethernet wird auf dem sicherheitsgerichteten Prozessormodul ausgeführt.
Ethernet-Schnittstellen	100 Mbit/s	Die verwendeten Ethernet-Schnittstellen sind simultan auch für andere Protokolle nutzbar.
Verbindungen	128	safeethernet-Verbindungen
Redundante Verbindungen	128	2-Kanal-Betrieb. Redundante safeethernet Verbindungen zwischen Steuerungen sind im safeethernet-Editor einstellbar.
Redundante Transportwege	Einschränkung da nur ein Gerät	Redundante safeethernet-Transportwege.
Prozessdatenmenge pro Verbindung	1100 Bytes	Pro safeethernet-Verbindung.

HINWEIS



Für safeethernet-Verbindungen zu Ressourcen außerhalb eines Projektes steht eine Exportfunktion zur Verfügung (siehe Kapitel "Projektübergreifende Kommunikation").

4.1 Was ist safeethernet?

Im Bereich der Prozess- und Automatisierungstechnik sind Anforderungen wie Determinismus, Zuverlässigkeit, Austauschbarkeit, Erweiterbarkeit und vor allem Sicherheit zentrale Themen.

safeethernet ist ein Übertragungsprotokoll zur Übertragung von sicherheitsgerichteten Daten bis SIL 3 auf Basis der Ethernet-Technologie.

safeethernet beinhaltet Mechanismen, die folgende Fehler erkennen und darauf sicherheitsgerichtet reagieren:

- Verfälschung von übertragenen Daten (doppelte, verlorene, geänderte Bits)
- Falsche Adressierung von Nachrichten (Sender, Empfänger)
- Falsche Reihenfolge von Daten (Wiederholung, Verlust, Tausch)
- Falsches Zeitverhalten (Verzögerung, Echo)

safeethernet basiert auf dem Standard IEEE 802.3.

safeethernet verwendet „unsichere Datenübertragungskanäle“ (Ethernet) nach dem Black-Channel-Prinzip und überwacht sie bei Sender und Empfänger durch sicherheitsgerichtete Protokollmechanismen. Dadurch sind Ethernet-Netzwerkkomponenten wie Hubs, Switches, Router innerhalb eines sicherheitsgerichteten Netzwerkes verwendbar.

safeethernet nutzt die Fähigkeiten von Standard-Ethernet in der Form, dass Sicherheit und Echtzeitfähigkeit ermöglicht werden. Ein spezieller Protokollmechanismus garantiert ein deterministisches Verhalten auch bei Ausfall oder Eintritt von Kommunikationsteilnehmern. Das System bindet neue Komponenten in das laufende System dann automatisch ein. Alle Komponenten eines Netzwerkes sind während des laufenden Betriebs austauschbar. Mit dem Einsatz von Switches lassen sich Übertragungszeiten klar definieren. Somit wird Ethernet echtzeitfähig.

Verbindungen zum firmeninternen Intranet als auch Verbindungen zum Internet sind mit safeethernet möglich. Damit ist nur noch ein Netzwerk für sichere und nicht sichere Datenübertragung nötig.

HINWEIS



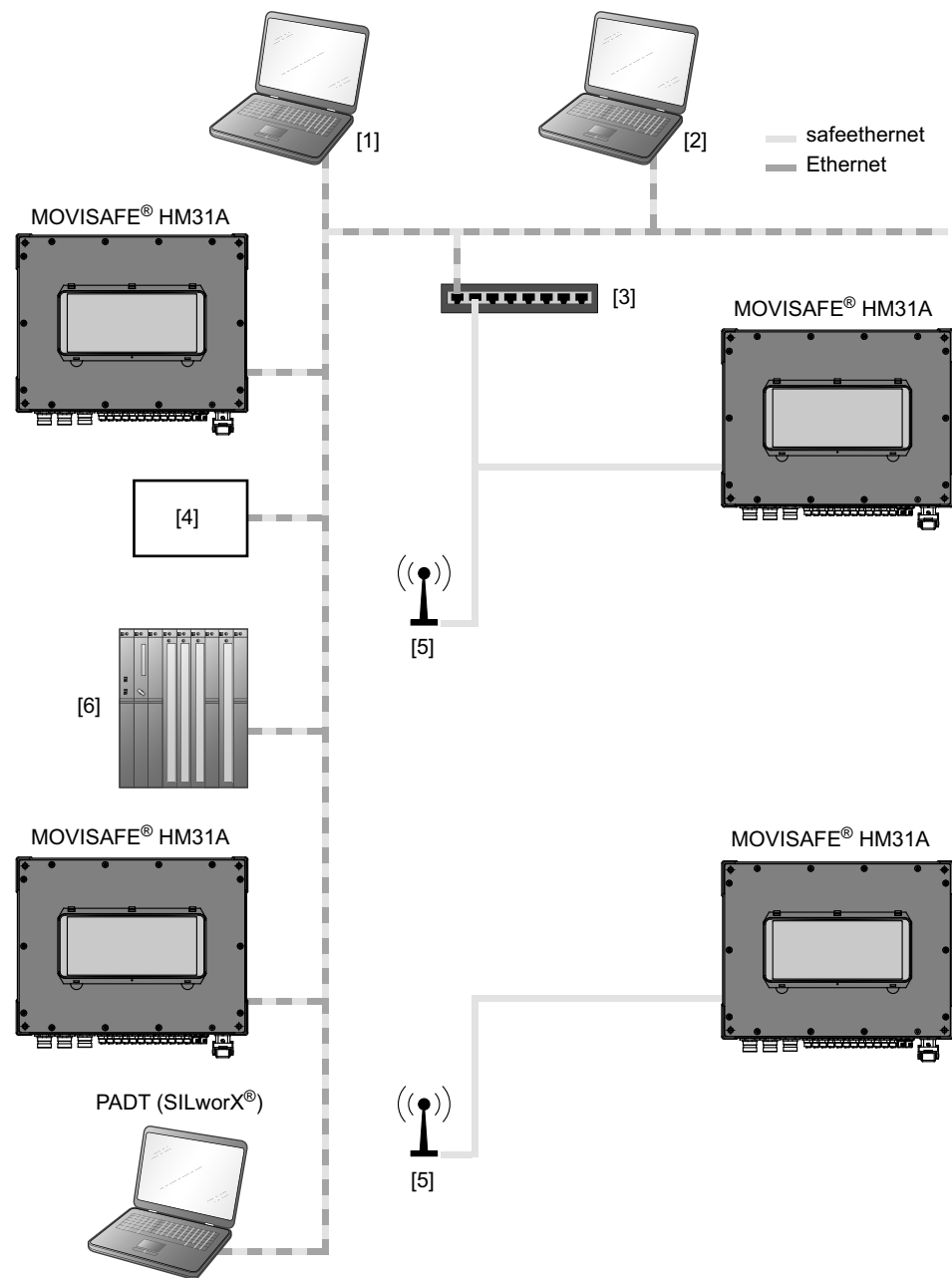
Das Netzwerk darf auch von anderen Teilnehmern benutzt werden, wenn genügend Übertragungskapazität zur Verfügung steht.



⚠ WARNUNG

Manipulation der sicherheitsgerichteten Datenübertragung!
Tod oder schwere Körperverletzung.

safeethernet ermöglicht flexible Systemstrukturen für die dezentrale Automatisierung mit definierten Reaktionszeiten. Je nach Anforderung können Sie die Intelligenz wahlweise zentral oder dezentral auf die Teilnehmer innerhalb des Netzwerkes verteilen.



9007204774660875

[1] PC des DCS-Leitsystems

[2] PADT (SILworX®)

[3] Switch

[4] DCS-Leitsystem

[5] Funk, Satellit, WLAN, Lichtwellenleiter, ISDN oder DSL

[6] SPS



HINWEIS

Unbeabsichtigter Übergang in den sicheren Zustand möglich!

- Bei der Zusammenschaltung ist zu beachten, dass keine Netzscheifen entstehen. Datenpakete dürfen nur auf **einem** Weg zu einer Steuerung gelangen.
- Verwenden Sie beim Aufbau einer Ethernet-Ringtopologie ausschließlich managementfähige Switches.

4.2 safeethernet-Editor

Im safeethernet-Editor erstellen und konfigurieren Sie die safeethernet-Verbindungen zu den Kommunikationspartnern (Ressourcen).

So öffnen Sie den safeethernet-Editor der lokalen Ressource:

1. Im Strukturbaum [Konfiguration] / [Ressource] öffnen.
2. Rechtsklick auf safeethernet und im Kontextmenü [Edit] wählen.

Der safeethernet-Editor enthält den Arbeitsbereich und die Objektauswahl.

Im safeethernet-Editor erstellen und konfigurieren Sie die safeethernet-Verbindungen zu den Kommunikationspartnern (Ressourcen). Dazu ziehen Sie die Ressourcen aus der Objektauswahl in den Arbeitsbereich.

Zur Konfiguration der safeethernet-Verbindung müssen Sie die folgenden safeethernet-Protokoll-Parameter einstellen:

Parameter	Beschreibung
Partner	Ressource-Name des Linkpartners.
IF CH...	Verfügbare Ethernet-Schnittstellen auf der Ressource (lokal) und Ressource (Ziel).
Profil	Kombination zueinander passender safeethernet-Parameter, siehe auch Kapitel "Safeethernet-Profiles".
Response Time [ms]	Zeit bis zur Empfangsbestätigung einer Nachricht beim Absender, siehe auch Kapitel "Response Time".
Receive Timeout [ms]	Überwachungszeit auf PES1, innerhalb der eine korrekte Antwort von PES2 empfangen werden muss, siehe auch Kapitel "Receive Timeout".
Resend Timeout [ms]	Überwachungszeit auf PES1, innerhalb welcher PES2 den Empfang eines Datenpakets bestätigt haben muss, ansonsten wird das Datenpaket wiederholt, siehe auch Kapitel "Resend Timeout".
Acknowledge Timeout [ms]	Zeit, nach der ein empfangenes Datenpaket von der CPU spätestens bestätigt werden muss, siehe auch Kapitel "Acknowledge Timeout".
Prod.-Rate	Produktionsrate: Kleinstes Zeitintervall zwischen zwei Datenpaketen, siehe auch Kapitel "Production Rate".
Speicher (Queue-Tiefe)	Anzahl der Datenpakete, die ohne Empfangsbestätigung versendet werden können, siehe auch Kapitel "Speicher".

Parameter	Beschreibung
Freeze-Daten bei Verbindungsverlust [ms]	<p>Verhalten der Input Variablen dieser safeethernet Verbindung bei Verbindungsunterbrechung¹⁾.</p> <ul style="list-style-type: none"> • Verwende Initialdaten: Für die Input Variablen werden die Initialdaten verwendet. Unbegrenzt Die Input Variablen werden auf dem momentanen Wert eingefroren und bis zur erneuten Verbindungsaufnahme verwendet. • Begrenzt Eingabe: Doppelklick auf Dropdown-Feld und Zeit eingeben. Die Input Variablen werden auf dem momentanen Wert eingefroren und bis nach dem parametrisierten Timeout verwendet. Danach werden die Initialdaten verwendet. Der Timeout kann sich um bis zu einem CPU Zyklus verlängern.
Fragmente pro Zyklus	<p>Feste Einstellung: Ein Fragment wird pro Zyklus der Steuerung zum Kommunikationspartner übertragen.</p> <p>Fragment ≤ 900 Byte</p>
Priorität Ereignisse	Funktion wird nicht unterstützt.
Priorität Zustandswerte	
Anzahl ignorierte Warnungen	Ist die Anzahl von Warnungen, die hintereinander in der Zeitspanne Zeitraum Warnungen [ms] auftreten müssen, bis diese in die Diagnose oder in die Kommunikations-Fehlerstatistik eingehen.
Zeitraum Warnungen [ms]	0 ms ist der derzeit einzig zugelassene Wert.
SER aktivieren	Standardwert: deaktiviert

1) Beachten Sie den folgenden Warnhinweis:

⚠ WARNUNG

Verhalten der Input-Variablen bei Verbindungsunterbrechung.
Tod oder schwere Körperverletzung.



Objektauswahl

Die Objektauswahl stellt alle Ressourcen innerhalb dieses Projektes zur Verfügung, mit denen diese Ressource über safeethernet verbunden werden kann.

HINWEIS



Für safeethernet-Verbindungen zu Ressourcen außerhalb eines Projektes steht eine Exportfunktion zur Verfügung (siehe Kapitel "Projektübergreifende Kommunikation").

4.3 Detailansicht des safeethernet-Editors

Die Detailansicht hat immer den Bezug auf die lokale Ressource, für die Sie den safeethernet-Editor gestartet haben.

So öffnen Sie die Detailansicht einer safeethernet-Verbindung:

1. Mit Rechtsklick auf [safeethernet Verbindung] Kontextmenü öffnen.
2. Auf [Detailansicht] klicken.

Die Detailansicht beinhaltet das Register Systemvariablen, Fragment-Definitionen und Ressource (lokal) ↔ Ressource (Ziel).

4.3.1 Register: Systemvariablen

Sie können die safeethernet-Verbindung im Anwenderprogramm mit Hilfe von Systemvariablen steuern und deren Status auswerten.

Systemvariable	Beschreibung	
Ack-Frame-Nr.	Empfangszähler (Umlaufend).	
Anzahl defekter Nachrichten	Anzahl aller defekter Nachrichten pro Kanal (falscher CRC, falscher Header, sonstige Fehler).	
Anzahl defekter Nachrichten des Red. Kanal		
Anzahl Verbindungserfolge	Anzahl der Verbindungserfolge seit Reset der Statistik.	
Anzahl verlorener Nachrichten	Anzahl der auf einem der beiden Transportwege ausgefallenen Nachrichten seit Reset der Statistik. Der Zähler wird nur bis zum Komplettausfall eines Kanals geführt.	
Anzahl verlorener Nachrichten des Red.-Kanal		
Early Queue Usage	Anzahl der Nachrichten die in Early Queue gelegt wurden seit Reset der Statistik, siehe auch Kapitel "Speicher".	
Fehlerhafte Nachrichten	Anzahl verworfener Nachrichten seit Reset der Statistik.	
Frame-Nr.	Sendungszähler (Umlaufend).	
Kanalzustand	Aktueller Kanalzustand von Kanal 1. Der Kanalzustand ist der aktuelle Zustand des Kanal 1 zum Zeitpunkt (Seq-No X-1) beim Empfang einer Nachricht mit Seq-No X.	
	Status	Beschreibung
	0	Keine Nachricht zum Zustand von Kanal 1.
	1	Kanal 1 OK.
	2	Letzte Nachricht war Fehlerhaft, aktuelle ist OK.
	3	Fehler auf Kanal 1.
Layoutversion	Signatur des in der Kommunikation verwendeten Datenlayouts.	

Systemvariable	Beschreibung		
Letzte Kanal Latenz	Die Kanal-Latenz gibt die Verzögerung zwischen beiden redundanten Transportpfaden zum Empfangszeitpunkt von Nachrichten mit identischer SeqNo an. Hierfür wird eine Statistik mit durchschnittlicher, minimaler, maximaler und letzter Latenz geführt. Ist der Min-Wert > Max-Wert, so sind die Statistikwerte ungültig. Letzte Kanal-Latenz und Mittlere Kanal-Latenz sind dann 0.		
Letzte Latenz des Red.-Kanal			
Max. Kanal Latenz			
Max. Kanal Latenz des Red. Kanal			
Min. Kanal Latenz			
Min. Kanal Latenz des Red. Kanal			
Mittlere Kanal Latenz			
Mittlere Kanal Latenz des Red. Kanal			
Monotonie	Nutzdatensendungszähler (Umlaufend).		
Neue Layoutversion	Signatur des neuen Datenlayouts.		
Qualität Kanal 1	Zustand des Haupt-Transportweges.		
	Bit Nr.	Bit = 0	Bit = 1
	0	Transportweg nicht freigegeben	Transportweg freigegeben
	1	Transportweg nicht genutzt	Transportweg aktiv genutzt
	2	Transportweg nicht verbunden	Transportweg verbunden
	3	-	Transportweg liefert Nachricht zuerst
	4–7	Reserviert	Reserviert
Qualität Kanal 2	Zustand des redundanten Transportweges, siehe Zustand Kanal 1 (Haupt-Transportweg).		
Receive Timeout	Zeit in Millisekunden (ms) auf PES1, innerhalb der eine gültige Antwort von PES2 empfangen werden muss, siehe auch Kapitel "Receive Timeout".		
Response Time	Zeit in Millisekunden (ms) bis zur Empfangsbestätigung einer Nachricht beim Absender, siehe auch Kapitel "Response Time".		
safeethernet-Statistik Zurücksetzen	Statistikwerte für die Kommunikationsverbindung im Anwenderprogramm zurücksetzen (z. B. Anzahl defekter Nachrichten, Kanalzustand, Zeitstempel des letzten Fehlers des Red.-Kanal ..., Wiederholungen).		
	Wert	Funktion	
	0	Kein Reset	
	1–255	Reset der safeethernet-Statistik	
Transport-Steuerung Kanal1	Transportsteuerung von Kanal1.		
	Bit 0	Funktion	
	FALSE	Transportweg für Tests freigegeben	
	TRUE	Transportweg gesperrt	
	Bit 2 – 7 reserviert.		
Transport-Steuerung Kanal2	Siehe Transportsteuerung Kanal 1.		

Systemvariable	Beschreibung	
Verbindungssteuerung	Mit dieser Systemvariablen kann die safeethernet-Verbindung vom Anwenderprogramm gesteuert werden.	
	Befehl	Beschreibung
	Autoconnect (0x0000)	Standardwert: Nach Verlust der safeethernet Kommunikation versucht die Steuerung im nächsten CPU-Zyklus, die Verbindung wieder aufzunehmen.
	Toggle Mode 0 (0x0100) Toggle Mode 1 (0x0101)	Nach dem Kommunikationsverlust kann durch einen programmgesteuerten Wechsel des Toggle-Modus die Verbindung erneut aufgebaut werden. <ul style="list-style-type: none"> TOGGLE MODE_0 (0x100) gesetzt: Auf TOGGLE MODE 1 (0x101) setzen um die Verbindung wieder aufzunehmen. TOGGLE MODE 1 (0x101) gesetzt: Auf TOGGLE_MODE_0 (0x100) setzen um die Verbindung wieder aufzunehmen.
	Disabled (0x8000)	safeethernet-Kommunikation abgeschaltet.
Verbindungszustand	Der Verbindungszustand wertet den Status der Kommunikation zwischen zwei Steuerungen im Anwenderprogramm aus.	
	Status/Wert	Beschreibung
	Closed (0)	Verbindung ist geschlossen und es wird auch nicht versucht sie zu öffnen.
	Try_open (1)	Verbindung wird versucht zu öffnen, sie ist jedoch noch nicht geöffnet. Dieser Zustand gilt gleichermaßen für die aktive und auch für die passive Seite.
	Connected (2)	Die Verbindung ist hergestellt und in Betrieb (aktive Zeitüberwachung und Datenaustausch).
Wiederholungen	Anzahl der Wiederholungen seit Reset der Statistik.	
Zeitstempel des letzten Fehlers des Red.-Kanal [ms]	Millisekunden Anteil des Zeitstempels (aktuelle Systemzeit).	
Zeitstempel des letzten Fehlers des Red.-Kanals [s]	Sekunden Anteil des Zeitstempels (aktuelle Systemzeit).	
Zeitstempel des letzten Fehlers [ms]	Millisekunden Anteil des Zeitstempels (aktuelle Systemzeit).	
Zeitstempel des letzten Fehlers [s]	Sekunden Anteil des Zeitstempels (aktuelle Systemzeit).	
Zustand des Red.-Kanal	Aktueller Kanalzustand von Kanal 2. Der Kanalzustand ist der aktuelle Zustand des Kanal 2 zum Zeitpunkt (Seq-No X-1) beim Empfang einer Nachricht mit Seq-No X.	
	Status	Beschreibung
	0	Keine Nachricht zum Zustand von Kanal 2
	1	Kanal 2 OK
	2	Letzte Nachricht war Fehlerhaft, aktuelle ist OK.

Systemvariable	Beschreibung	
	3	Fehler auf Kanal 2.

4.4 safeethernet Parameter

Die sicherheitsgerichtete Kommunikation richten Sie im safeethernet-Editor ein. Dazu müssen Sie die in diesem Kapitel beschriebenen Parameter parametrieren. Für die Berechnung der safeethernet Parameter *Receive Timeout* und *Response Time* gilt folgende Bedingung: Die Kommunikations-Zeitscheibe muss ausreichend groß sein, um in einem CPU-Zyklus alle safeethernet Verbindungen abzuarbeiten, siehe Kapitel "Maximale Kommunikationsscheibe"

4.4.1 Maximale Zykluszeit der Sicherheitssteuerung

Zur Bestimmung der maximalen Zykluszeit für eine Sicherheitssteuerung MOVISAFE® HM31 empfiehlt SEW-EURODRIVE die folgende Vorgehensweise.

Maximale Zykluszeit der Sicherheitssteuerung MOVISAFE® HM31 bestimmen:

1. System unter voller Last betreiben. Dabei müssen alle Kommunikationsverbindungen in Betrieb sein, sowohl über **safeethernet** als auch über Standardprotokolle. Die Zykluszeit im Control Panel öfter ablesen, und die maximale Zykluszeit notieren.
2. Schritt 1 für den Kommunikationspartner (zweite Sicherheitssteuerung) wiederholen.
3. Die größere der beiden ermittelten maximalen Zykluszeiten ist die gesuchte maximale Zykluszeit.

Die maximale Zykluszeit ist ermittelt und geht in die nachfolgenden Berechnungen ein.

4.4.2 Receive Timeout

ReceiveTMO ist die Überwachungszeit in Millisekunden (ms), innerhalb der eine korrekte Antwort des Kommunikationspartners empfangen werden muss.

Trifft innerhalb der *ReceiveTMO* keine korrekte Antwort des Kommunikationspartners ein, wird die sicherheitsgerichtete Kommunikation geschlossen. Die Input Variablen dieser **safeethernet** Verbindung verhalten sich gemäß dem eingestellten Parameter *Freeze-Daten bei Verbindungsverlust [ms]*.

Für sicherheitsgerichtete Funktionen, die über **safeethernet** realisiert werden, darf nur die Einstellung *Verwende Initialdaten* benutzt werden.

Da die *ReceiveTMO* sicherheitsrelevant und Bestandteil der Worst Case Reaction Time T_R (maximale Reaktionszeit, siehe Sicherheitshandbuch Kapitel 8.2.4) ist, muss die *ReceiveTMO* wie folgt berechnet und im **safeethernet** Editor eingetragen werden:

$$\text{ReceiveTMO} \geq 4 \times \text{Delay} + 5 \times \text{max. Zykluszeit}$$

Bedingung: Die Kommunikations-Zeitscheibe muss ausreichend groß sein, um in einem CPU-Zyklus alle **safeethernet** Verbindungen abzuarbeiten.

Delay: Verzögerung auf der Übertragungsstrecke, z.B. durch Switch, Satellit

Max. Zykluszeit: maximale Zykluszeit der beiden Steuerungen

HINWEIS



- Eine erwünschte Fehlertoleranz der Kommunikation kann über eine Erhöhung der *ReceiveTMO* erreicht werden, sofern dies für den Anwendungsprozess zeitlich zulässig ist.
- Der maximal zulässige Wert für *ReceiveTMO* hängt vom Anwendungsprozess ab und wird im **safeethernet**-Editor zusammen mit der maximal zu erwartenden *Response Time* und dem Profil eingestellt.

4.4.3 Response Time

Die *ResponseTime* ist die Zeit in Millisekunden (ms), die verstreicht, bis der Absender einer Nachricht die Empfangsbestätigung des Empfängers erhält.

Für die Parametrierung unter Verwendung eines **safeethernet** Profils muss eine durch die physikalischen Gegebenheiten der Übertragungsstrecke erwartete *ResponseTime* vorgegeben werden.

Die vorgegebene *ResponseTime* hat Einfluss auf die Konfiguration aller Parameter der **safeethernet** Verbindung, die wie folgt zu berechnen sind:

$$\text{ResponseTime} \leq \text{ReceiveTMO} / n$$

$$n = 2, 3, 4, 5, 6, 7, 8, \dots$$

Das Verhältnis der *ReceiveTMO* und der *ResponseTime* beeinflusst die Fähigkeit zur Fehlertoleranz, z. B. bei Paketverlusten (Wiederholung von verloren gegangenen Datenpaketen) oder Verzögerungen auf dem Übertragungsweg.

In einem Netzwerk, in dem es zu Paketverlusten kommen kann, muss die folgende Bedingung erfüllt sein:

$$\text{Min. Response Time} \leq \text{ReceiveTMO} / 2 \geq 2 \times \text{Delay} + 2,5 \times \text{max. Zykluszeit}$$

Ist diese Bedingung erfüllt, kann der Verlust wenigstens eines Datenpaketes abgefangen werden, ohne dass die **safeethernet** Verbindung unterbrochen wird.

HINWEIS



- Ist diese Bedingung nicht erfüllt, kann die Verfügbarkeit einer **safeethernet** Verbindung nur in einem kollisions- und störungsfreien Netzwerk garantiert werden. Dies bedeutet jedoch kein Sicherheitsproblem für das Prozessormodul!
- Es ist sicherzustellen, dass das Kommunikationssystem die parametrierte *Response-Time* einhält! Für Fälle, in denen dies nicht immer garantiert werden kann, steht zur Überwachung der *Response-Time* eine entsprechende Systemvariable der Verbindung zur Verfügung. Kommt es nicht nur in seltenen Einzelfällen zu einer Überschreitung der gemessenen *Response-Time* über die halbe *ReceiveTMO*, muss die parametrierte *Response Time* erhöht werden. Die *Receive Timeout* ist der neu parametrierten *Response-Time* anzupassen.
- In den folgenden Beispielen gelten die Formeln für die Berechnung der maximalen Reaktionszeit im Fall einer Verbindung mit der Sicherheitssteuerung nur dann, wenn auf diesen die Sicherheitszeit = $2 \times \text{Watchdog-Zeit}$ eingestellt ist.

4.4.4 Sync/Async

Sync: Zurzeit nicht unterstützt.

Async: Ist die Standardeinstellung. Bei der Einstellung Async empfängt die **safeethernet** Protokolleinstanz in der Input-Phase der CPU und sendet gemäß ihren Senderegeln in der Output-Phase der CPU.

4.4.5 Resend Timeout

ResendTMO kann nicht manuell eingegeben werden, sondern wird aus dem Profil und der *Response-Time* berechnet. Überwachungszeit in Millisekunden (ms) auf PES1, innerhalb welcher PES2 den Empfang eines Datenpaketes bestätigt haben muss, ansonsten wird das Datenpaket wiederholt.

Regel: $ResendTMO \leq Receive-Timeout$

Bei unterschiedlicher Konfiguration der Resend-Timeout bei den Kommunikationspartnern bestimmt der aktive Protokollpartner (kleinere SRS) den tatsächlichen Wert der Resend-Timeout der Protokollverbindung.

4.4.6 Acknowledge Timeout

AckTMO kann nicht manuell eingegeben werden, sondern wird aus dem Profil und der *Response-Time* berechnet. *AckTMO* ist die Zeit, nach der ein empfangenes Datenpaket von der CPU spätestens bestätigt werden muss.

Für ein schnelles Netzwerk ist *AckTMO* null, d. h. der Empfang eines Datenpaketes wird sofort bestätigt. Für ein langsames Netzwerk (z. B. Telefonmodemstrecke) ist *AckTMO* größer null. In diesem Fall wird versucht, die Bestätigungsmeldung zusammen mit Prozessdaten zu übermitteln, um die Netzbelastung durch Vermeidung von Adressierungs- und Sicherungsblöcken zu reduzieren.

Regeln:

AckTMO muss $\leq Receive-Timeout$ sein

AckTMO muss $\leq Resend-Timeout$ sein, wenn *Production-Rate* $> Resend-Timeout$ ist.

4.4.7 Production Rate

ProdRate kann nicht manuell eingegeben werden, sondern wird aus dem Profil und der *Response-Time* berechnet.

Kleinstes Zeitintervall in Millisekunden (ms) zwischen zwei Datenpaketen.

Das Ziel von *ProdRate* ist, die Menge an Datenpaketen auf ein Maß zu begrenzen, welches einen (langsamen) Kommunikationskanal nicht überlastet. Dadurch wird eine gleichmäßige Auslastung des Übertragungsmediums erreicht und der Empfang veralteter Daten auf der Empfängerseite vermieden.

Regeln:

- $ProdRate \leq Receive-Timeout$
- $ProdRate \leq Resend-Timeout$, wenn *Acknowledge-Timeout* $> Resend-Timeout$

HINWEIS



Eine Production Rate von null bedeutet, dass mit jedem Zyklus des Anwenderprogramms Datenpakete übertragen werden können.

4.4.8 Speicher

Speicher kann nicht manuell eingegeben werden, sondern wird aus dem Profil und der *Response-Time* berechnet.

Speicher (Queue-Tiefe) ist die Anzahl der Datenpakete, die ausgesendet werden können, ohne auf deren Empfangsbestätigung warten zu müssen.

Der Wert ist abhängig von der Übertragungskapazität des Netzwerkes und möglichen Verzögerungen durch Netzwerklaufzeiten.

Alle safeethernet Verbindungen teilen sich den zur Verfügung stehenden Message-Speicher in der CPU.

4.5 Maximale Reaktionszeit für safeethernet

In den folgenden Beispielen gelten die Formeln für die Berechnung der maximalen Reaktionszeit nur dann, wenn die Sicherheitszeit = $2 \times$ Watchdog-Zeit eingestellt ist.

HINWEIS



Die zulässige maximale Reaktionszeit ist abhängig vom Prozess und ist mit der abnehmenden Prüfstelle abzustimmen.

Begriff	Bedeutung
ReceiveTMO	Überwachungszeit im PES 1, in der eine gültige Antwort vom PES 2 empfangen werden muss. Nach Ablauf der Zeit wird die sicherheitsgerichtete Kommunikation andernfalls geschlossen.
Production Rate	Mindestabstand zwischen zwei Datensendungen.
Watchdog-Zeit	Maximal erlaubte Dauer eines RUN-Zyklus in einer Steuerung. Die Dauer des RUN-Zyklus hängt von Komplexität des Anwenderprogramms und der Anzahl der safeethernet Verbindungen ab. Watchdog-Zeit (WDZ) ist in den Eigenschaften der Ressource einzutragen.
Worst Case Reaction Time	Maximale Reaktionszeit für die Übertragung der Änderung des Zustands eines physikalischen Einganges (In) eines PES 1 bis zur Änderung des physikalischen Ausgangs (Out) eines PES 2.
Delay	Verzögerung einer Übertragungsstrecke z. B. bei Modem- oder Satellitenverbindung. Bei direkter Verbindung kann zunächst eine Verzögerung von 2 ms angenommen werden. Die tatsächliche Verzögerung der Übertragungsstrecke kann von dem zuständigen Netzwerkadministrator ausgemessen werden.

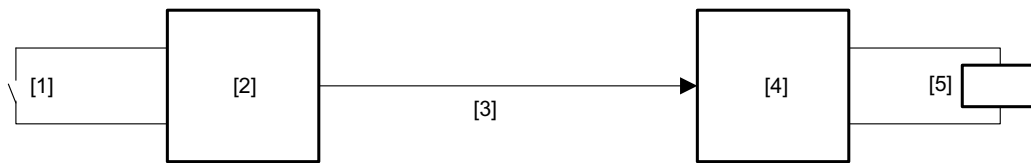
Für die folgenden Berechnungen der zulässigen maximalen Reaktionszeiten gelten folgende Bedingungen:

- Die Variablen, die mit safeethernet übertragen werden, müssen in den jeweiligen Steuerungen innerhalb eines CPU-Zyklus verarbeitet werden.
- Die Reaktionszeiten der Sensoren und Aktoren sind zusätzlich zu addieren.

Die Berechnungen gelten auch für Variablen in umgekehrter Richtung.

4.5.1 Berechnung der maximalen Reaktionszeit

Die maximale Reaktionszeit T_R (Worst Case) vom Wechsel eines Eingangs des PES 1 bis zur Reaktion des Ausgangs des PES 2 kann wie folgt berechnet werden:



4784751883

- [1] Eingang
- [2] Sicherheitssteuerung PES 1
- [3] Sicherheitsgerichtetes Protokoll
- [4] Sicherheitssteuerung PES 2
- [5] Ausgang

$$T_R = t_1 + t_2 + t_3$$

- T_R Worst Case Reaction Time
- t_1 2 × Watchdog-Zeit der Sicherheitssteuerung 1
- t_2 ReceiveTMO
- t_3 2 × Watchdog-Zeit der Sicherheitssteuerung 2

Die maximale Reaktionszeit ist abhängig vom Prozess und mit der abnehmenden Prüfstelle abzustimmen.

4.5.2 safeethernet-Profile

safeethernet-Profile sind Kombinationen zueinander passender Parameter, die automatisch bei Auswahl eines der safeethernet Profile eingestellt werden. Für die Parametrierung muss nur die *Receive-Timeout* und die erwartete *Response-Time* einzeln konfiguriert werden.

Das Ziel eines safeethernet-Profiles besteht darin, den Datendurchsatz im Netzwerk unter Berücksichtigung der physikalischen Gegebenheiten zu optimieren.

Voraussetzung für die Wirksamkeit der Optimierung sind die nachfolgenden Bedingungen:

- Kommunikations-Zeitscheibe muss ausreichend groß sein, damit in einem CPU-Zyklus alle safeethernet Verbindungen abgearbeitet werden.
- Mittlere CPU Zykluszeit < *Response-Time*
- Mittlere CPU Zykluszeit < *ProdRate* oder *ProdRate* = 0

ACHTUNG



Unpassende Kombinationen von CPU-Zyklus, Kommunikations-Zeitscheibe, *Response-Time* und *ProdRate* werden bei der Codegenerierung und beim Download/Reload nicht abgelehnt. Diese Kombinationen können aber zu Störungen bis hin zum Ausfall der safeethernet-Kommunikation führen.

Mögliche Beschädigung des Antriebssystems.

- In den Control Panels der beiden Steuerungen die Anzeigen "**Fehlerhafte Nachrichten**" und "**Wiederholungen**" überprüfen.

Sechs safeethernet-Profilen stehen zur Verfügung, aus denen das für die Übertragungsstrecke geeignete safeethernet-Profil ausgewählt werden kann. Beachten Sie dazu den folgenden Warnhinweis:



▲ WARNUNG

Lediglich die **Noisy-Profile** sind für sicherheitsgerichtete Prozessdatenkommunikation geeignet!

Tod oder schwere Körperverletzung.

- Geeignete safeethernet-Profile: Fast & Noisy, Medium & Noisy und Slow & Noisy.

Die folgende Tabelle zeigt Ihnen die verfügbaren safeethernet-Profile:

safeethernet-Profil	Verwendung
Fast & Cleanroom	Nur für störungsfreies Netzwerk empfohlen.
Fast & Noisy	Empfohlen, für eine hohe Verfügbarkeit der safeethernet-Verbindung.
Medium & Cleanroom	Nur für störungsfreies Netzwerk empfohlen.
Medium & Noisy	Empfohlen, für eine hohe Verfügbarkeit der safeethernet-Verbindung.
Slow & Cleanroom	Nur für störungsfreies Netzwerk empfohlen.
Slow & Noisy	Empfohlen, für eine hohe Verfügbarkeit der safeethernet-Verbindung.

4.5.3 Profil I (Fast & Cleanroom)



▲ WARNUNG

Lediglich die **Noisy-Profile** sind für sicherheitsgerichtete Prozessdatenkommunikation geeignet!

Tod oder schwere Körperverletzung.

- Geeignete safeethernet-Profile: Fast & Noisy, Medium & Noisy und Slow & Noisy.

Verwendung

Das Profil **Fast & Cleanroom** ist geeignet für Anwendungen in idealer Umgebung, z. B. Labor.

- Für schnellsten Datendurchsatz.
- Für Anwendungen, die eine schnelle Datenübermittlung erfordern.
- Für Anwendungen, die eine möglichst geringe Worst Case ReactionTime erfordern.
- Fast: 100-Mbit-Technologie (100 Base TX), 1-Gbit-Technologie.
- Clean: Störungsfreies Netzwerk.
- Datenverlust durch Netzüberlastung, Einflüsse von außen oder Netzwerkmanipulationen müssen vermieden werden.
- LAN-Switches erforderlich!

Netzwerkanforderungen

Charakteristika des Kommunikationspfads	<ul style="list-style-type: none"> • Minimale Verzögerungen. • Erwartete ResponseTime \leq ReceiveTMO (anderenfalls FEHLER bei Parametrierung)
--	---

4.5.4 Profil II (Fast & Noisy)

Verwendung	<p>Das Profil Fast & Noisy ist das SILworX®-Standardprofil für die Kommunikation über safeethernet.</p> <ul style="list-style-type: none"> • Für schnellsten Datendurchsatz. • Für Anwendungen, die eine schnelle Datenübermittlung erfordern. • Für Anwendungen, die eine möglichst geringe Worst Case ReactionTime erfordern.
Netzwerkanforderungen	<ul style="list-style-type: none"> • Fast: 100-Mbit-Technologie (100 Base TX), 1-Gbit-Technologie. • Noisy: Netzwerk ist nicht störungsfrei. Geringe Wahrscheinlichkeit für Verlust von Datenpaketen. Zeit für ≥ 1 Wiederholung. • LAN-Switches erforderlich!
Charakteristika des Kommunikationspfads	<ul style="list-style-type: none"> • Minimale Verzögerungen. • Erwartete ResponseTime \leq ReceiveTMO / 2 (anderenfalls FEHLER bei Parametrierung)

4.5.5 Profil III (Medium & Cleanroom)



▲ WARNUNG

Lediglich die **Noisy-Profile** sind für sicherheitsgerichtete Prozessdatenkommunikation geeignet!

Tod oder schwere Körperverletzung.

- Geeignete safeethernet-Profile: Fast & Noisy, Medium & Noisy und Slow & Noisy.

Verwendung	<p>Das Profil Medium & Cleanroom ist für Anwendungen in einem störungsfreien Netzwerk, die eine nur mäßig schnelle Datenübermittlung erfordern.</p> <ul style="list-style-type: none"> • Für mittleren Datendurchsatz. • Geeignet für Virtual Private Networks (VPN), in denen der Datenaustausch durch zwischengeschaltete Sicherheitseinrichtungen (Firewalls, Verschlüsselung) langsam, aber fehlerfrei ist. • Geeignet für Anwendungen, in denen die Worst Case ReactionTime kein kritischer Faktor ist.
Netzwerkanforderungen	<ul style="list-style-type: none"> • Medium: 10-Mbit- (10 Base T), 100-Mbit- (100 Base TX), 1-Gbit-Technologie • LAN-Switches erforderlich! • Clean: Störungsfreies Netzwerk. Datenverlust durch Netzüberlastung, Einflüsse von außen oder Netzwerkmanipulationen müssen vermieden werden. Zeit für ≥ 0 Wiederholungen.

- | | |
|---|---|
| Charakteristika
des Kommunikati-
onspfads | <ul style="list-style-type: none"> • Moderate Verzögerungen. • Erwartete ResponseTime \leq ReceiveTMO
(anderenfalls FEHLER bei Parametrierung) |
|---|---|

4.5.6 Profil IV (Medium & Noisy)

- | | |
|---|--|
| Verwendung | <p>Das Profil Medium & Noisy ist für Anwendungen, die eine nur mäßig schnelle Datenübermittlung erfordern.</p> <ul style="list-style-type: none"> • Für mittleren Datendurchsatz. • Für Anwendungen, die nur eine mäßig schnelle Datenübermittlung erfordern. • Geeignet für Anwendungen, in denen die Worst Case ReactionTime kein kritischer Faktor ist. |
| Netzwerkanforde-
rungen | <ul style="list-style-type: none"> • Medium: 10-Mbit- (10 Base T), 100-Mbit- (100 Base TX), 1-Gbit-Technologie • LAN-Switches erforderlich! • Noisy: Netzwerk ist nicht störungsfrei.
Geringe Wahrscheinlichkeit für Verlust von Datenpaketen.
Zeit für ≥ 1 Wiederholung. |
| Charakteristika
des Kommunikati-
onspfads | <ul style="list-style-type: none"> • Moderate Verzögerungen • Erwartete ResponseTime \leq ReceiveTMO / 2
(anderenfalls FEHLER bei Parametrierung) |

4.5.7 Profil V (Slow & Cleanroom)



▲ WARNUNG

Lediglich die **Noisy-Profile** sind für sicherheitsgerichtete Prozessdatenkommunikation geeignet!

Tod oder schwere Körperverschädigung.

- Geeignete safeethernet-Profile: Fast & Noisy, Medium & Noisy und Slow & Noisy.

- | | |
|---|---|
| Verwendung | <p>Das Profil Slow & Cleanroom ist für Anwendungen in einem störungsfreien Netzwerk, die nur eine langsame Datenübermittlung erfordern.</p> <ul style="list-style-type: none"> • Für langsamen Datendurchsatz. • Für Anwendungen, die nur eine langsame Datenübermittlung zu (möglicherweise weit entfernten) Steuerungen erfordern, und dort, wo die Bedingungen der Kommunikationsstrecke nicht vorhersagbar sind. |
| Netzwerkanforde-
rungen | <ul style="list-style-type: none"> • Slow: Datentransfer über ISDN, Standleitung oder Richtfunkverbindung. • Clean: Störungsfreies Netzwerk.
Datenverlust durch Netzüberlastung, Einflüsse von außen oder Netzwerkmanipulationen müssen vermieden werden.
Zeit für ≥ 0 Wiederholungen. |
| Charakteristika
des Kommunikati-
onspfads | <ul style="list-style-type: none"> • Moderate Verzögerungen. • Erwartete ResponseTime \leq ReceiveTMO
(anderenfalls FEHLER bei Parametrierung) |

4.5.8 Profil VI (Slow & Noisy)

Verwendung	<p>Das Profil Slow & Noisy ist für Anwendungen, die nur eine langsame Datenübermittlung zu (möglicherweise weit entfernten) Steuerungen erfordern.</p> <ul style="list-style-type: none"> Für langsamen Datendurchsatz. Für Anwendungen, hauptsächlich für Datentransfer über schlechte Telefonleitungen oder gestörte Richtfunkstrecken.
Netzwerkanforderungen	<ul style="list-style-type: none"> Slow: Datentransfer über Telefon, Satellit, Funk usw. Noisy: Netzwerk ist nicht störungsfrei. <p>Geringe Wahrscheinlichkeit für Verlust von Datenpaketen.</p> <p>Zeit für ≥ 1 Wiederholung.</p>
Charakteristika des Kommunikationspfads	<ul style="list-style-type: none"> Moderate bis lange Verzögerungen. Erwartete ResponseTime $\leq \text{ReceiveTMO} / 2$ (anderenfalls FEHLER bei Parametrierung)

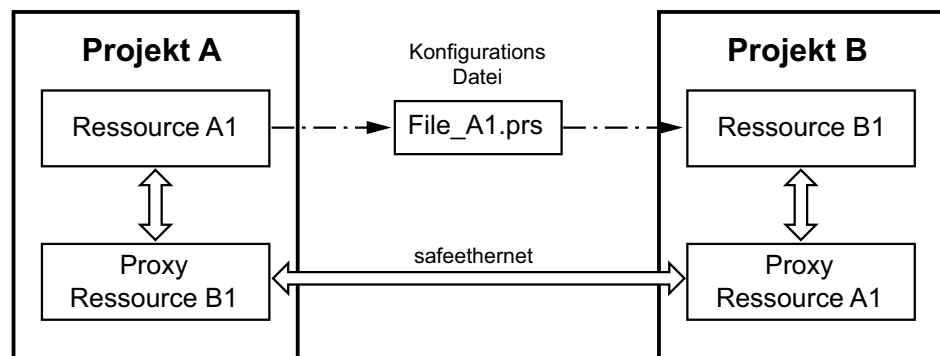
4.6 Projektübergreifende Kommunikation

Die projektübergreifende Kommunikation wird für Folgendes verwendet:

- Um Ressourcen aus verschiedenen Projekten miteinander zu verbinden.
- Um Steuerungen mit SILworX®-Betriebssystem und Steuerungen über safeethernet miteinander zu verbinden.

Die Kommunikation zwischen den beiden Projekten erfolgt über safeethernet und wird im safeethernet-Editor konfiguriert.

safeethernet-Verbindung zwischen Ressource A1 im Projekt A und der Ressource B1 im Projekt B:



5306777483

Als lokales Projekt wird das Projekt bezeichnet, in dem Sie die Konfiguration der safeethernet-Verbindung durchführen und die Konfigurationsdatei erstellen.

Als Zielprojekt wird das Projekt bezeichnet, in das Sie die Konfigurationsdatei importieren.

Beim Datenaustausch sind das lokale Projekt und das Zielprojekt gleichberechtigte Kommunikationspartner.

Die jeweilige Proxy-Ressource dient als Platzhalter für die jeweilige Ressource aus dem externen Projekt und wird für den Import und Export der safeethernet-Verbindungen genutzt.

Die *Proxy-Ressource B1* im Projekt A ist der Platzhalter der *Ressource B1* aus dem Projekt B.

Die *Proxy-Ressource A1* im Projekt B ist der Platzhalter der *Ressource A1* aus dem Projekt A.

Im lokalen Projekt (hier *Projekt A*) müssen Sie die *Proxy-Ressource* (hier *Proxy-Ressource B1*) manuell erstellen und konfigurieren. Nach der Konfiguration die Konfigurationsdatei (hier *File_A1.prs*) im Ziel-Projekt (hier von *Ressource B1*) importieren.

Die Konfigurationsdatei *File_A1.prs* enthält die komplette Beschreibung der *Ressource A1* für die safeethernet-Verbindung mit der *Ressource B1*. Nach dem Import der Konfigurationsdatei *File_A1.prs* in die *Ressource B1* wird die *Proxy-Ressource A1* automatisch im Projekt B angelegt.

4.6.1 Varianten zur projektübergreifenden Kommunikation

In den folgenden beiden Varianten kommunizieren die Projekte A und B über safeethernet miteinander.

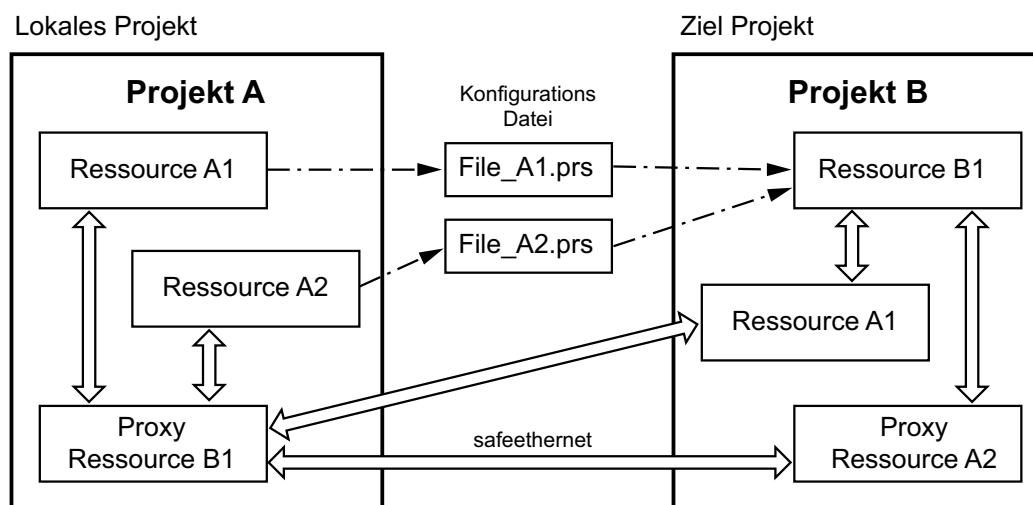
Dabei ist in der ersten Variante das Projekt A das lokale Projekt und in der zweiten Variante das Projekt B das lokale Projekt. Grundsätzlich bleibt es dem Anwender überlassen, in welchem der beiden Projekte er die Konfiguration erstellt.

Der Aufwand für beide Wege der Konfiguration ist ungefähr gleich und führt zur gleichen Konfiguration.

Lokales Projekt A

Im lokalen Projekt A konfigurieren Sie die Kommunikation zum Zielprojekt B und erstellen die Konfigurationsdateien. Das hat den Vorteil, dass Sie nur die *Proxy-Ressource B1* im lokalen Projekt manuell anlegen müssen.

Variante Projekt A als lokales Projekt:

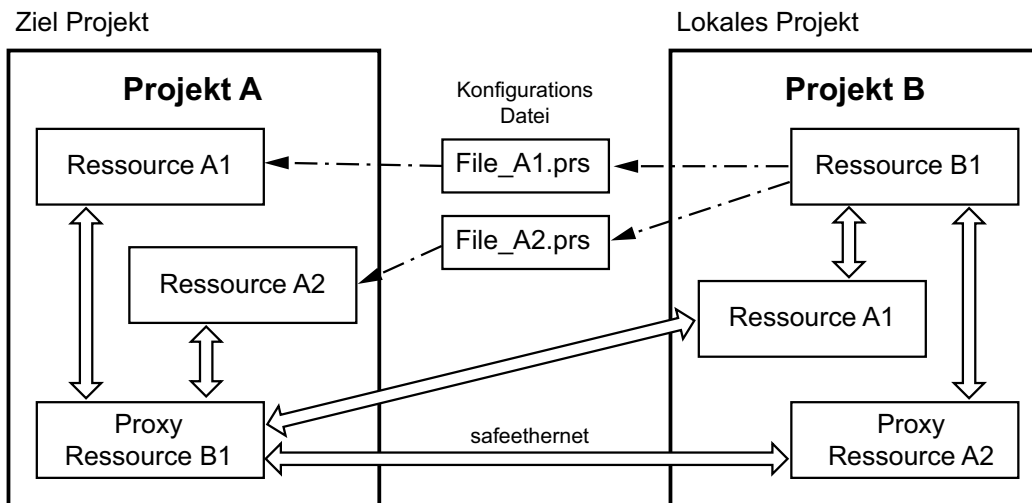


5306781963

Lokales Projekt B

Im lokalen Projekt B konfigurieren Sie die Kommunikation zum Zielprojekt A und erstellen die Konfigurationsdateien. Das hat den Nachteil, dass Sie zwei *Proxy-Ressourcen* (A1 und A2) im lokalen Projekt B manuell anlegen müssen.

Variante Projekt B als lokales Projekt:



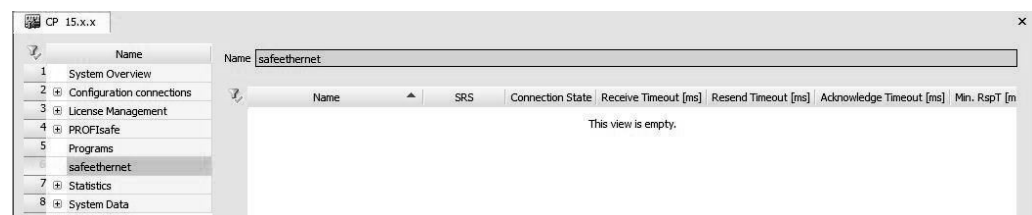
5306786827

4.7 Control Panel (safeethernet)

Im Control Panel kann der Anwender die Einstellungen der safeethernet-Verbindung überprüfen und steuern. Zudem werden aktuelle Statusinformationen (z. B. Zykluszeit, Buszustand usw.) der safeethernet-Verbindung angezeigt.

So öffnen Sie das Control Panel zur Überwachung der safeethernet-Verbindung:

1. Im Strukturbaum [Ressource] wählen.
2. Aus dem Kontextmenü der Ressource [Online] wählen.
3. Im System-Login, Zugangsdaten eingeben um das Control Panel der Ressource zu öffnen.
4. Im Strukturbaum des Control Panels [safeethernet] wählen.



5428388235

Statistikwerte zurücksetzen:

Mit der Kontextmenüfunktion können Sie die statistischen Daten (Zykluszeit min, max usw.) auf null zurücksetzen.

So setzen Sie die statistischen Daten der safeethernet-Verbindung zurück:

- Im Strukturbaum safeethernet-Verbindung selektieren.
- Wählen Sie aus dem Kontextmenü der safeethernet-Verbindung den Menüpunkt [safeethernet Statistik zurücksetzen].

4.7.1 Anzeigefeld (safeethernet-Verbindung)

In dem Anzeigefeld werden die folgenden Werte der selektierten safeethernet-Verbindung angezeigt:

Element	Beschreibung
Name	Ressourcenname des Kommunikationspartners.
SRS	System.Rack.Slot
Verbindungszustand	Zustand der safeethernet-Verbindung (siehe auch Kapitel "Detailansicht des safeethernet-Editors").
Receive-Timeout [ms]	siehe Kapitel "safeethernet-Parameter".
Resend-Timeout [ms]	siehe Kapitel "safeethernet-Parameter".
Acknowledge Timeout [ms]	siehe Kapitel "safeethernet-Parameter".
Min. RspT [ms]	Tatsächliche <i>Response-Time</i> als Minimal-, Maximal-, Letzte- und Durchschnittswert (siehe Kapitel "safeethernet-Parameter").
Max. RspT [ms]	
Letzte RspT [ms]	
Mittel RspT [ms]	
Fehlerhafte Nachrichten	Anzahl verworfener Nachrichten seit Reset der Statistik.
Wiederholungen	Anzahl der Wiederholungen seit Reset der Statistik.
Anzahl Verbindungserfolge	Anzahl der Verbindungserfolge seit Reset der Statistik.
Early Queue Usage	Anzahl der Nachrichten die in <i>Early Queue</i> gelegt wurden seit Reset der Statistik (siehe Kapitel "safeethernet-Parameter").
Frame-Nr.	Umlaufender Sendungszähler.
Ack-Frame-Nr.	Umlaufender Empfangszähler.
Monotonie	Umlaufender Nutzdatensendungszähler.
Layoutversion	Signatur des aktuellen Kommunikationsendpunkts.
Neue Layoutversion	Signatur des neuen Kommunikationsendpunkts.
Verbindungssteuerung	Status der Verbindungssteuerung.
Transport-Steuerung Kanal 1	Freigabe von Transportweg Kanal 1 (siehe Kapitel "safeethernet-Parameter").
Transport-Steuerung Kanal 2	Freigabe von Transportweg Kanal 2 (siehe Kapitel "safeethernet-Parameter").
Qualität Kanal 1	Zustand von Transportweg Kanal 1 (siehe Kapitel "safeethernet-Parameter").
Qualität Kanal 2	Zustand von Transportweg Kanal 2 (siehe Kapitel "safeethernet-Parameter").
Spät erhaltene redundante Nachrichten	Bei redundanten Transportwegen. Anzahl der verspätet empfangenen Nachrichten seit Reset der Statistik.

Element	Beschreibung
Verlorene redundante Nachrichten	Bei redundanten Transportwegen. Anzahl der auf nur einem der beiden Transportwege empfangenen Nachrichten seit Reset der Statistik.
Protokollversion	2: Neue Protokollversion für CPU-Betriebssystem ab V7.

4.8 Maximale Kommunikationszeitscheibe

Die maximale Kommunikationszeitscheibe ist die zugeteilte Zeit in Millisekunden (ms) pro Zyklus, innerhalb welcher das Prozessorsystem die Kommunikationsaufgaben abarbeitet. Können nicht alle in einem Zyklus anstehenden Kommunikationsaufgaben ausgeführt werden, erfolgt die komplette Übertragung der Kommunikationsdaten über mehrere Zyklen (Anzahl der Kommunikationszeitscheiben > 1).

HINWEIS



Es gilt die Bedingung, dass die Anzahl der Kommunikationszeitscheiben = 1 ist. Die Dauer der Kommunikationszeitscheibe ist so hoch einzustellen, dass der Zyklus die vom Prozess vorgegebene Watchdog-Zeit nicht überschreiten kann, wenn er die Kommunikationszeitscheibe ausnutzt (siehe auch Kapitel "Maximale Reaktionszeit für safeethernet").

4.9 Anschlüsse für safeethernet/Ethernet

Für die Vernetzung über safeethernet/Ethernet verfügt die Sicherheitssteuerung über die folgenden Schnittstellen:

Es stehen die folgenden Schnittstellen zur Verfügung:

- **2 Ethernet-Schnittstellen:** X4233_1 und X4233_2

Die beiden Schnittstellen befinden sich auf der Anschlussleiste des Geräts

- **1 Ethernet-Service-Schnittstelle:** X4223

Zum Anschluss eines Programmiergeräts (PADT)

Die unterschiedlichen Systeme können beliebig über Ethernet miteinander vernetzt werden (stern- oder linienförmig). Auch der Anschluss eines Programmiergeräts (PADT) ist an jeder Stelle möglich.

HINWEIS



Störungen des Ethernet-Betriebs möglich!

- Bei der Zusammenschaltung ist zu beachten, dass keine Netzringe entstehen.
- Datenpakete dürfen nur auf **einem** Weg zu einem System gelangen.

5 PROFINET IO

PROFINET-IO ist das auf der Ethernet-Technologie basierende Übertragungsprotokoll der PROFIBUS-Nutzerorganisation für die Automatisierung. Wie bei PROFIBUS-DP werden die dezentralen Feldgeräte bei PROFINET-IO über eine Gerätebeschreibung (GSDML-Datei bis V2.2) in SILworX® eingebunden.

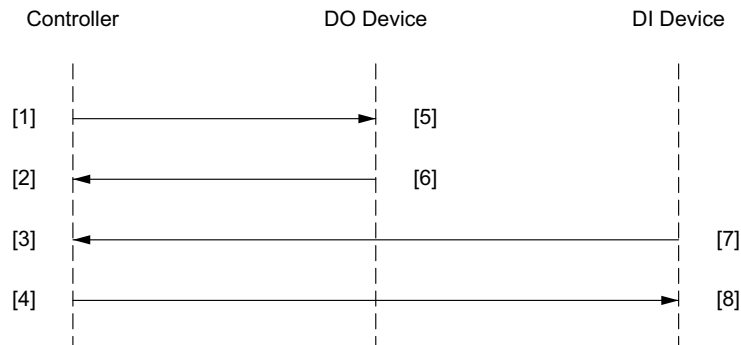
Der PROFINET-IO-Controller entspricht der Conformance Class A und unterstützt die Non-Real-Time- (NRT) und die Real-Time- (RT) Kommunikation zu den PROFINET-IO-Devices. Die Real-Time-Kommunikation wird für den zeitkritischen Datenaustausch und die Non-Real-Time-Kommunikation für nicht zeitkritische Vorgänge (z. B. azyklisches Lesen/Schreiben) verwendet.

Eine redundante PROFINET-IO-Verbindung kann nur durch die Konfiguration eines zweiten PROFINET-IO-Controller/IO-Device und Anpassungen im Anwenderprogramm erreicht werden.

5.1 Steuerung des Consumer/Provider Status (IOxS)

Mit den in diesem Kapitel beschriebenen Systemvariablen kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) gesteuert werden. Ist keine Steuerung durch das Anwenderprogramm gewünscht, so müssen die Ausgangsvariablen eines Geräts mit einer Konstanten mit dem Wert TRUE belegt werden. Dann werden die Status auf GOOD gesetzt, sobald das Kommunikationsmodul gültige Prozesswerte vom Prozessormodul erhalten hat.

Das folgende Bild zeigt den Austausch der Systemvariablen zwischen dem Controller und jeweils einem DO-Device und einem DI-Device.



12225616395

- | | |
|---|---|
| [1] Ausgangsdaten gültig | [5] Ausgangsdaten vom Controller akzeptiert |
| [2] Ausgangsdaten vom Device akzeptiert | [6] Ausgangsdaten gültig |
| [3] Eingangsdaten gültig | [7] Eingangsdaten vom Device akzeptiert |
| [4] Eingangsdaten vom Controller akzeptiert | [8] Eingangsdaten gültig |

5.1.1 Steuervariablen im Controller

Mit den Ausgangsvariablen *Ausgangsdaten gültig* [1] und *Eingangsdaten vom Controller akzeptiert* [4] kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) gesteuert werden.

Mit den Eingangsvariablen *Ausgangsdaten vom Device akzeptiert* [2] und *Eingangsdaten gültig* [3] kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) ausgelesen werden.

5.1.2 Steuervariablen im DO-Device

Mit der Ausgangsvariable *Ausgangsdaten gültig* [6] kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) gesteuert werden. Mit der Eingangsvariable *Ausgangsdaten vom Controller akzeptiert* [5] kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) ausgelesen werden.

5.1.3 Steuervariablen im DI-Device

Mit der Ausgangsvariable *Eingangsdaten vom Device akzeptiert* [7] kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) gesteuert werden. Mit der Eingangsvariable *Eingangsdaten gültig* [3] kann über das Anwenderprogramm der Consumer/Provider Status (IOxS) ausgelesen werden.

5.2 PROFIsafe

Die PROFIsafe-Spezifikation der PNO wird als bekannt vorausgesetzt. PROFIsafe benutzt das PROFINET-Protokoll zur Übertragung von sicherheitsgerichteten Daten bis SIL 3 auf Basis der Ethernet-Technologie.

Das PROFIsafe-Protokoll ist dem PROFINET-Protokoll überlagert und enthält die sicheren Nutzdaten, sowie die Informationen zur Datensicherung. Die sicheren PROFIsafe-Daten werden gemeinsam mit nicht sicherheitsrelevanten PROFINET-Daten über das unterlagerte PROFINET-Protokoll übertragen.

PROFIsafe verwendet "unsichere Datenübertragungskanäle" (Ethernet), ähnlich dem Black-Channel-Prinzip für die Übertragung von sicheren Daten. Auf diesem Weg tauschen der F-Host und das F-Device die sicheren PROFIsafe-Daten aus.

Nach der PROFIsafe-Spezifikation versendet der F-Host solange wiederholte Nachrichtenpakete, bis das F-Device eine Empfangsbestätigung an den F-Host zurückgibt. Erst dann wird vom F-Host ein neues Nachrichtenpaket an das F-Device gesendet.

Über jedes wiederholte PROFIsafe-Nachrichtenpaket wird der momentane Wert des Prozesswerts übertragen. So kann es vorkommen, dass das gleiche Prozesssignal in den wiederholten Nachrichtenpaketen unterschiedliche Werte hat.

PROFIsafe ist in SEW-Geräten empfangsseitig so ausgelegt, dass Prozesswerte nur einmalig beim ersten Empfang eines Nachrichtenpakets übernommen werden. Die Prozesswerte der wiederholten Nachrichtenpakete (mit gleicher fortlaufender Nummer des Nachrichtenpakets) werden verworfen.

Bei einem Verbindungsverlust nehmen nach Ablauf der *F_WD_Time* die PROFIsafe Prozesswert-Variablen den Initialwert ein. Damit ein bestimmter Prozesswert auf der Gegenseite (F-Host/F-Device) empfangen wird, muss der Prozesswert für mindestens die folgende Zeit unverändert anstehen:

$$2 \times F_WD_Time + F_WD_Time2$$

Das PROFIsafe-System muss vom Anwender so parametrierbar werden, dass die SFRT (Safety Function Response Time) für die jeweiligen Sicherheitsfunktionen geeignet ist (siehe Kap. "Anmerkung zu SFRT-Berechnungen").

HINWEIS



Für PROFIsafe konformes Verhalten müssen folgende Bedingungen erfüllt sein:

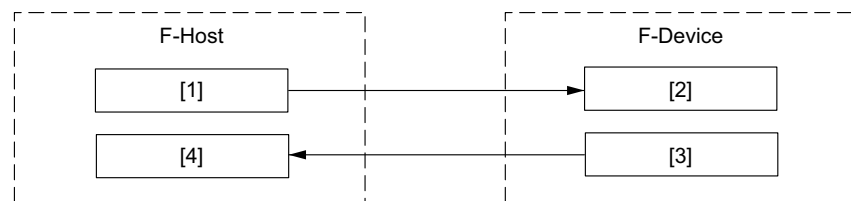
Die Standardeinstellungen in SILworX® entsprechen dem PROFIsafe konformen Verhalten.

- Die Initialwerte der Prozesswert-Variablen müssen mit "0" konfiguriert sein.
- Der Parameter *AutoAwaitFParamsOnConnLoss* muss deaktiviert sein (siehe Kapitel "Menüfunktionen des PROFINET IO-Device").

5.2.1 PROFIsafe Control-Byte und Status-Byte

Die beiden Systemvariablen *Control-Byte* und *Status-Byte* sind in jedem PROFIsafe-Submodul enthalten und werden bei der Kommunikation zwischen F-Host und F-Device ausgetauscht (siehe Kapitel "F-Parameter von Submodul Input" und Kapitel "PROFINET IO und PROFIsafe Modul").

Das PROFIsafe *Control-Byte* wird im F-Host beschrieben und im F-Device gelesen. Das PROFIsafe *Status-Byte* wird im F-Device beschrieben und im F-Host gelesen.



12214037131

[1] Control-Byte schreiben

[3] Status-Byte schreiben

[2] Control-Byte lesen

[4] Status-Byte lesen

HINWEIS



Die Systemvariablen *Control-Byte* und *Status-Byte* haben zusätzliche Funktionen, die von der PROFIsafe-Spezifikation abweichen.

5.2.2 PROFIsafe Watchdog-Zeit (F_WD_Time)

Für eine funktionierende PROFIsafe-Verbindung zwischen einem SEW F-Host und einem F-Device gilt die folgende Ungleichung:

$F_WD_Time > \text{Summe der folgenden Komponenten:}$

- $3 \times \text{CPU-Zykluszeit} \times \text{Anzahl Kommunikationszeitscheiben}$
- $2 \times \text{PROFINET-Controller-Produktionsabstand}$
- $1 \times \text{DAT (F-Device Acknowledgement Time)}$
- $2 \times \text{interne Buszeit des F-Device}$
- $2 \times \text{PROFINET-Device-Produktionsabstand}$
- $2 \times \text{Ethernet-Delay}$

PROFINET-Controller- und PROFINET-Device-Produktionsabstand sind im Allgemeinen gleich und werden folgendermaßen berechnet:

$\text{Reduktionsfaktor} \times \text{SendClockFaktor} \times 31,25 \mu\text{s}$

HINWEIS



DAT (F-Device Acknowledgement Time) und die interne Buszeit des Devices sind aus der Gerätebeschreibung des F-Device-Herstellers zu entnehmen.

Anmerkungen zur PROFIsafe Watchdog-Zeit (F_WD_Time)

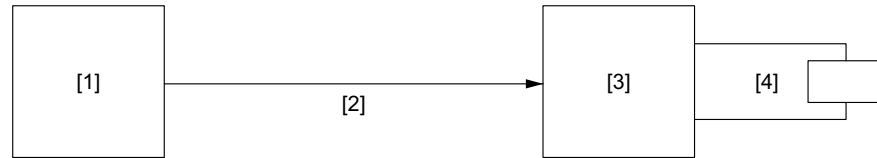
- DAT (F-Device Acknowledgement Time) ist die Zeit, die ein F-Device benötigt, um auf eine empfangene PROFIsafe-Nachricht mit einer Antwort zu reagieren. Mit F-Device ist die sichere Einheit (bei SEW-EURODRIVE das CPU-Modul) gemeint, die den F-Device-Stack ausführt. Insbesondere bei modularen Systemen/Geräten sind hier die Zeiten der nicht sicherheitsgerichteten Funktionen/Komponenten nicht enthalten. Diese Definition von DAT weicht von der PROFIsafe-Spezifikation V2.5c Kapitel 9.3.3. in den folgenden Punkten ab:
 - DAT enthält nicht die Zeiten für den internen Bus des F-Devices.
 - DAT enthält nicht den Anteil des PROFINET-Device-Produktionsabstands.
 - DAT enthält nicht die Verzögerungen, wie z. B. Filter der Eingangs-/Ausgangswerte, Verzögerungen der Ausgangs-/Eingangsphysik etc.
 - DAT steht je nach Verbindung für DATin (Eingang) oder DATout (Ausgang).
 - Für alle Zeiten sind die jeweiligen Maxima zu verwenden.
- Interne Buszeit des F-Devices ist bei der Sicherheitssteuerung:
 $(\text{Max. Anzahl Kommunikationszeitscheiben} - 1) \times \text{WDZ} - \text{CPU}$
- Voraussetzung: das F-Device arbeitet zyklisch und dessen DAT ist:
 $\text{DAT} = 2 \times \text{max. Zyklus}$
 - F-Device arbeitet **nicht mit** Kommunikationszeitscheiben. Wenn $(\text{SEW-CPU-Zykluszeit} \times \text{Anzahl-Kommunikationszeitscheiben}) < \text{F-Device-Zykluszeit}$ ist, dann muss für SEW-CPU-Zykluszeit der F_WD_Time Berechnung noch $\Delta = (\text{F-Device-Zykluszeit}) - (\text{SEW-CPU-Zykluszeit} \times \text{Anzahl der Kommunikationszeitscheiben})$ addiert werden.

 F-Device arbeitet **mit** Kommunikationszeitscheiben. Wenn $(\text{SEW-CPU-Zykluszeit} \times \text{Anzahl-Kommunikationszeitscheiben}) < \text{F-Device-Zykluszeit} \times \text{Anzahl-F-Device-Kommunikationszeitscheiben}$ ist, dann muss für SEW-CPU-Zykluszeit der F_WD_Time Berechnung noch $\Delta = (\text{F-Device-Zykluszeit} \times \text{Anzahl-Device-Kommunikationszeitscheiben}) - (\text{SEW-CPU-Zykluszeit} \times \text{Anzahl-Kommunikationszeitscheiben})$ addiert werden.

5.2.3 Safety Function Response Time (SFRT)

Berechnung der SFRT zwischen einem F-Device und einem SEW F-Host

Die zulässige SFRT für eine PROFIsafe-Verbindung zwischen einem F-Device und einem SEW F-Host mit lokaler Ausgabe wird wie folgt berechnet.



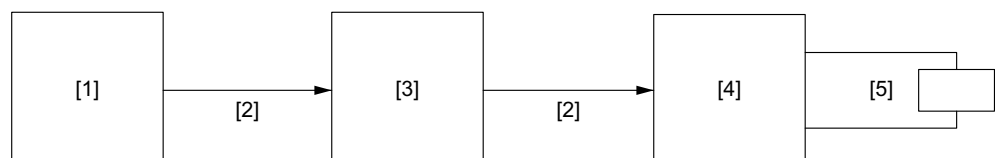
12214416523

- | | |
|-------------------------|----------------|
| [1] F-Device | [3] SEW F-Host |
| [2] PROFIsafe-Protokoll | [4] Ausgang |

$$SFRT \leq \text{MaxDataAgeIn} + 2 \times F_WD_Time + \text{MaxDataAgeOut} + T_u$$

Berechnung der SFRT mit F-Devices und einem SEW F-Host

Die zulässige SFRT für eine PROFIsafe-Verbindung zwischen einem SEW-Host und einem F-Device mit lokaler Ausgabe wird wie folgt berechnet.



12214421899

- | | |
|-------------------------|--------------|
| [1] F-Device | [4] F-Device |
| [2] PROFIsafe-Protokoll | [5] Ausgang |
| [3] SEW F-Host | |

$$SFRT \leq \text{MaxDataAgeIn} + 2 \times F_WD_Time \text{ (Input)} + 2 \times \text{WDZ-CPU} + 2 \times F_WD_TIME \text{ (Output)} + \text{MaxDataAgeOut} + T_u$$

Anmerkungen zu Safety Function Response Time-Berechnungen

1. Definition SFRT gemäß IEC 61784-3 Ed.2.
2. Alle etwaige zusätzliche Verzögerungen im Anwenderprogramm (z. B. durch Funktionsbausteine TOF, TON, etc.) oder Module/Baugruppen (z. B. durch Ausgangsfilter, Eingangsfilter, Relais etc.) müssen addiert werden.
3. *MaxDataAgeIn* ist das maximale Alter eines an einem physikalischen Eingang eingelesenen Prozesswertes, den ein F-Device in eine PROFIsafe-Nachricht einfügt. Jedoch nur der Anteil, der nicht schon in *DATin* enthalten ist.

HINWEIS



Bei der Sicherheitssteuerung hat *MaxDataAgeIn* folgende Werte:

- Für physikalische Eingänge bis zu $FTZ\text{-}CPU - 2 \times \text{WDZ} - \text{CPU} - \text{DATin}$ (FTZ: Fehlertoleranzzeit des CPU-Moduls)
- Für vom Anwenderprogramm gebildete Daten = 0 ms

4. *MaxDataAgeOut*

Dies ist die Worst-Case-Reaktionszeit eines F-Output-Devices oder F-Hosts für

- Ausgabe empfangener Prozesswerte an einem physikalischen Ausgang.
- Ansteuern der physikalischen Ausgänge nach Ablauf von F_WD_Time
- Deaktivieren der physikalischen Ausgänge bei Geräteausfall

Bei Ablauf der F_WD_TIME reagiert die Sicherheitssteuerung ohne Fehler spätestens nach $2 \times WDZ - CPU$.

- Wenn der F-Host/F-Device unmittelbar vor dieser Reaktion ausfällt, so werden bei der Sicherheitssteuerung nach $WDZ - CPU$ die Ausgänge energielos.
 - Nimmt man nur einen Fehler an, so kann $1 \times WDZ - CPU$ von $MaxDataAgeOut$ subtrahiert werden.
5. T_u ist das Minimum von DAT_{in} , DAT_{out} und $WDZ - CPU$. Theoretisch kann man für DAT_{in} und DAT_{out} jeweils die Hälfte ansetzen, dazu muss der Hersteller jedoch spezifiziert haben, mit welcher Ungenauigkeit das Device die F_WD_Time überwacht. Falls das Device zyklisch arbeitet, dann ist $DAT = 2 \times \max.$ Zyklus des Devices.
6. F_WD_TIME , siehe Kapitel "F_WD_Time (PROFIsafe Watchdog-Zeit)".

5.3 Auflagen für den sicheren Betrieb von PROFIsafe

5.3.1 Adressierung

Das PROFIsafe-Netzwerk entspricht bei SEW-EURODRIVE dem PROFINET-Ethernet-Netzwerk, über das die PROFIsafe-Nachrichten übertragen werden können. Als Netzwerk ist hier ein logisches Netzwerk gemeint, dass sich auch über mehrere physikalische Teilnetze erstrecken kann.

Eine geeignete Netzwerktrennung für PROFIsafe liegt vor, wenn es nicht möglich ist, dass PROFIsafe-Nachrichten die Netzwerktrennung überwinden können. Dies ist zum Beispiel dann der Fall, wenn ein Router auf IP-Basis verwendet wird und die Netzwerke an unterschiedlichen Ethernet-Interfaces des Routers angeschlossen sind.

Eine Netzwerktrennung für PROFIsafe ist nicht gegeben, wenn zum Beispiel die Netzwerke über Ein-Port-Router, Switches, Hubs oder Ethernet-Bridges verbunden werden.

Auch wenn managebare Switches verwendet werden und die PROFIsafe-Netzwerke zum Beispiel durch Port-Based-VLANs getrennt sind, sollte dennoch eine Eineindeutigkeit der Adressierung angestrebt werden. Dies verhindert, dass zum Beispiel bei Wartungsarbeiten oder Erweiterungen versehentlich Verbindungen zwischen PROFIsafe-Netzwerken hergestellt werden.

Folgende Punkte zur Adressierung von PROFIsafe-Geräten sind einzuhalten:

- Es muss sichergestellt werden, dass die F-Adressen der PROFIsafe-Geräte/Module in einem PROFIsafe-Netzwerk eineindeutig sind.
- Zur Adressierungssicherheit wird weiterhin empfohlen, die F-Adressen auch bei getrennten PROFIsafe-Netzwerken eineindeutig zu wählen.

- Bei der Inbetriebnahme und bei Änderungen der Sicherheitsfunktionen muss überprüft werden, dass die Sicherheitsfunktion über PROFIsafe hinweg die richtigen Ein- und Ausgänge der richtigen PROFIsafe-Geräte nutzt.
- Die PROFIsafe F-Module sind so zu konfigurieren (z. B. durch Vergabe einer geeigneten F-Adresse oder F_WD_Time), dass in einem PROFIsafe-Netzwerk F-Module gleicher Input- und gleicher Output-Datenlänge unterschiedliche CRC1-Signaturen haben. Der Anwender kann den CRC1 in SILworX® ablesen.

Dies ist dann auf jeden Fall für die betroffenen F-Module gewährleistet, wenn in einem PROFIsafe-Netzwerk nur ein F-Host genutzt wird und die F-Parameter der F-Module in einem PROFIsafe-Netzwerk sich nur in der F-Adresse unterscheiden. Damit nicht versehentlich eine gleiche CRC1-Signatur erzeugt wird, muss darauf geachtet werden, dass zum Beispiel die Parameter *F_WD_Time* und *F_Prm_Flag1/2* für alle F-Module gleich sind und die F-Module keinen i-Par-CRC verwenden.

Risiko von PROFIsafe-Geräten mit gleicher Input- und Output-Datenlänge

PROFIsafe-Geräte dürfen nur betrieben werden, wenn die F-INPUT-Datenlänge ungleich der F-OUTPUT-Datenlänge derselben PROFIsafe-Verbindung ist. Andernfalls könnten Adressierungsfehler der Standardkomponenten und/oder der Standardübertragungstechnik von PROFIsafe unentdeckt bleiben und zu sicherheitsgerichtetem Fehlverhalten führen.

Bei SEW F-Modulen, die als Teil der Sicherheitssteuerung konfiguriert werden, müssen die Längen von F-Input und F-Output unterschiedlich gewählt werden. Um das Risiko einer sicherheitsgerichteten Fehlfunktion vorzubeugen, nur F-Input oder F-Output Module verwenden. Keine F-Input-/Output-Module verwenden!

5.3.2 Netzwerkaspekte

Das für die Übertragung von PROFIsafe-Nachrichten verwendete Netzwerk muss eine ausreichende Verfügbarkeit und Übertragungsqualität aufweisen.

HINWEIS



Erkennt PROFIsafe eine verminderte Übertragungsqualität, die von den Standardübertragungseinrichtungen (Ethernet) nicht erkannt wird, erfolgt die Sicherheitsreaktion.

Nach einer Sicherheitsreaktion auf Grund verminderter Übertragungsqualität, müssen die Probleme beseitigt werden, um wieder eine ausreichende Übertragungsqualität zu gewährleisten. Erst nach den dazu erforderlichen Maßnahmen darf die Quittierung für den Wiederanlauf von PROFIsafe ausgelöst werden. Diese Quittierung erfolgt durch das Signal „Operator-Acknowledge“ oder „Reset“.

HINWEIS



„Operator-Acknowledge“ und „Reset“ dürfen nur genutzt werden, wenn keine gefährlichen Zustände mehr existieren.

Ein PROFIsafe-Netzwerk muss vor unzulässigen Eingriffen (DoS, Hacker, ...) geschützt werden. Die Maßnahmen sind mit der überwachenden Behörde abzustimmen. Dies hat besondere Relevanz, wenn Wireless-Techniken zur Übertragung eingesetzt werden. Für weitere Informationen, siehe PROFIsafe Specification V2.5c Tabelle 23 und Tabelle 24.

Verfügbarkeit gegenüber eingefügten Nachrichten

Nachrichtenpakete können z. B. durch Netzwerkkomponenten wie Switches gespeichert und zu einem späteren Zeitpunkt eingefügt (gesendet) werden. Diese Nachrichtenpakete führen zu einer Abschaltung, wenn diese älter (siehe Consecutive Number Tabelle 46 und Tabelle 63) als das zuletzt vom PROFIsafe-Gerät empfangene Nachrichtenpaket ist.

5.4 PROFINET IO-Controller und PROFIsafe F-Host

Dieses Kapitel beschreibt die Eigenschaften des PROFINET-IO-Controller und PROFIsafe F-Host sowie die Menüfunktionen und Dialoge in SILworX®, die zur Konfiguration des PROFINET-IO Controllers und PROFIsafe Host benötigt werden.

- Systemanforderungen PROFINET-IO-Controller:

Element	Beschreibung
Steuerung	Sicherheitssteuerung PFF-HM31A.
CPU-Modul	Die Ethernet-Schnittstellen des Prozessormoduls können für PROFINET-IO nicht verwendet werden.
COM-Modul	Ethernet 10/100BaseT.
Aktivierung	Die Freischaltung erfolgt per Software-Freischaltcode, siehe Kapitel "Kommunikation".

- Eigenschaften PROFINET-IO-Controller:

Eigenschaften	Beschreibung
Sicherheitsgerichtet	Nein
Übertragungsrate	100 Mbit/s Vollduplex
Transportweg	Ethernet-Schnittstellen der COM-Module. Verwendete Ethernet-Schnittstellen simultan auch für andere Protokolle nutzbar.
Konformitätsklasse	Der PROFINET-IO-Controller entspricht den Anforderungen der Conformance Class A.
Real Time Class	RT-Klasse 1
Max. Anzahl PROFINET-IO-Controller	Für jedes COM-Modul kann ein PROFINET-IO-Controller konfiguriert werden.
Max. Anzahl PROFINET-IO-Devices Applikationsbeziehungen (AR's)	Ein PROFINET-IO Controller kann mit max. 64 PROFINET-IO-Devices eine Applikationsbeziehung (AR) aufbauen.
Anzahl (CR's pro AR) Kommunikationsbeziehungen	Standard: 1 Input CR, 1 Output CR, 1 Alarm CR.
Max. Prozessdatenlänge einer Kommunikationsbeziehung (CR)	Output: max. 1440 Bytes Input: max. 1440 Bytes
Sendetakt	Möglich über die Einstellung der Reduction Rate auf der Device-Ebene.

- Eigenschaften PROFIsafe Host:

Eigenschaften	Beschreibung
Max. Anzahl F-Hosts	1012 oder 512
Max. Prozessdatenlänge einer Kommunikationsbeziehung (CR)	Output: max. 123 Bytes Nutzdaten + 5 Bytes Verwaltungsdaten Input: max. 123 Bytes Nutzdaten + 5 Bytes Verwaltungsdaten
Max. Größe Nutzdaten	512 x 123 Bytes = 62976 Bytes

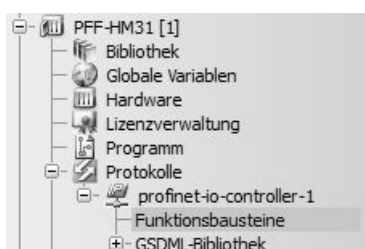
5.5 PROFINET-IO / PROFIsafe-Beispiel (Controller)

In diesem Beispiel wird beschrieben, wie auf der Sicherheitssteuerung ein PROFINET-IO-Controller konfiguriert wird, der eine Verbindung zu einem PROFINET-IO-Device hat. Das PROFINET-IO-Device ist mit einem PROFINET-IO-Modul und einem PROFIsafe-Modul bestückt. Im PROFINET-IO-Controller muss das PROFINET-IO-Device genau so konfiguriert werden, wie es tatsächlich aufgebaut ist.

5.5.1 Anlegen des PROFINET-IO-Controllers in SILworX®

Gehen Sie so vor:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle].
2. Um einen neuen PROFINET-IO-Controller hinzuzufügen, wählen Sie im Kontextmenü von Protokolle den Menüpunkt [Neu] / [PROFINET-IO Controller].
3. Wählen Sie im Kontextmenü des PROFINET-IO-Controllers den Menüpunkt [Eigenschaften].
4. Tragen Sie im Feld "Name" den Gerätenamen des Controllers ein.
5. Wählen Sie ein COM-Modul aus.



12843370507

Konfiguration des Device im PROFINET-IO-Controller

Gehen Sie so vor, um ein PROFINET-IO-Device im PROFINET-IO-Controller anzulegen:

1. Wählen Sie im Kontextmenü des PROFINET-IO-Controllers den Menüpunkt [Neu] / [PROFINET-IO-Device].

GSDML-Bibliotheksdatei aus externer Datenquelle lesen

So lesen Sie die GSDML-Bibliotheksdatei aus einer externen Datenquelle ein (z. B. CD, USB-Stick, Internet):

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [GSDML-Bibliothek].

2. Wählen Sie im Kontextmenü der GSDML-Bibliothek den Menüpunkt [Neu] und fügen Sie die zum PROFINET-IO-Device zugehörige GSDML-Datei hinzu.

GSDML-Datei für ein neues PROFINET-IO-Device laden

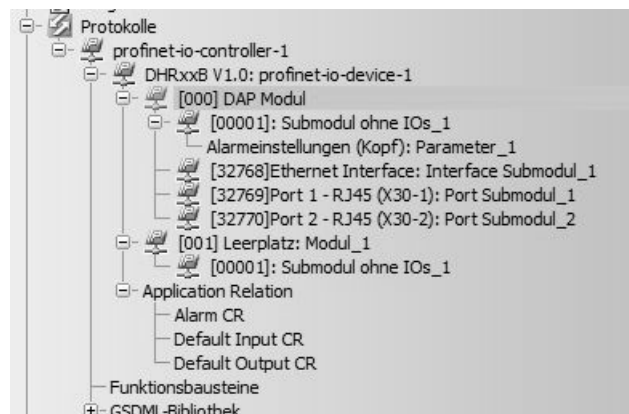
So laden Sie die GSDML-Datei für ein neues PROFINET-IO-Device:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [PROFINET-IO-Device].
2. Wählen Sie im Kontextmenü des PROFINET-IO-Device den Menüpunkt [Eigenschaften] und öffnen Sie das Register "Parameter".
3. Tragen Sie im Feld "Name" den Gerätenamen des Devices ein.
4. Geben Sie die IP-Adresse des PROFINET-IO-Device in das Feld "IP-Adresse" ein.
5. Wählen Sie im Dropdown-Menü "GSDML-Datei" die zum PROFINET-IO-Device zugehörige GSDML-Bibliotheksdatei aus. Schließen Sie das Fenster "Eigenschaften".

Device Access Point (DAP) für das PROFINET-IO-Device auswählen

So wählen Sie den Device Access Point (DAP) für das PROFINET-IO-Device:

1. Wählen Sie im Strukturbaum [Protokolle] / [PROFINET-IO-Controller] / [PROFINET-IO-Device] / [DAP Modul].
2. Rufen Sie im Kontextmenü den Menüpunkt [Edit] auf und wählen Sie den passenden Device Access Point (DAP)-Datensatz zum PROFINET-IO-Device aus.



12843376011

HINWEIS



Die GSDML-Bibliotheksdatei beinhaltet oft mehrere Device Access Points (DAP) eines Herstellers.

Modulsteckplätze konfigurieren

So konfigurieren Sie Modul-Steckplätze:

1. Öffnen Sie im Strukturbaum [Protokolle] / [PROFINET-IO-Device].
2. Um die Modul-Liste zu öffnen, wählen Sie im Kontextmenü des PROFINET-IO-Device den Menüpunkt [Neu].
3. Wählen Sie für das PROFINET-IO-Device die passenden Module aus der Modul-Liste und bestätigen Sie mit [Modul(e) einfügen].

PROFINET-IO-Device-Module nummerieren

So nummerieren Sie die PROFINET-IO-Device Module:

Das Device Access Point (DAP)-Modul hat den festen Steckplatz 0. Alle weiteren PROFINET-IO-Device Module müssen nummeriert werden.

1. Wählen Sie im Kontextmenü des PROFINET-IO-Device Modul den Menüpunkt [Eigenschaften].
2. Tragen Sie im Eingabefeld "Steckplatz" die Steckplätze der Device-Module so ein, wie das PROFINET-IO-Device tatsächlich aufgebaut ist.
3. Wiederholen Sie diesen Schritt für weitere PROFINET-IO Device Module.

Die Registerkarten "Modell" und "Features" zeigen weitere Einzelheiten der GSDML-Datei.

Application Relation konfigurieren

So konfigurieren Sie die Application Relation:

1. Öffnen Sie im Strukturbaum [PROFIsafe-IO-Device] / [Application Relation].
2. Wählen Sie im Kontextmenü von Default Input CR den Menüpunkt [Eigenschaften].
3. Passen Sie den Parameter *Reduktionsfaktor* an (z. B. auf 4 setzen).
4. Wählen Sie im Kontextmenü von Default Output CR den Menüpunkt [Eigenschaften].
5. Passen Sie den Parameter *Reduktionsfaktor* an (z. B. auf 4 setzen).

Konfiguration des Moduls Device Access Point (DAP)

Gehen Sie so vor:

1. Wählen Sie [000] DAP Modul, [xxxxx] DAP Submodul,
2. Wählen Sie im Kontextmenü von [xxxxx] DAP Submodul den Menüpunkt [Edit].
3. Rufen Sie im Dialogfenster "Edit" die Registerkarte "Systemvariablen" auf. Wenn eine Steuerung des Consumer/Provider Status mit spezieller Anwenderprogramm-Logik nicht gewünscht ist, belegen Sie die Ausgangsvariable *Eingangsdaten vom Controller akzeptiert* mit einer Globalen Variablen mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

Kopfparameter des DAP für das PROFINET-IO-Device einstellen

So stellen Sie die Kopfparameter des Device Access Point (DAP) für das PROFINET-IO-Device ein:

1. Wählen Sie im Strukturbaum [Protokolle] / [PROFINET-IO-Controller] / [PROFINET-IO-Device] / [DAP Modul] / [[xxxxx] DAP Submodul] / [Alarmeinstellungen (Kopf): Parameter].
2. Wählen si im Kontextmenü den Menüpunkt [Eigenschaften].
3. Tragen Sie im Eingabefeld "Name" den Parameternamen der Kopfparameter ein.
4. Über die Schaltfläche [Edit] wird ein Dialog geöffnet, in dem Sie Einstellungen zu Schnittstellen und Diagnose/Alarme bearbeiten können.

Konfiguration der PROFINET-IO-Device Module

HINWEIS



Die Summe der Variablen in Byte muss mit der Größe des jeweiligen Moduls in Byte übereinstimmen.

Gehen Sie zur Konfiguration des PROFINET-IO-Device Moduls so vor:

1. Wählen Sie das Submodul [[001] PROFINET-IO-Device Modul] / [[xxxxx] PROFINET-IO Device Submodul].
2. Wählen Sie im Kontextmenü von [xxxxx] Submodul den Menüpunkt [Edit].
3. Wählen Sie im Dialogfenster "Edit" die Registerkarte "Prozessvariablen".
4. Wählen Sie in der Objektauswahl die passende Variable und ziehen Sie diese per Drag & Drop in den Bereich Eingangssignale.
5. Klicken Sie mit der rechten Maustaste auf eine leere Stelle im Bereich Eingangssignale und wählen Sie im Kontextmenü den Menüpunkt [Neue Offsets], damit die Offsets der Variablen neu generiert werden.
6. Rufen Sie im Dialogfenster "Edit" die Registerkarte "Systemvariablen" auf. Wenn eine Steuerung des Consumer/Provider Status mit spezieller Anwenderprogramm-Logik nicht gewünscht ist, belegen Sie die Ausgangsvariablen *Eingangsdaten vom Controller akzeptiert* und *Ausgangsdaten gültig* mit einer Globalen Variablen mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

Konfiguration der PROFIsafe-IO-Device Module

HINWEIS



Die Summe der Variablen in Byte muss mit der Größe des jeweiligen Moduls in Byte übereinstimmen.

Gehen Sie zur Konfiguration des PROFIsafe-IO-Device Modul so vor:

1. Wählen Sie im Strukturbaum [[001] PROFIsafe-IO-Device Modul] / [[xxxxx] PROFIsafe-IO Device Submodul].
2. Wählen Sie im Kontextmenü von [xxxxx] Submodul den Menüpunkt [Edit].
3. Wählen Sie im Dialogfenster "Edit" die Registerkarte "Prozessvariablen".
4. Wählen Sie in der Objektauswahl die passende Variable und ziehen Sie diese per Drag and Drop in den Bereich Eingangssignale.
5. Klicken Sie mit der rechten Maustaste auf eine leere Stelle im Bereich Eingangssignale und wählen Sie im Kontextmenü den Menüpunkt [Neue Offsets], damit die Offsets der Variablen neu generiert werden.
6. Rufen Sie im Dialogfenster "Edit" die Registerkarte "Systemvariablen" auf. Wenn eine Steuerung des Consumer/Provider Status mit spezieller Anwenderprogramm-Logik nicht gewünscht ist, belegen Sie die Ausgangsvariablen *Eingangsdaten vom Controller akzeptiert* und *Ausgangsdaten gültig* mit einer Globalen Variablen mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

F-Parameter konfigurieren

So konfigurieren Sie die F-Parameter:

1. Wählen Sie im Strukturbaum [[001] PROFIsafe-IO-Device Modul] / [[xxxxx] PROFIsafe-IO Device Submodul] / [F-Parameter].
2. Wählen Sie im Kontextmenü von [xxxxx] Submodul den Menüpunkt [Edit].
3. Wählen Sie im Dialogfenster "Edit" die Registerkarte "Prozessvariablen".
4. Wählen Sie in der Objektauswahl die passende Variable und ziehen Sie diese per Drag and Drop in den Bereich Eingangssignale.
5. Klicken Sie mit der rechten Maustaste auf eine leere Stelle im Bereich Eingangssignale und wählen Sie im Kontextmenü den Menüpunkt [Neue Offsets], damit die Offsets der Variablen neu generiert werden.
6. Rufen Sie im Dialogfenster "Edit" die Registerkarte "Systemvariablen" auf. Wenn eine Steuerung des Consumer/Provider Status mit spezieller Anwenderprogramm-Logik nicht gewünscht ist, belegen Sie die Ausgangsvariablen *Eingangsdaten vom Controller akzeptiert* und *Ausgangsdaten gültig* mit einer Globalen Variablen mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

PROFINET-IO-Konfiguration verifizieren

1. Öffnen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller].
2. Wählen Sie im Kontextmenü des PROFINET-IO-Controller den Menüpunkt [Verifikation].
3. Prüfen Sie die Einträge im Logbuch sorgfältig und korrigieren Sie gegebenenfalls Fehler.

HINWEIS



Die Ressource neu kompilieren und in die Steuerungen laden, damit die Konfiguration für die PROFINET-IO-Kommunikation wirksam wird.

Ermitteln eines PROFINET-IO-Devices im Ethernet-Netzwerk

Um ein PROFINET-IO-Device im Ethernet-Netzwerk zu finden, gehen Sie so vor:

1. Modul-Login auf das Kommunikationsmodul mit dem PROFINET-IO-Controller ausführen.
2. Wählen Sie im Strukturbaum der Online-Ansicht [PROFINET-IO-Controller] / [PROFINET-IO Netzwerkteilnehmer].
3. Wählen Sie im Kontextmenü den Menüpunkt [PROFINET-Netzwerkteilnehmer ermitteln]. Alle PROFINET-Devices im Netzwerk dieses PROFINET-IO-Controllers werden aufgelistet.

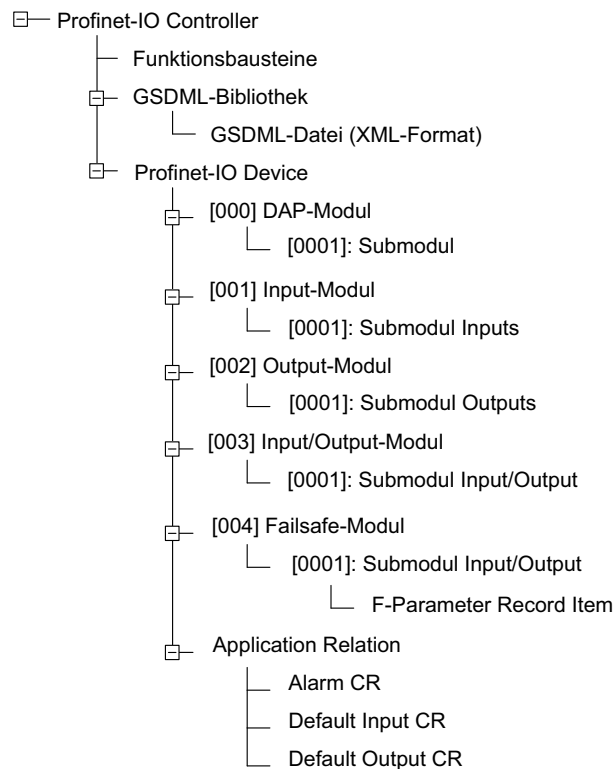
PROFINET-IO-Device in der Online-Ansicht konfigurieren

So konfigurieren Sie das PROFINET-IO-Device in der Online-Ansicht:

1. Um die Einstellungen zu ändern, rufen Sie in der Geräteliste das Kontextmenü des zu konfigurierenden PROFINET-IO Device auf.
2. Wählen Sie den Menüpunkt [Setze PROFINET-IO-Gerätenamen]. Geben Sie den Gerätenamen ein. Stellen Sie sicher, dass der PROFINET-IO-Geräte name mit der Projektierung übereinstimmt (es sind nur Kleinbuchstaben erlaubt).
3. Wählen Sie im Kontextmenü den Menüpunkt [Netzwerkeinstellungen]. Setzen Sie die IP-Adresse, Subnetzmaske und das Gateway.

HINWEIS

Der PROFINET-IO-Geräte name und die Netzwerkeinstellungen des PROFINET-IO-Device müssen im PROFINET-IO-Controller eingetragen sein, sonst ist eine Kommunikation nicht möglich.

5.6 Menüfunktionen des PROFINET-IO-Controllers**5.6.1 Beispiel für einen Strukturbaum des PROFINET-IO-Controllers**

12760425611

5.6.2 PROFINET-IO Controller

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü des PROFINET-IO Controller öffnet den Dialog „Eigenschaften“. Das Dialogfenster enthält das folgende Register.

Register PROFINET-IO (Eigenschaften)

Element	Beschreibung
Typ	PROFINET-IO Controller.
Name	Geätename des PROFINET-IO Controller.
Aktualisierungsintervall der Prozessdaten [ms]	<p>Aktualisierungszeit in Millisekunden, mit der die Daaten des Protokolls zwischen COM und CPU ausgetauscht werden. Ist die <i>Re-fresh Rate</i> Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <p>Wertebereich: 4 bis ($2^{31} - 1$)</p> <p>Standardwert: 0</p>
Prozessdaten-Konsistenz erzwingen	<ul style="list-style-type: none"> Aktiviert (Standardwert): Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU Deaktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU-Zyklen zu je 1100 Byte pro Datenrichtung. Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.
Modul	Auswahl des COM-Moduls auf dem dieses Protokoll abgearbeitet wird.
Max. µP-Budget aktivieren	<ul style="list-style-type: none"> Aktiviert: Limit des µP-Budget aus dem Feld <i>Max. µP-Budget in [%]</i> übernehmen. Deaktiviert: Kein Limit des µP-Budget für dieses Protokoll verwenden.
Max. µP-Budget in [%]	<p>Maximales µP-Budget des Moduls, das bei der Abarbeitung des Protokolls produziert werden darf.</p> <p>Wertebereich: 1 – 100 %</p> <p>Standardwert: 30 %</p>
RPC Port Server	<p>Remote Procedure Call Port</p> <p>Wertebereich: 1024 – 65535</p> <p>Standardwert: 49152</p> <p>RPC Port Server und RPC Port Client dürfen nicht identisch sein.</p>

Element	Beschreibung
RPC Port Client	Remote Procedure Call Port Wertebereich: 1024 – 65535 Standardwert: 49153 RPC Port Server und RPC Port Client dürfen nicht identisch sein.
F_Source_Add	Adresse des Controllers (F-Host). Der Anwender muss eindeutige PROFIsafe Controller-/Device-Adressen in einem PROFIsafe-Netzwerk verwenden (siehe auch IEC 61784-3-3 V2.5c Kap. 9.7)

5.6.3 PROFINET-IO Device (im Controller)

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü des PROFINET-IO Device öffnet den Dialog „Eigenschaften“, der die folgenden Register enthält.

Register Parameter (Eigenschaften)

Element	Beschreibung
Name	Geätename des PROFINET-IO Device.
IP-Adresse	IP-Adresse des Kommunikationspartners. Wertebereich: 0.0.0.0 – 255.255.255.255 Standardwert: 192.168.0.99 Keine belegten IP-Adressen verwenden (siehe Kap. „Verwendete Netzwerports für Ethernet-Kommunikation“).
Subnet Mask	Subnetzmaske des adressierten Subnetzes in dem sich das Device befindet. Wertebereich: 0.0.0.0 – 255.255.255.255 Standardwert: 255.255.255.0
GSDML-Datei	GSDML ist die Abkürzung für Generic Station Description Markup Language und ist eine auf XML Basierende Beschreibungssprache. Die GSDML-Datei enthält die Stammdaten des PROFINET-Device.

Register Modell und Features

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Herstellername*, *Gerätebeschreibung* oder *unterstützte Faktoren*. Diese Daten dienen als unterstützende Information zur Parametrierung des Geräts und können nicht geändert werden.

5.6.4 DAP-Modul (Device Access Point Modul)

Innerhalb eines PROFINET-Devices wird immer ein Modul für die Busanschaltung (DAP) angelegt. Das DAP-Modul ist Standard und kann nicht gelöscht werden.

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü des DAP-Moduls öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Register Parameter (Eigenschaften)

Element	Beschreibung
Name	Name des DAP-Moduls
Steckplatz	Nicht änderbar. Standardwert: 0

Register „Modell“
und „Features“

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Modulkennung*, *Hardware-/Software Version*. Diese Daten dienen als unterstützende Informationen zur Parametrierung des Geräts und können nicht geändert werden.

DAP-Submodul (Eigenschaften)

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü des DAP-Submoduls öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Register Parame-
ter

Element	Beschreibung
Name	Name des Input-Submoduls.
Substeckplatz	Standardwert: 1
IO Data CR, Eingänge	Auswahl der Kommunikationsbeziehung (Communication Relation CR) in der die Eingänge des Submoduls übertragen werden sollen: <ul style="list-style-type: none"> • Keine • Default: Input CR
Eingangsdaten vom Controller akzeptiert	Auswahl der Kommunikationsbeziehung (Communication Relation CR) in der der Consumer Status (CS) des Submoduls übertragen werden sollen: <ul style="list-style-type: none"> • Keine • Default: Output CR

Register „Modell“
und „Features“

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Submodulkennung*, *Datenlängen*. Diese Daten dienen als unterstützende Informationen zur Parametrierung des Geräts und können nicht geändert werden.

DAP-Submodul (Edit)

Register „Systemvariablen“

Der Menüpunkt [Edit] aus dem Kontextmenü der Input Submodule öffnet das Dialogfenster „Edit“, dass die folgenden Register enthält.

Das Register „Systemvariablen“ stellt die folgenden Systemvariablen bereit, die es erlauben, den Zustand des PROFINET-IO Submoduls im Anwenderprogramm auszuwerten oder zu steuern.

Element	Beschreibung
Eingangsdaten gültig	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)
Eingangsdaten vom Controller akzeptiert	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)

HINWEIS

Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden (siehe Kapitel „Consumer/Provider-Status“).

Kopf-Parameter

Manche Devices enthalten sogenannte Kopf-Parameter, in denen z. B. Diagnose, Alarm und Schnittstellen aktiviert oder deaktiviert werden können.

5.6.5 Input/Output PROFINET-IO Module

Ein Input/Output PROFINET-IO Modul kann mehrere Submodule haben. PROFINET-IO Controller haben in jedem Input/Output PROFINET-IO Modul ein Submodul.

In den PROFINET-IO Input-Modulen werden die Eingangsvariablen des PROFINET-IO Controllers eingetragen, die vom PROFINET-IO Device gesendet werden.

In den PROFINET-IO Output-Modulen werden die Ausgangsvariablen des PROFINET-IO Controllers eingetragen, die zum PROFINET-IO Device gesendet werden.

So legen Sie die benötigten PROFINET-IO Module an:

1. Öffnen Sie im Strukturbaum das Menü [Konfiguration] / [Protokolle] / [PROFINET-IO Device]
2. Wählen Sie im Kontextmenü des PROFINET-IO Device den Menüpunkt [Neu].
3. Wählen Sie passende Module aus.

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü der Input/Output PROFINET-IO Module öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Element	Beschreibung
Name	Name des Input/Output PROFINET-IO Moduls

Element	Beschreibung
Steckplatz	0 – 32767 Standardwert: 1

Register „Modell“
und „Features“

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Modulkennung*, *Hardware-/Software Version*. Diese Daten dienen als unterstützende Informationen zur Parametrierung des Geräts und können nicht geändert werden.

5.6.6 Submodul Input

Mit den Parametern der Submodule werden die Kommunikationsbeziehungen und das Verhalten der Module bei Verbindungsunterbrechung eingestellt.

Submodul Input (Eigenschaften)

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü der Input Submodule öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Element	Beschreibung
Name	Name des Input Submoduls.
Substeckplatz	Nicht änderbar, Standardwert: 1
IO Data CR, Eingänge	Auswahl der Kommunikationsbeziehung (Communication Relation CR) in der die Eingänge des Submoduls übertragen werden sollen: <ul style="list-style-type: none"> • Keine • Default Input CR
Eingangsdaten vom Controller akzeptiert	Auswahl der Kommunikationsbeziehung (Communication Relation CR), in der der IO Consumer Status (CS) des Submoduls übertragen werden soll: <ul style="list-style-type: none"> • Keine • Default Input CR
Shared Input	<ul style="list-style-type: none"> • Aktiviert Mehrere PROFINET-IO Controller können auf die Eingänge zugreifen. • Deaktiviert Nur ein PROFINET-IO Controller kann auf die Eingänge zugreifen.

Element	Beschreibung
Input-Werte bei IO-CR-Unterbrechung	<p>Verhalten der Input-Variablen dieses PROFINET-IO Submodul bei Verbindungsunterbrechung.</p> <ul style="list-style-type: none"> • Letzten Wert behalten <p>Die Input-Variablen werden auf dem momentanen Wert eingefroren und bis zur erneuten Verbindungsaufnahme verwendet.</p> <ul style="list-style-type: none"> • Initialwerte annehmen <p>Für die Input-Variablen werden die Initialdaten verwendet.</p>

Register „Modell“ und „Features“

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Submodulkennung*, *Hardware-/Software Version* oder *Datenlängen*. Diese Daten dienen als unterstützende Informationen zur Parametrierung des Geräts und können nicht geändert werden.

Submodul Input (Edit)

Der Menüpunkt [Edit] aus dem Kontextmenü der Input Submodule öffnet das Dialogfenster „Edit“, das die folgenden Register enthält.

Register „Systemvariablen“

Das Register „Systemvariablen“ stellt die folgenden Systemvariablen bereit, die es erlauben, den Zustand des PROFINET-IO Submoduls im Anwenderprogramm auszuwerten.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden (siehe Kapitel „Consumer/Provider-Status“).

Element	Beschreibung
Eingangsdaten gültig	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)
Eingangsdaten vom Controller akzeptiert	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)
Die folgenden Parameter sind nur für PROFIsafe-Module verfügbar	
Ausgangsdaten gültig	<ul style="list-style-type: none"> • TRUE Ausgangsdaten gültig (GOOD) • FALSE Ausgangsdaten ungültig (BAD)

Element	Beschreibung
Ausgangsdaten vom Device akzeptiert	<ul style="list-style-type: none"> • TRUE Ausgangsdaten gültig (GOOD) • FALSE Ausgangsdaten ungültig (BAD)

Element	Beschreibung
PROFIsafe Control	<p>PROFIsafe sendet vom Controller mit jeder Nachricht das PROFIsafe Control-Byte an das Device, das im Anwenderprogramm gesetzt werden kann (siehe Kapitel „PROFIsafe Control-Byte und Status-Byte“).</p> <ul style="list-style-type: none"> • Bit 0 iPar_EN_C: Um neue Parameter in das F-Device zu laden, muss iPar_EN_C auf TRUE gesetzt werden um das F-Device zu entriegeln.. Solange iPar_EN_C auf TRUE gesetzt ist, werden sichere Werte „0“ zwischen F-Host und F-Device ausgetauscht. • Bit 1 OA_C: Operator Acknowledge Bit 1 muss nach einem PROFIsafe-Fehler (z. B. CRC-Fehler oder Timeout) für mindestens einen PROFIsafe-Zyklus auf TRUE gesetzt werden. Soll eine PROFIsafe-Verbindung (neu) gestartet werden, darf die Freigabe (Operator-Acknowledge) erst dann geben, wenn keine gefährlichen Zustände existieren. • Bit 2, Bit 3 Reserved • Bit 4 Activate_FV_C: <ul style="list-style-type: none"> – FALSE Prozesswerte werden zwischen F-Host und F-Device ausgetauscht. – TRUE Sichere Werte „0“ werden zwischen F-Host und F-Device ausgetauscht. • Bit 5 Reserved • Bit 6 • Bit 7 Reset_Comm: Reset der PROFIsafe-Kommunikation, der Protokoll-Stück wird in den initialen Zustand versetzt. Bit 7 muss solange auf TRUE gesetzt werden, bis vom PROFIsafe-Status <i>Bit 2 Reset_Comm</i> der Wert TRUE zurückgelesen wird.

Element	Beschreibung
PROFIsafe RoundTrip Time last	Dies ist für einen F-Host die Zeit zwischen dem Absenden einer Datennachricht (mit Consecutive Number N) und dem Empfang des zugehörigen Acknowledgements (mit Consecutive Number N), gemessen in Millisekunden.

Element	Beschreibung
PROFIsafe Status	<p>PROFIsafe empfängt auf dem Host in jeder Nachricht das PROFIsafe Status-Byte, das im Anwenderprogramm ausgewertet werden kann.</p> <ul style="list-style-type: none"> • Bit 0 iPar_OK_S – TRUE Neue Parameter erhalten – FALSE Keine Änderung • Bit 1 OA_Req_S Operator Acknowledge Requested • Bit 2 Reset_Comm Ist der rückgelesene Wert <i>Reset_Comm</i> aus dem Host-Control-Byte. Dieses Bit signalisiert, ob der <i>Reset_Comm</i> angekommen ist. • Bit 3 FV_activated_S • Bit 4 Toggle_h • Bit 5 Device_Fault – TRUE Das F-Device hat einen Device-Fault gemeldet. – FALSE Das F-Device hat keinen Device-Fault gemeldet. • Bit 6 WD_Timeout – TRUE Entweder hat das F-Device einen WD-Timeout gemeldet, oder auf dem F-Host ist das Host-Timeout abgelaufen. – FALSE

Element	Beschreibung
	<p>Weder auf dem F-Device noch auf dem F-Host ist ein Timeout aufgetreten.</p> <ul style="list-style-type: none"> • Bit 7 <p>CRC</p> <ul style="list-style-type: none"> – TRUE <p>Entweder hat das F-Device einen CRC-Fehler gemeldet, oder auf dem F-Host ist ein CRC-Fehler aufgetreten.</p> <ul style="list-style-type: none"> – FALSE <p>Weder auf dem F-Device noch auf dem F-Host ist ein CRC-Fehler aufgetreten.</p>

Im Register Prozessvariablen werden die Eingangsvariablen eingetragen.

F-Parameter von Submodul Input (nur für PROFIsafe-Module)

PROFIsafe F-Devices benötigen für den sicheren Prozessdatenaustausch die genormten F-Parameter. Das F-Device nimmt die Kommunikation erst auf, wenn gültige F-Parameter eingestellt wurden. Die ausgegrauten Parameter können nicht bearbeitet werden und werden teilweise von der GSDML-Datei vorgegeben oder automatisch berechnet.

Element	Beschreibung
Name	Name des Moduls.
Index	Index des Moduls.
F_Par_Version	Nur V2-Mode wird unterstützt. V1-Mode wird abgelehnt. Wird über die GSDML-Datei bestimmt.
F_Source_Add	<p>Die <i>F_Source_Adresse</i> des F-Hosts muss innerhalb des PROFIsafe-Netzwerks eindeutig sein.</p> <p>Wertebereich: 1 – 65534</p>
F_Dest_Add	<p>Die <i>F_Destination_Adresse</i> des F-Device muss innerhalb des PROFIsafe-Netzwerks eindeutig sein.</p> <p>Wertebereich: 1 – 65534</p>
F_WD_Time	<p>Watchdog-Zeit.</p> <p>Wertebereich: 1 ms bis 65534 ms</p>
F_iPar_CRC	<i>F_iPar_CRC</i> des F-Device in dieses Feld eingeben.

Element	Beschreibung
F_SIL	Anzeige des SIL-Level. 0: SIL 1 1: SIL 2 2: SIL 3 3: No SIL Wird über die GSDML-Datei bestimmt.
F_Check_iPar	Anzeige des <i>iParameter</i> CRC. Wird über die GSDML-Datei bestimmt.
F_Block_ID	Struktur der <i>F-Parameter</i> . Wird über die GSDML-Datei bestimmt.
F_CRC_Length	Gibt an, ob der 3-Byte-CRC oder der 4-Byte-CRC verwendet wird. Wird über die GSDML-Datei bestimmt.
F_Par_CRC	Anzeige des F-Parameter CRC (CRC1). Wird aus den aktuellen F-Parametern berechnet.

5.6.7 Submodul Output

Mit den Parametern der Submodule werden die Kommunikationsbeziehungen und das Verhalten der Module bei Verbindungsunterbrechung eingestellt.

Submodul Output (Eigenschaften)

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü der Output Submodule öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Register Parameter

Element	Beschreibung
Name	Name des Output-Submoduls.
Substeckplatz	Nicht änderbar für PROFINET-IO Controller. Standardwert: 1.
IO Data CR, Ausgänge	Auswahl der Kommunikationsbeziehung (Communication Relation CR), in der die Ausgänge des Submoduls übertragen werden sollen. <ul style="list-style-type: none"> Keine Default Input CR
Ausgangsdaten vom Device akzeptiert	Auswahl der Kommunikationsbeziehung (Communication Relation CR) in der der IO Consumer-Status (CS) des Submoduls übertragen werden soll. <ul style="list-style-type: none"> Keine Default Output CR

Register „Modell“ und „Features“

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Submodulkennung*, *Hardware-/Software Version* oder *Datenlänge*. Diese Daten dienen als unterstützende Informationen zur Parametrierung des Geräts und können nicht geändert werden.

Submodul Output (Edit)

Der Menüpunkt [Edit] aus dem Kontextmenü des Submoduls Output öffnet das Dialogfenster „Edit“, das die folgenden Register enthält.

Register „Systemvariablen“

Das Register „Systemvariablen“ stellt die folgenden Systemvariablen bereit, die es erlauben, den Zustand des PROFINET-IO Submoduls im Anwenderprogramm auszuwerten.

Element	Beschreibung
Ausgangsdaten gültig	<ul style="list-style-type: none"> • TRUE Ausgangsdaten gültig (GOOD) • FALSE Ausgangsdaten ungültig (BAD)
Ausgangsdaten vom Device akzeptiert	<ul style="list-style-type: none"> • TRUE Ausgangsdaten gültig (GOOD) • FALSE Ausgangsdaten ungültig (BAD)
Die folgenden Parameter sind nur für PROFIsafe-Module verfügbar	
Eingangsdaten gültig	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)
Eingangsdaten vom Controller akzeptiert	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)

Weitere Parameter für PROFIsafe Module siehe Kapitel „Submodul Input (Edit)“.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden (siehe Kapitel „Consumer/Provider-Status“).

Im Register Prozessvariablen werden die Ausgangsvariablen eingetragen.

F-Parameter von Submodul Output (nur für PROFIsafe-Module)

Die Beschreibung der F-Parameter finden Sie im Kapitel „F-Parameter von Submodul Input (nur für PROFIsafe-Module)“.

5.6.8 Submodul Inputs und Outputs

Mit den Parametern der Submodule werden die Kommunikationsbeziehungen und das Verhalten der Module bei Verbindungsunterbrechung eingestellt.

Submodul Inputs und Outputs (Eigenschaften)

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü der Input und Output Submodule öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Register „Parameter“

Element	Beschreibung
Name	Name des Input/Output-Submoduls.
Substeckplatz	Nicht änderbar für PROFINET-IO Controller. Standardwert: 1.
IO Data CR, Eingänge	Auswahl der Kommunikationsbeziehung (Communication Relation CR), in der die Eingänge des Submoduls übertragen werden sollen. <ul style="list-style-type: none"> • Keine • Default Input CR
IO Data CR, Ausgänge	Auswahl der Kommunikationsbeziehung (Communication Relation CR), in der die Ausgänge des Submoduls übertragen werden sollen. <ul style="list-style-type: none"> • Keine • Default Output CR
Eingangsdaten vom Controller akzeptiert	Auswahl der Kommunikationsbeziehung (Communication Relation CR), in der der IO Consumer-Status (CS) des Submoduls übertragen werden soll. <ul style="list-style-type: none"> • Keine • Default Output CR
Ausgangsdaten vom Device akzeptiert	Auswahl der Kommunikationsbeziehung (Communication Relation CR), in der der IO Consumer-Status (CS) des Submoduls übertragen werden soll. <ul style="list-style-type: none"> • Keine • Default Input CR

Register „Modell“ und „Features“

Die Register „Modell“ und „Features“ zeigen weitere Einzelheiten der GSDML-Datei, wie zum Beispiel *Submodulkennung*, *Hardware-/Software Version* oder *Datenlänge*. Diese Daten dienen als unterstützende Informationen zur Parametrierung des Geräts und können nicht geändert werden.

Submodul Inputs und Outputs (Edit)

Der Menüpunkt [Edit] aus dem Kontextmenü der Submodule Input/Output öffnet das Dialogfenster „Edit“, das die folgenden Register enthält.

Register „Systemvariablen“

Das Register „Systemvariablen“ stellt die folgenden Systemvariablen bereit, die es erlauben, den Zustand des PROFINET-IO Submoduls im Anwenderprogramm auszuwerten.

Element	Beschreibung
Ausgangsdaten gültig	<ul style="list-style-type: none"> • TRUE Ausgangsdaten gültig (GOOD) • FALSE Ausgangsdaten ungültig (BAD)
Ausgangsdaten vom Device akzeptiert	<ul style="list-style-type: none"> • TRUE Ausgangsdaten gültig (GOOD) • FALSE Ausgangsdaten ungültig (BAD)
Eingangsdaten gültig	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)
Eingangsdaten vom Controller akzeptiert	<ul style="list-style-type: none"> • TRUE Eingangsdaten gültig (GOOD) • FALSE Eingangsdaten ungültig (BAD)

Weitere Parameter für PROFIsafe Module siehe Kapitel „Submodul Input (Edit)“.

Im Register Prozessvariablen werden die Ein- und Ausgangsvariablen im jeweiligen Bereich eingetragen.

HINWEIS



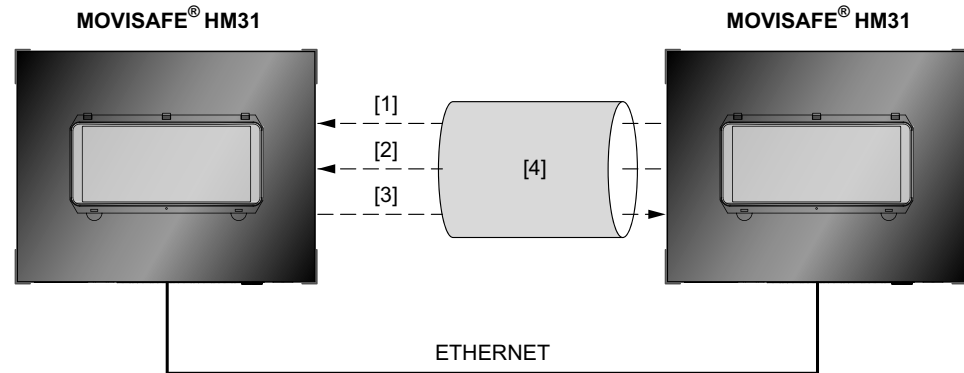
Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden (siehe Kapitel „Consumer/Provider-Status“).

F-Parameter von Submodul Inputs/Outputs (nur für PROFIsafe-Module)

Die Beschreibung der F-Parameter finden Sie im Kapitel „F-Parameter von Submodul Input (nur für PROFIsafe-Module)“.

5.6.9 Applikationsbeziehung (Eigenschaften)

Eine Applikationsbeziehung (AR: Application Relation) ist ein logisches Gebilde zum Datenaustausch zwischen Controller und Device. Die Datenübertragung innerhalb der Applikationsbeziehung findet in diesem Beispiel (siehe folgendes Bild) über die Standard-Kommunikationsbeziehungen (Alarm CR, Default Input CR und Default Output CR) statt. Diese Kommunikationsbeziehungen sind bereits standardmäßig in den Input- und Output-Modulen eingestellt



12225623051

- [1] Alarm CR
- [2] Default Input CR
- [3] Default Output CR
- [4] Applikationsbeziehung (AR)

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü von Application Relation öffnet das Dialogfenster „Eigenschaften“.

Element	Beschreibung
Name	Nicht änderbar.
AR UUID	Nicht änderbare Kennzahl zur eindeutigen Identifizierung der Applikationsbeziehung (AR).
Verbindungsaufbau Timeout Faktor	Mit diesem Parameter wird die Zeit berechnet, die aus Sicht eines PROFINET-IO-Devices während des Verbindungsaufbaus zwischen dem Senden der Antwort auf den Connect-Request und dem Empfang eines neuen Requests vom PROFINET-IO-Controller maximal verstreichen darf. Wertebereich: 1 – 1000 (× 100 ms) Standardwert: 600
Supervisor darf die AR übernehmen	Definition, ob ein PROFINET-IO Supervisor die Applikationsbeziehung (AR) übernehmen darf. <ul style="list-style-type: none">• 0: Nicht erlaubt• 1: Erlaubt Standardwert: 0

5.6.10 Alarm CR (Eigenschaften)

Innerhalb einer Applikationsbeziehung können mehrere Kommunikationsbeziehungen (Communication Relations, CR) aufgebaut werden. Über die Kommunikationsbeziehung Alarm CR überträgt ein PROFINET-IO Device Alarmer an den PROFINET-IO Controller.

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü von Application Relation öffnet das Dialogfenster „Eigenschaften“, das die folgenden Register enthält.

Element	Beschreibung
Name	Nicht änderbar.
VLAN-ID, hochpriori Alarmer	<p>Jedem Virtuellen LAN (VLAN) wird zur Abschottung eine eindeutige Nummer zugeordnet. Ein Gerät, das zum VLAN mit der ID = 1 gehört, kann mit jedem anderen Gerät im gleichen VLAN kommunizieren, nicht jedoch mit einem Gerät in einem anderen VLAN mit ID = 2, 3, ...</p> <p>Wertebereich, siehe auch IEC 61158-6</p> <ul style="list-style-type: none"> • 0x000: Kein VLAN • 0x001: Standard VLAN • 0x002 – 0xFFFF: Siehe IEEE 802.1 Q • Standardwert: 0
VLAN-ID, niederpriori Alarmer	<p>Beschreibung, siehe VLAN-ID, hochpriori Alarmer</p> <p>Standardwert: 0</p>
Alarm-Priorität	<ul style="list-style-type: none"> • Benutzer Priorität verwenden Die vom Benutzer vergebene Priorität wird verwendet. • Benutzer Priorität ignorieren Die vom Benutzer vergebene Priorität wird ignoriert. Der generierte Alarm ist immer niedrigprior.
Alarm Sendeversuche	<p>Maximale Anzahl an Sendeversuchen des Devices, falls der Controller nicht antwortet.</p> <p>Wertebereich: 3 – 15</p> <p>Standardwert: 10</p>
Alarm Timeout Faktor	<p>Mit dem RTA Timeout Factor wird die Timeout-Zeit des Devices berechnet, die nach dem Versenden eines RTA_Data(Alarm)-Frames und dem Erhalt des RTA_ACK-Frames maximal verstreichen darf.</p> <p>Wertebereich: 1 – 65535</p> <p>Standardwert: 5</p>

5.6.11 Input CR (Eigenschaften)

Über die Kommunikationsbeziehung Input CR überträgt ein PROFINET-IO Device Variablen zum PROFINET-IO Controller.

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü von Default Input CR öffnet das Dialogfenster „Eigenschaften“, das die folgenden Parameter enthält.

Element	Beschreibung
Name	Beliebiger, eindeutiger Name für ein Input CR. Das Default Input CR ist nicht änderbar.
Typ	1 (nicht änderbar).
Send Clock Faktor	Der Send Clock Faktor bestimmt den Send Clock der zyklischen Datenübertragung einer IO-CR. Send Clock = Send Clock Faktor × 31.25 µs Wertebereich: 1 – 128 Standardwert: 32
Reduktions Faktor	Der Reduktions Faktor ermöglicht die Unter- setzung der tatsächlichen Zykluszeit des Sen- dens der Daten einer IO-CR zur Send Clock. Die tatsächliche Zykluszeit der Daten errech- net sich so: Send Cycle = Reduktions Faktor × Send Clock Wertebereich: 1 – 16384 Standardwert: 32 (abhängig vom Gerät)
Watchdog Faktor	Mit dem Watchdog Faktor wird diejenige Zeit berechnet, die aus Sicht eines IO-CR-Consumers zwischen dem Empfang zweier Frames maximal verstreichen darf: Watchdog Time = Watchdog Faktor × Send Cycle Wertebereich: 1 – 7680 Standardwert: 3
VLAN-ID	Jedem Virtuellen LAN (VLAN) wird zur Ab- schottung eine eindeutige Nummer zugeord- net. Ein Gerät, das zum VLAN mit der ID = 1 gehört, kann mit jedem anderen Gerät im glei- chen VLAN kommunizieren, nicht jedoch mit einem Gerät in einem anderen VLAN mit ID = 2, 3, ... Wertebereich, siehe auch IEC 61158-6 <ul style="list-style-type: none"> • 0x000: Kein VLAN • 0x001: Standard VLAN • 0x002 – 0xFFFF: Siehe IEEE 802.1 Q • Standardwert: 0

Input CR (Edit)

Der Menüpunkt [Edit] aus dem Kontextmenü des Submoduls Output öffnet das Dialogfenster „Edit“, das die folgenden Register enthält.

Element	Wert	Beschreibung
Daten Status Input CR	0	State <i>Primary</i> beschreibt bei redundanten Verbindungen den führenden Kanal. <ul style="list-style-type: none"> • 1 = Primary • 0 = Backup Bei Mono-Verbindungen gilt: <ul style="list-style-type: none"> • 1 = Verbunden • 0 = Nicht verbunden
	1	Nicht verwendet
	2	Data Valid <i>Invalid</i> wird in der Hochlaufphase gesetzt oder wenn die Applikation nicht in der Lage ist, Fehler über den IOPS zu melden. <ul style="list-style-type: none"> • 1 = Valid • 0 = Invalid
	3	Nicht verwendet
	4	Process State Hat nur informativen Charakter. Die tatsächliche Gültigkeit der Daten wird über IOPS gemeldet. <ul style="list-style-type: none"> • 1 = Run • 0 = Stop
	5	Problem Indicator Bei <i>Problem detected</i> werden Details über Diagnosedaten der Alarm-CR zur Verfügung gestellt. <ul style="list-style-type: none"> • 1 = Regular operation • 0 = Problem detected
	6	Nicht verwendet.
	7	Nicht verwendet.

Output CR (Eigenschaften)

Innerhalb einer Applikationsbeziehung können mehrere Kommunikationsbeziehungen (Communication Relations, CR) aufgebaut werden. Über die Kommunikationsbeziehung Output CR überträgt der PROFINET-IO Controller Variablen zum PROFINET-IO Device.

Der Menüpunkt [Eigenschaften] aus dem Kontextmenü von Output CR öffnet das Dialogfenster „Eigenschaften“, das die folgenden Parameter enthält.

Element	Beschreibung
Name	Beliebiger, eindeutiger Name für ein Output CR. Das Default Output CR ist nicht änderbar.
Typ	2 (nicht änderbar).
Send Clock Faktor	Der Send Clock Faktor bestimmt den Send Clock der zyklischen Datenübertragung einer IO-CR. Send Clock = Send Clock Faktor × 31.25 µs Wertebereich: 1 – 128 Standardwert: 32
Reduktions Faktor	Zur Einstellung der Übertragungshäufigkeit. Der Reduktions Faktor ermöglicht die Unter- setzung der tatsächlichen Zykluszeit des Sen- dens der Daten einer IO-CR. Die tatsächliche Zykluszeit der Daten errechnet sich so: Send Cycle = Reduktions Faktor × Send Clock Wertebereich: 1 – 16384 Standardwert: 32
Watchdog Faktor	Mit dem Watchdog Faktor wird diejenige Zeit berechnet, die aus Sicht eines IO-CR-Consumers zwischen dem Empfang zweier Frames maximal verstreichen darf: Watchdog Time = Watchdog Faktor × Send Cycle Wertebereich: 1 – 7680 Standardwert: 3
VLAN-ID	Jedem Virtuellen LAN (VLAN) wird zur Ab- schottung eine eindeutige Nummer zugeord- net. Ein Gerät, das zum VLAN mit der ID = 1 gehört, kann mit jedem anderen Gerät im glei- chen VLAN kommunizieren, nicht jedoch mit einem Gerät in einem anderen VLAN mit ID = 2, 3, ... Wertebereich, siehe auch IEC 61158-6 <ul style="list-style-type: none"> • 0x000: Kein VLAN • 0x001: Standard VLAN • 0x002 – 0xFFFF: Siehe IEEE 802.1 Q • Standardwert: 0

5.7 PROFINET-IO-Device

Dieses Kapitel beschreibt die Eigenschaften des PROFINET-IO-Device sowie die Menüfunktionen und Dialoge in SILworX®, die zur Konfiguration des PROFINET-IO-Controllers benötigt werden.

5.7.1 Systemanforderung

- Benötigte Ausstattung und Systemanforderung.

Element	Beschreibung
Steuerung	Sicherheitssteuerung MOVISAFE® HM31
CPU-Modul	Die Ethernet-Schnittstellen des Prozessormoduls können für PROFINET-IO nicht verwendet werden.
COM-Modul	Ethernet 10/100BaseT
Aktivierung	Die Freischaltung erfolgt per Software-Freischaltcode (siehe Kapitel "Kommunikation" im Systemhandbuch "Sicherheitssteuerung MOVISAFE® HM31").

- PROFINET-IO-Device Eigenschaften.

Element	Beschreibung
Sicherheitsgerichtet	Nein
Übertragungsrate	100 Mbit/s Vollduplex
Transportweg	Ethernet-Schnittstellen der COM-Module. Verwendete Ethernet-Schnittstellen sind simultan auch für andere Protokolle nutzbar.
Konformitätsklasse	Das PROFINET-IO-Device entspricht den Anforderungen der Conformance Class A.
Real Time Class	RT-Klasse 1.
Max. Anzahl PROFINET-IO-Devices	Für jedes COM-Modul kann ein PROFINET-IO-Device konfiguriert werden.
Max. Anzahl Applikationsbeziehungen (AR's) zum PROFINET-IO-Controller	Ein PROFINET-IO-Device kann maximal 5 Applikationsbeziehungen (AR's) zu einem PROFINET-IO-Controller aufbauen.
Max. Anzahl (CR's pro AR) Kommunikationsbeziehungen	Standard: 1 Input, 1 Output, 1 Alarm
Max. Prozessdatenlänge aller konfigurierten PROFINET-IO-Module	Output: max. 1440 Bytes Input: max. 1440 Bytes
Priorisierung der Daten	Möglich über die Einstellung der <i>Reduction Rate</i> auf der Device-Ebene.

5.8 PROFINET-IO / PROFIsafe-Beispiel (Device)

In diesem Kapitel wird die Konfiguration des PROFINET-IO / PROFIsafe Device beschrieben.

5.8.1 Konfiguration des PROFINET-IO-Device in SILworX®

Gehen Sie so vor:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle].
2. Um ein neues PROFINET-IO-Device hinzuzufügen, wählen Sie im Kontextmenü von Protokolle den Menüpunkt [Neu] / [PROFINET-IO Device].
3. Wählen Sie im Kontextmenü des PROFINET-IO-Device den Menüpunkt [Eigenschaften].
4. Tragen Sie im Feld "Name" den Gerätenamen des PROFINET-IO-Device ein.
5. Wählen Sie ein COM-Modul aus



12843484683

PROFINET-IO-Module anlegen

So legen Sie die benötigten PROFINET-IO-Module an:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Device].
2. Wählen Sie im Kontextmenü des PROFINET-IO-Device den Menüpunkt [Neu].
3. Wählen Sie für dieses Beispiel die folgenden passenden Module.

PROFINET-IO / PROFIsafe-Modul	Steckplatz
In 1 Byte	1
Safe Out 1Byte	2

PROFINET-IO-Device-Module nummerieren

So nummerieren Sie die PROFINET-IO-Device-Module:

1. Wählen Sie im Kontextmenü des ersten PROFINET-IO-Device-Moduls den Menüpunkt [Eigenschaften].
2. Geben Sie im Feld „Steckplatz“ den Wert „1“ ein.
3. Wiederholen Sie diesen Schritt für alle weiteren PROFINET-IO-Device-Module mit fortlaufender Nummerierung.

HINWEIS



Nummerieren Sie die Module so, wie das PROFINET-IO Device tatsächlich aufgebaut ist.

4. Der folgende Schritt ist nur für die PROFIsafe-Module notwendig.

Tragen Sie im Feld „PROFIsafe F_Destination_Address“ die Adresse des PROFIsafe-Moduls ein.

Konfiguration des PROFINET-IO-Device Eingangsmoduls

HINWEIS



Die Summe der Variablen in Byte muss mit der Größe des jeweiligen Moduls in Byte übereinstimmen.

Gehen Sie so vor, um das Eingangsmodul [01] In 1 Byte zu konfigurieren:

1. Wählen Sie im PROFINET-IO-Device das Eingangsmodul [01] In 1 Byte.
2. Rufen Sie im Kontextmenü des Eingangsmoduls [01] In 1 Byte den Menüpunkt [Edit] auf.
3. Wählen Sie im Dialogfenster „Edit“ die Registerkarte „Prozessvariablen“.
4. Wählen Sie in der Objektauswahl die passende Variable und ziehen Sie sie per Drag & Drop in den Bereich „Eingangssignale“.
5. Rufen Sie das Kontextmenü durch einen Rechtsklick auf einer leeren Stelle im Bereich „Eingangssignale“ auf und wählen Sie den Menüpunkt [Neue Offsets], damit die Offsets der Variablen neu generiert werden.
6. Wählen Sie im Dialogfenster „Edit“ die Registerkarte „Systemvariablen“. Wenn eine Steuerung des Consumer/Provider Status nicht gewünscht ist, belegen Sie die Ausgangsvariable *Eingangsdaten vom Device akzeptiert* mit einer Globalen Variable mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

Konfiguration des PROFIsafe-Device Ausgangsmoduls

Gehen Sie so vor, um das Ausgangsmodul [02] Safe Out 1 Byte zu konfigurieren:

1. Wählen Sie im PROFIsafe Device das Ausgangsmodul [02] Safe Out 1 Byte.
2. Wählen Sie im Kontextmenü des Ausgangsmoduls [02] Safe 1 Byte den Menüpunkt [Edit].
3. Wählen Sie im Dialogfenster „Edit“ die Registerkarte „Prozessvariablen“.
4. Wählen Sie in der Objektauswahl die passende Variable aus und ziehen Sie sie per Drag & Drop in den Bereich Ausgangssignale.
5. Rufen Sie das Kontextmenü durch einen Rechtsklick auf einer leeren Stelle im Bereich „Ausgangssignale“ auf und wählen Sie den Menüpunkt [Neue Offsets], damit die Offsets der Variablen neu generiert werden.
6. Wählen Sie im Dialogfenster „Edit“ die Registerkarte „Systemvariablen“. Wenn eine Steuerung des Consumer/Provider Status nicht gewünscht ist, belegen Sie die Ausgangsvariable *Ausgangsvariable gültig* und *Eingangsdaten vom Device akzeptiert* mit einer Globalen Variable mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

Verifikation der PROFINET-IO-Device Konfiguration

Gehen Sie zur Verifikation der PROFINET-IO-Device Konfiguration so vor:

1. Wählen Sie im Strukturbaum den Menüpunkt [Konfiguration] / [Ressource] / [PROFINET-IO Device].
2. Klicken Sie in der Action Bar auf die Schaltfläche [Verifikation] und bestätigen Sie mit [OK].
3. Prüfen Sie die Einträge im Logbuch sorgfältig, korrigieren Sie sie gegebenenfalls.

HINWEIS



Die Konfiguration des PROFINET-IO-Device müssen Sie mit dem Anwenderprogramm der PROFINET-IO-Device-Ressource neu kompilieren und in die Steuerung laden, damit sie für die PROFINET-IO-Kommunikation wirksam wird.

5.9 Menüfunktionen des PROFINET-IO-Device

5.9.1 Menü Eigenschaften

Der Menübefehl "Eigenschaften" aus dem Kontextmenü des PROFINET-IO- Device öffnet das Dialogfenster "Eigenschaften".

Element	Beschreibung
Typ	PROFINET-IO-Device
Name	Beliebiger, eindeutiger Name für ein PROFINET-IO-Device.
Aktualisierungsintervall der Prozessdaten in ms	<p>Aktualisierungsintervall in Millisekunden, mit der die Daten des Protokolls zwischen COM und CPU ausgetauscht werden. Ist das Aktualisierungsintervall Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <ul style="list-style-type: none"> • Wertebereich: 4 bis ($2^{31} - 1$) • Standardwert: 0
Prozessdaten-Konsistenz erzwingen	<ul style="list-style-type: none"> • Aktiviert (Standardwert): Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU. • Deaktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU Zyklen zu je 1100 Byte pro Datenrichtung. Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.
Modul	Auswahl des COM-Moduls auf dem dieses Protokoll abgearbeitet wird.
Max. µP-Budget aktivieren	<ul style="list-style-type: none"> • Aktiviert: Limit des µP-Budget aus dem Feld <i>Max. µP-Budget</i> in % übernehmen. • Deaktiviert: Kein Limit des µP-Budget, für dieses Protokoll verwenden.

Element	Beschreibung
Max. µP-Budget in %	<p>Maximales µP-Budget des Moduls, welches bei der Abarbeitung des Protokolls produziert werden darf.</p> <ul style="list-style-type: none"> Wertebereich: 1 – 100 % Standardwert: 30 %
RPC Port Server	<p>Remote Procedure Call Port.</p> <ul style="list-style-type: none"> Wertebereich: 1024 – 65535 Standardwert: 49152 <p>RPC Port Server und RPC Port Client dürfen nicht identisch sein!</p>
RPC Port Client	<p>Remote Procedure Call Port.</p> <ul style="list-style-type: none"> Wertebereich: 1024 – 65535 Standardwert: 49153 <p>RPC Port Server und RPC Port Client dürfen nicht identisch sein!</p>
AutoAwaitFParamsOnConnLoss	<p>Dieser Parameter wird nur für PROFIsafe-Module verwendet.</p> <p>Nach jedem Verbindungsverlust zum F-Host müssen die F-Parameter erneut in das F-Device geladen werden. Um die Inbetriebnahme von PROFIsafe zu erleichtern kann dieser Parameter aktiviert werden. Danach übernimmt das F-Device bei Neustart oder bei Verbindungsverlust automatisch die erforderlichen F-Parameter.</p> <p>Nach der Inbetriebnahme muss dieser Parameter für PROFIsafe konformes Verhalten zwingend deaktiviert werden.</p> <ul style="list-style-type: none"> Aktiviert (Standardwert): Das F-Device geht automatisch in den Zustand <i>Warten auf F-Parameter</i>. Deaktiviert: Das F-Device muss vom Anwender manuell über ein Online-Kommando in den Zustand <i>Warten auf F-Parameter</i> gebracht werden.

5.9.2 PROFINET-IO-Module

Die folgenden PROFINET-IO-Module stehen im PROFINET-IO-Device zur Verfügung.

PROFINET-IO-Modul	Max. Größe Eingangsvariablen	Max. Größe Ausgangsvariablen
In 1 Byte	1 Byte	
In 2 Bytes	2 Bytes	
In 4 Bytes	4 Bytes	
In 8 Bytes	8 Bytes	
In 16 Bytes	16 Bytes	
In 32 Bytes	32 Bytes	

PROFINET-IO-Modul	Max. Größe Eingangsvariablen	Max. Größe Ausgangsvariablen
In 64 Bytes	64 Bytes	
In 128 Bytes	128 Bytes	
In 256 Bytes	256 Bytes	
In 512 Bytes	512 Bytes	
In 1024 Bytes	1024 Bytes	
In-out 1 Byte	1 Byte	1 Byte
In-out 2 Bytes	2 Bytes	2 Bytes
In-out 4 Bytes	4 Bytes	4 Bytes
In-out 8 Bytes	8 Bytes	8 Bytes
In-out 16 Bytes	16 Bytes	16 Bytes
In-out 32 Bytes	32 Bytes	32 Bytes
In-out 64 Bytes	64 Bytes	64 Bytes
In-out 128 Bytes	128 Bytes	128 Bytes
In-out 256 Bytes	256 Bytes	256 Bytes
In-out 512 Bytes	512 Bytes	512 Bytes
Out 1 Byte		1 Byte
Out 2 Bytes		2 Bytes
Out 4 Bytes		4 Bytes
Out 8 Bytes		8 Bytes
Out 16 Bytes		16 Bytes
Out 32 Bytes		32 Bytes
Out 64 Bytes		64 Bytes
Out 128 Bytes		128 Bytes
Out 256 Bytes		256 Bytes
Out 512 Bytes		512 Bytes
Out 1024 Bytes		1024 Bytes

5.9.3 PROFIsafe-Module

Die folgenden PROFIsafe-Module stehen im PROFINET-IO-Device zur Verfügung.

PROFIsafe-Modul	Max. Größe Eingangsvariablen	Max. Größe Ausgangsvariablen
Safe In 1 Byte	1 Byte	
Safe In 2 Bytes	2 Bytes	
Safe In 4 Bytes	4 Bytes	
Safe In 8 Bytes	8 Bytes	
Safe In 16 Bytes	16 Bytes	
Safe In 32 Bytes	32 Bytes	

PROFIsafe-Modul	Max. Größe Eingangsvariablen	Max. Größe Ausgangsvariablen
Safe In 64 Bytes	64 Bytes	
Safe In 123 Bytes	123 Bytes	
Safe In-out 1 Byte	1 Byte	1 Byte
Safe In-out 2 Bytes	2 Bytes	2 Bytes
Safe In-out 4 Bytes	4 Bytes	4 Bytes
Safe In-out 8 Bytes	8 Bytes	8 Bytes
Safe In-out 16 Bytes	16 Bytes	16 Bytes
Safe In-out 32 Bytes	32 Bytes	32 Bytes
Safe In-out 64 Bytes	64 Bytes	64 Bytes
Safe In-out 123 Bytes	123 Bytes	123 Bytes
Safe Out 1 Byte		1 Byte
Safe Out 2 Bytes		2 Bytes
Safe Out 4 Bytes		4 Bytes
Safe Out 8 Bytes		8 Bytes
Safe Out 16 Bytes		16 Bytes
Safe Out 32 Bytes		32 Bytes
Safe Out 64 Bytes		64 Bytes
Safe Out 123 Bytes		123 Bytes

So legen Sie ein PROFINET- oder PROFIsafe-Modul an:

1. Öffnen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Device].
2. Wählen Sie im Kontextmenü von PROFINET-IO-Device den Menüpunkt [Module einfügen].
3. Wählen Sie ein passendes Modul aus, um die erforderlichen Prozessdaten zu transportieren.

Klicken Sie mit der rechten Maustaste auf das ausgewählte Modul und wählen Sie im Kontextmenü den Menüpunkt [Edit].

- Fügen Sie im Register "Prozessvariablen" die Ein- und/oder Ausgangsvariablen ein.
- Wenn eine Steuerung des Consumer/Provider-Status mit spezieller Anwenderprogrammlogik nicht gewünscht ist, belegen Sie im Register Systemvariablen die Ausgangsvariablen *Eingangsdaten vom Controller akzeptiert* und *Ausgangsdaten gültig* mit einer Globalen Variablen mit dem Initialwert TRUE.

HINWEIS



Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

5. Die folgende Einstellung ist nur für PROFIsafe-Module notwendig

Geben Sie im Register "Eigenschaften" den Steckplatz und die *F_Destination_Address* ein.

HINWEIS

Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

5.9.4 PROFINET-IO- und PROFIsafe-Modul

Mit den Parametern der Module werden die Kommunikationsbeziehungen und das Verhalten der Module bei Verbindungsunterbrechung eingestellt.

- Dialogfenster "Eigenschaften"

Der Menübefehl [Eigenschaften] aus dem Kontextmenü der Module öffnet das Dialogfenster "Eigenschaften". Das Dialogfenster "Eigenschaften" hat folgende Register.

Element	Beschreibung
Name	Name des Device-Moduls.
Steckplatz	0 bis 32767.
Modul Kennung	Eindeutige Zahl.
Substeckplatz	Anzahl der Substeckplätze.
Prozessdatenverhalten	Wert der Prozessdaten nach einem Verbindungsabbruch. <ul style="list-style-type: none"> • Letzte gültige Prozessdaten behalten • Initialdaten annehmen
Länge der IO-Input Daten	1 bis 123.
Länge der IO Output Daten	1 bis 123.
PROFIsafe F_destination_address	Die F_Destination_Adresse des F-Devices muss innerhalb des PROFIsafe-Netzwerkes eindeutig sein. Wertebereich: 1 – 65534

- Dialogfenster "Edit"

Der Menübefehl [Edit] aus dem Kontextmenü der Submodule öffnet das Dialogfenster "Edit".

Das Register "Systemvariablen" stellt die folgenden Systemvariablen bereit, um den Zustand des Submoduls im Anwenderprogramm auszuwerten.

HINWEIS

Diese Systemvariablen können zur Steuerung des Consumer/Provider-Status verwendet werden, siehe Kapitel "Steuerung des Consumer/Provider Status (IOxS)".

Element	Beschreibung
Ausgangsdaten gültig	<ul style="list-style-type: none"> • TRUE: Ausgangsdaten gültig • FALSE: Ausgangsdaten ungültig

Element	Beschreibung
Ausgangsdaten vom Controller akzeptiert	<ul style="list-style-type: none"> • TRUE: Ausgangsdaten gültig • FALSE: Ausgangsdaten ungültig
Eingangsdaten gültig	<ul style="list-style-type: none"> • TRUE: Eingangsdaten gültig • FALSE: Eingangsdaten ungültig
Eingangsdaten vom Device akzeptiert	<ul style="list-style-type: none"> • TRUE: Eingangsdaten gültig • FALSE: Eingangsdaten ungültig

- Die folgenden Parameter sind nur für PROFIsafe-Module verfügbar.

Element	Beschreibung
PROFIsafe Control	<p>PROFIsafe sendet vom Controller mit jeder Nachricht das PROFIsafe Control-Byte an das Device, welches im Anwenderprogramm gesetzt werden kann (siehe Kap. "PROFIsafe Control-Byte und Status-Byte").</p> <ul style="list-style-type: none"> • Bit 0: iPar_EN_DC Freigabe vom Controller entriegelt das Device um neue iParameter in das Device zu laden. • Bit 1: OA_Req_DC (Operator Acknowledge) Operator Acknowledge aus dem Host-Control Byte. • Bit 2: Reset_Comm Dies ist der rückgelesene Wert Reset_Comm aus dem Host-Control-Byte. • Bit 3: activate_FV_DC FALSE: Prozesswerte werden zwischen F-Host und F-Device ausgetauscht. TRUE: Failsafe Values "0" werden zwischen F-Host und F-Device ausgetauscht. • Bit 4: Toggle_d Toggle Bit des F-Device. • Bit 5: Cons_nr_R Die Consecutive Number wird immer dann übernommen, wenn das Toggle_d Bit eine Änderung zwischen zwei aufeinander folgenden Control Bytes aufweist, d. h. dies ist unabhängig vom Auftreten eines Fehlers. • Bit 6: F_ParamValid TRUE: F-Parametrierung ist erfolgt FALSE: sonst • Bit 7: F_Param_ConfiguredTwice TRUE: Das F-Device ist mindestens zweimal mit unterschiedlichen F-Parameter parametrierung worden. FALSE: sonst
PROFIsafe F_Par_iPar_CRC	<p>iParameter sind individuelle oder Technologie spezifische F-Device-Parameter. Der iPar_CRC ergibt sich aus der Konfiguration des F-Device.</p> <p>Hinweis:</p> <p>Es liegt in der Verantwortung des Benutzers, nach der iParametrierung den richtigen iPar_CRC einzustellen und dann in den scharfen Betrieb zu wechseln.</p>

Element	Beschreibung
PROFIsafe F_SIL	<ul style="list-style-type: none"> • 0: SIL 1 • 1: SIL 2 • 2: SIL 3 • 3: Kein SIL
PROFIsafe RoundTrip Time last	Dies ist für einen F-Host die Zeit zwischen dem Absenden einer Datennachricht (mit Consecutive Number N) und dem Empfang des zugehörigen Acknowledgments (mit Consecutive Number N), gemessen in Millisekunden.
PROFIsafe Status	<p>PROFIsafe sendet vom Device mit jeder Nachricht das PROFIsafe Status-Byte an den Controller, welches im Device-Anwenderprogramm gesetzt werden kann.</p> <ul style="list-style-type: none"> • Bit 0: iPar_OK_DS Neue iParameter erhalten. • Bit 1: Default_Fault_DS TRUE: Device-Fehler FALSE: Kein Device-Fehler Wird erst ab dem PROFIsafe-Zustand 21 <i>Await Message</i> berücksichtigt. • Bit 2: Reserviert • Bit 3: Reserviert • Bit 4: FV_activated_DS Failsafe value aktiviert. • Bit 5: Reserviert • Bit 6: • Bit 7: Reset_Comm Protokoll-Stack wird in den initialen Zustand überführt.

Im Register "Prozessvariablen" werden die Eingangsvariablen eingetragen.

5.10 PROFINET-IO-Funktionsbausteine

Zum azyklischen Datenaustausch stehen Ihnen in SILworX® die folgenden PROFINET-IO-Funktionsbausteine zur Verfügung.

Die Funktionsbausteine werden im Anwenderprogramm parametrierbar, sodass die Funktionen des Controllers und der Devices (Alarmer, Diagnosedaten, Zustände) im Anwenderprogramm gesetzt und gelesen werden können.

Funktionsbaustein	Beschreibung der Funktion
MSTAT	Zustand des Controllers durch das Anwenderprogramm steuern.
RALRM	Alarmmeldungen der IO-Devices lesen.

Funktionsbau- stein	Beschreibung der Funktion
RDREC	Datensätze der IO-Devices lesen.
SLACT	Zustand der IO-Devices durch das Anwenderprogramm steuern.
WRREC	Datensätze der IO-Devices schreiben.

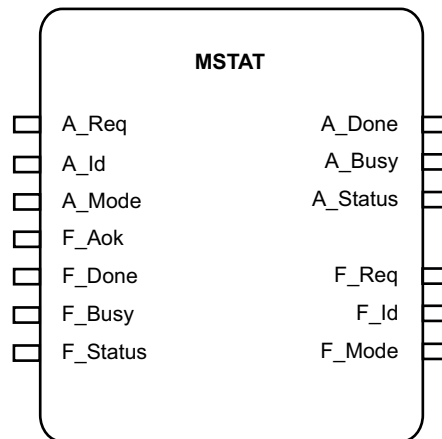
HINWEIS



- Die PROFINET-IO-Funktionsbausteine müssen über die Bibliothek *PROFlib.A3* eingebunden werden. Die Bibliothek *PROFlib.A3* finden Sie auf der SILworX®-Installations-CD.
- Die Parametrierung der Funktionsbausteine ist im Kapitel "Konfiguration der Funktionsbausteine" beschrieben.

5.10.1 Funktionsbaustein MSTAT

Das folgende Bild zeigt den Aufbau des Funktionsbausteins MSTAT mit allen Eingängen (links) und Ausgängen (rechts).



7435913099

Mit dem Funktionsbaustein MSTAT kann der PROFINET-IO-Controller vom Anwenderprogramm gesteuert werden. Somit ist es möglich, den PROFINET-IO-Controller durch einen mechanischen Schalter an einem physikalischen Eingang oder durch einen Timer in einen der folgenden Zustände zu setzen:

- 0: OFFLINE
- 1: OPERATE

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

Ein- und Ausgänge des Funktionsbaustein 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-Eingänge	Beschreibung	Typ
A_ID	Master-ID (nicht genutzt)	DWORD
A_Mode	Der PROFINET-IO-Controller kann in folgende Zustände gesetzt werden: <ul style="list-style-type: none"> 0: OFFLINE 1: OPERATE 	INT

A-Ausgänge	Beschreibung	Typ
A_Done	TRUE: Der PROFINET-IO-Controller wurde in den am Eingang A_Mode definierten Zustand gesetzt.	BOOL
A_Busy	TRUE: Das Setzen des PROFINET-IO-Controllers ist noch nicht beendet.	BOOL
A_Status	Status oder Fehlercode (siehe Kapitel "Fehlercodes der Funktionsbausteine").	DWORD

Ein- und Ausgänge des Funktionsbaustein mit dem Präfix "F"

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

HINWEIS



Die Verbindung des Funktionsbausteins MSTAT im Strukturbaum (im Ordner "Funktionsbausteine") mit dem Funktionsbaustein MSTAT (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor im Variableneditor erstellt werden.

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

F-Eingänge	Typ
F_ACK	BOOL
F_DONE	BOOL
F_BUSY	BOOL
F_STATUS	DWORD

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

F-Ausgänge	Typ
F_REQ	BOOL
F_ID	DWORD
F_MODE	INT

So erstellen Sie den Funktionsbaustein MSTAT im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [Funktionsbausteine] / [Neu] öffnen.
2. Funktionsbaustein MSTAT wählen.

3. Rechtsklick auf Funktionsbaustein MSTAT und Menüpunkt [Edit] wählen. Die Variablenzuweisung zum Funktionsbaustein wird geöffnet.
4. Die Eingänge des Funktionsbausteins MSTAT im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins MSTAT im Anwenderprogramm verbunden wurden.

Eingänge	Typ
M_ID	DWORD
MODE	INT
REQ	BOOL

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

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Funktionsbaustein MSTAT bedienen

So bedienen Sie den Funktionsbaustein MSTAT:

1. Im Anwenderprogramm den Eingang *A_Mode* auf den gewünschten Zustand setzen. Wird *A_Mode* nicht gesetzt, wird nach Schritt 2 ein Fehlercode am Ausgang *A_Status* ausgegeben und der PROFINET-IO-Controller wird nicht gesetzt.
2. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen. Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*. Der Ausgang *A_Busy* ist TRUE, bis der MSTAT-Befehl abgearbeitet ist. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Done* auf TRUE.

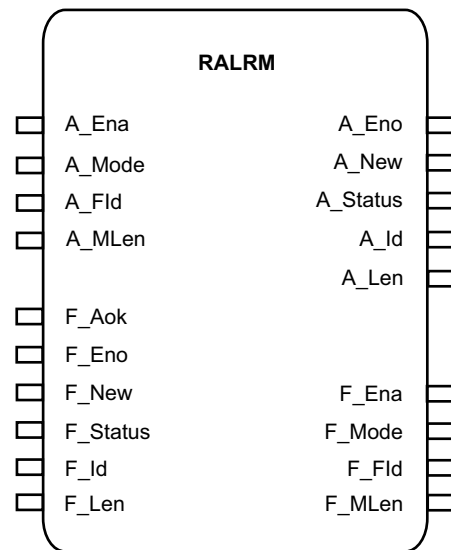
HINWEIS



Konnte der vorgegebene Mode nicht gesetzt werden, wird ein Fehlercode am Ausgang *A_Status* ausgegeben. Der aktuelle Mode des Controllers kann der Variablen *Controller_Status* entnommen werden (siehe Kapitel "PROFINET-IO-Hilfsfunktionsbausteine").

5.10.2 Funktionsbaustein RALRM

Das folgende Bild zeigt den Aufbau des Funktionsbausteins RALRM mit allen Eingängen (links) und Ausgängen (rechts).



7436172555

Der Funktionsbaustein RALRM dient zur Auswertung von Alarmen.

Alarme sind eine spezielle Form von Diagnosemeldungen, die vorrangig behandelt werden. Alarme melden der Anwendung wichtige Ereignisse, die Reaktionen seitens der Anwendung erfordern (z. B. ein WRREC). Dies ist jedoch herstellerabhängig und kann dem Gerätehandbuch des PROFINET-IO-Device entnommen werden.

Solange der Funktionsbaustein RALRM aktiv ist, wartet dieser auf Alarmmeldungen der Slaves. Wird ein Alarm empfangen, wird der Ausgang **A_NEW** für mindestens einen Zyklus auf TRUE geschaltet und die Alarmdaten können per Alarmtelegramm ausgelesen werden. Vor dem nächsten Alarm geht **A_NEW** für mindestens einen Zyklus auf FALSE. Alle Alarme werden implizit bestätigt. Es geht kein Alarm verloren.

Das Anwenderprogramm ist bei Verwendung mehrerer Funktionsbausteine RALRM so anzulegen, dass immer nur ein Funktionsbaustein RALRM aktiv ist.

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine")

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_Ena	Mit TRUE wird der Funktionsbaustein freigegeben.	BOOL
A_Mode	Nicht genutzt.	INT
A_FID	Nicht genutzt.	DWORD
A_MLen	Maximal erwartete Länge der zu empfangenden Alarmdaten in Bytes.	INT

A-Ausgänge	Beschreibung	Typ
A_Eno	TRUE: Funktionsbaustein aktiv FALSE: Funktionsbaustein nicht aktiv	BOOL
A_New	TRUE: Neuer Alarm wurde empfangen FALSE: Kein neuer Alarm	BOOL
A_Status	Status oder Fehlercode (siehe Kapitel "Fehlercodes der Funktionsbausteine").	DWORD
A_ID	Identifikationsnummer des Alarm auslösenden IO-Device.	DWORD
A_Len	Länge der empfangenen Alarmdaten in Bytes.	INT

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix "F"

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

HINWEIS



Die Verbindung des Funktionsbausteins RALRM im Strukturbaum (im Ordner "Funktionsbausteine") mit dem Funktionsbaustein RALRM (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor im Variableneditor erstellt werden.

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

F-Eingänge	Typ
F_ACK	BOOL
F_ENO	BOOL
F_NEW	BOOL
F_STATUS	DWORD
F_ID	DWORD
F_LEN	INT

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

F-Ausgänge	Typ
F_ENA	BOOL
F_MODE	INT
F_FID	DWORD
F_MLEN	INT

So erstellen Sie den Funktionsbaustein RALRM im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [Funktionsbausteine] / [Neu] öffnen.
2. Funktionsbaustein RALRM wählen.
3. Rechtsklick auf Funktionsbaustein RALRM und Menüpunkt [Edit] wählen. Die Variablenzuweisung zum Funktionsbaustein wird geöffnet.
4. Die Eingänge des Funktionsbausteins RALRM im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins RALRM im Anwenderprogramm verbunden wurden.

Eingänge	Typ
EN	BOOL
F_ID	DWORD
MLEN	INT
MODE	INT

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

Ausgänge	Typ
ACK	BOOL
ENO	BOOL
ID	DWORD
LEN	INT
NEW	BOOL
STATUS	DWORD

Alarmdaten

Im Register "Prozessvariablen des Funktionsbausteins RALRM im Strukturbaum sind Variablen zu definieren, deren Struktur zu den Alarmdaten passt. Werden keine Variablen definiert, können Alarmdaten zwar angefordert, aber nicht gelesen werden.

Eine Alarmmeldung enthält mindestens vier Bytes. Die ersten vier Bytes der Alarmmeldung enthalten die Standard-Alarmdaten. Zur Dekodierung der Standard-Alarme stellt SEW-EURODRIVE den Hilfsfunktionsbaustein ALARM (siehe Kapitel "Hilfsfunktionsbaustein ALARM") bereit.

HINWEIS

Enthält ein Alarmtelegramm mehr Bytes als im Register "Daten" definiert wurden, wird nur die Anzahl der definierten Bytes übernommen. Der Rest wird abgeschnitten.

Alarm-daten	Beschreibung
Byte 0	Länge der Alarmmeldung in Byte (4 – 126)
Byte 1	Kennung für den Alarmtyp: <ul style="list-style-type: none"> • 1: Diagnosealarm • 2: Prozessalarm • 3: Ziehenalarm • 4: Steckenalarm • 5: Statusalarm • 6: Updatealarm • 31: Ausfall einer Erweiterung eines Controllers oder IO-Device • 32 – 126: Herstellerspezifisch Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.
Byte 2	Steckplatznummer der Alarm auslösenden Komponente.
Byte 3	<ul style="list-style-type: none"> • 0: keine weitere Information • 1: ankommender Alarm, Slot gestört • 2: ausgehender Alarm, Slot nicht mehr gestört • 3: ausgehender Alarm, Slot weiterhin gestört
Byte 4 bis 126	Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.

HINWEIS

Die Struktur der Standard-Alarme (Bytes 0 – 3 ist normiert und für alle Hersteller identisch. Für die herstellerspezifisch genutzten Bytes 4 – 126, siehe Gerätehandbuch des PROFINET-IO-Device.

So bedienen Sie den Funktionsbaustein RALRM:

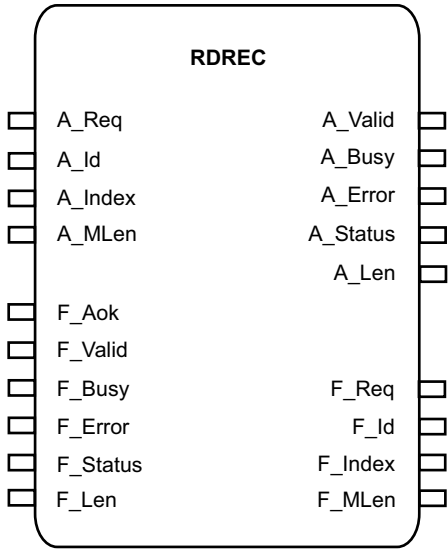
1. Im Anwenderprogramm den Eingang **A_Mlen** auf die Anzahl der maximal zu erwartenden Alarmdaten in Bytes definieren. Während des Betriebs kann **A_Mlen** nicht geändert werden.
2. Im Anwenderprogramm den Eingang **A_Ena** auf TRUE setzen. Im Gegensatz zu den anderen Funktionsbausteinen ist der Funktionsbaustein RALRM nur aktiv, solange der Eingang **A_Ena** TRUE ist.

Wurde der Baustein erfolgreich gestartet, dann geht der Ausgang **A_Eno** auf TRUE. Konnte der Baustein nicht gestartet werden, wird ein Fehlercode am Ausgang **A_Status** ausgegeben.

Trifft ein neuer Alarm ein, geht der Ausgang *A_New* für mindestens einen Zyklus auf TRUE. Für diese Zeit enthalten die Ausgänge die Alarmdaten des Alarm auslösenden IO-Device, die ausgewertet werden können. Danach geht der Ausgang *A_New* wieder für mindestens einen Zyklus auf FALSE. Die Ausgänge *A_ID* und *A_Len* werden auf Null zurückgesetzt, bevor die nächste Alarmmeldung empfangen und ausgewertet werden kann.

5.10.3 Funktionsbaustein RDREC

Das folgende Bild zeigt den Aufbau des Funktionsbausteins RDREC mit allen Eingängen (links) und Ausgängen (rechts).



7436364427

Der Funktionsbaustein RDREC dient zum azyklischen Lesen eines am Eingang *A_Index* adressierten Datensatzes von einem IO-Device. Welche Daten gelesen werden können, muss der Betriebsanleitung des IO-Device entnommen werden.

Diese Funktion ist optional. Im PROFINET-IO-Controller können gleichzeitig bis zu 32 RDREC- und/oder WRREC-Bausteine aktiv sein.

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine")

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 die Anforderung zum Lesen.	BOOL
A_ID	Identifikationsnummer des IO-Device (siehe Kap. "PROFINET-IO-Hilfsbausteine").	DWORD
A_Index	Datensatznummer des zu lesenden Datensatzes. Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.	INT

21233780 / DE – 07/2014

A-Eingänge	Beschreibung	Typ
A_MLen	Maximal erwartete Länge der zu lesenden Daten in Bytes.	INT

A-Ausgänge	Beschreibung	Typ
A_Valid	Eine neue Diagnosemeldung wurde empfangen und ist gültig.	BOOL
A_Busy	TRUE: Das Lesen ist noch nicht beendet.	BOOL
A_Error	TRUE: Beim Lesen trat ein Fehler auf. FALSE: Kein Fehler.	BOOL
A_Status	Status oder Fehlercode (siehe Kapitel "Fehlercodes der Funktionsbausteine").	DWORD
A_Len	Länge der empfangenen Diagnosedaten in Bytes.	INT

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix "F"

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

HINWEIS



Die Verbindung des Funktionsbausteins RDREC im Strukturbaum (im Ordner "Funktionsbausteine") mit dem Funktionsbaustein RDREC (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor im Variableneditor erstellt werden.

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

F-Eingänge	Typ
F_ACK	BOOL
F_VALID	BOOL
F_BUSY	BOOL
F_ERROR	BOOL
F_STATUS	DWORD
F_LEN	INT

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

F-Ausgänge	Typ
F_REQ	BOOL
F_ID	DWORD
F_INDEX	INT
F_MLEN	INT

So erstellen Sie den Funktionsbaustein RDREC im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [Funktionsbausteine] / [Neu] öffnen.

2. Funktionsbaustein RDREC wählen.
3. Rechtsklick auf Funktionsbaustein RDREC und Menüpunkt [Edit] wählen. Die Variablenzuweisung zum Funktionsbaustein wird geöffnet.
4. Die Eingänge des Funktionsbausteins RDREC im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins RDREC im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	DWORD
INDEX	INT
MLEN	INT
REQ	BOOL

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

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

Daten	Beschreibung
Es sind keine Variablen vorgegeben.	Im Register "Prozessvariablen" kann eine beliebige Datenstruktur definiert werden, die allerdings auf die Struktur des Datensatzes passen muss. Die Struktur des Datensatzes muss aus der Bedienungsanleitung des Herstellers des IO-Device entnommen werden.

Funktionsbaustein RDREC bedienen

So bedienen Sie den Funktionsbaustein RDREC:

1. Im Anwenderprogramm die IO-Device-Adresse an den Eingang *A_ID* setzen.
2. Im Anwenderprogramm den IO-Device-spezifischen Index für den Datensatz (siehe Handbuch des Herstellers) am Eingang *A_Index* setzen.
3. Im Anwenderprogramm die Länge des zu lesenden Datensatzes am Eingang *A_Len* setzen.
4. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen. Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

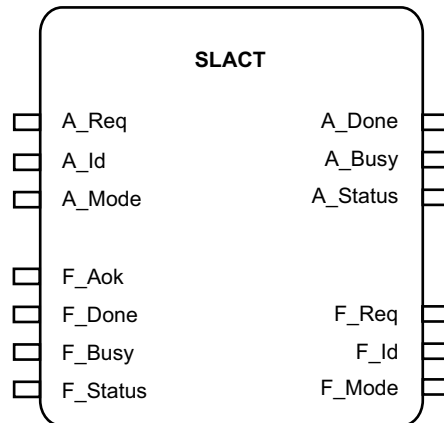
Der Ausgang *A_Busy* ist TRUE, bis die Datensatzanforderung abgearbeitet ist. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Valid* oder *A_Error* auf TRUE.

Ist der Datensatz gültig, geht der Ausgang *A_Valid* auf TRUE. Der Datensatz kann über die im Register "Daten" definierten Variablen ausgewertet werden. Der Ausgang *A_Len* enthält die tatsächliche Länge des ausgelesenen Datensatzes.

Konnte der Datensatz nicht erfolgreich gelesen werden, dann wechselt der Ausgang *A_Error* auf TRUE und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

5.10.4 Funktionsbaustein SLACT

Das folgende Bild zeigt den Aufbau des Funktionsbausteins SLACT mit allen Eingängen (links) und Ausgängen (rechts).



7436368651

Der Funktionsbaustein SLACT dient zum Aktivieren und Deaktivieren eines IO-Device aus dem Anwenderprogramm des PROFINET-IO-Controllers. Somit ist es möglich, das IO-Device durch einen mechanischen Schalter an einem physikalischen Eingang des PROFINET-IO-Controllers oder durch einen Timer in einen der folgenden Zustände zu setzen.

- $\neq 0$: Aktiv
- 0: Nicht aktiv

Das Anwenderprogramm ist bei Verwendung mehrerer Funktionsbausteine SLACT so anzulegen, dass immer nur ein Funktionsbaustein SLACT aktiv ist.

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

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 des IO-Device (siehe Kap. "PROFINET-IO-Hilfsbausteine").	DWORD
A_Mode	Zustand, in den das PROFINET-IO-Device gesetzt werden soll: <ul style="list-style-type: none"> • $\neq 0$: Aktiv (verbunden) • 0: Nicht aktiv (deaktiviert) 	INT

A-Ausgänge	Beschreibung	Typ
A_Done	TRUE: Das PROFINET-IO-Device wurde in den am Eingang <i>A_Mode</i> definierten Zustand gesetzt.	BOOL
A_Busy	TRUE: Das Setzen des PROFINET-IO-Device ist noch nicht beendet.	BOOL
A_Status	Status oder Fehlercode (siehe Kapitel "Fehlercodes der Funktionsbausteine").	DWORD

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix "F"

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

HINWEIS



Die Verbindung des Funktionsbausteins SLACT im Strukturbaum (im Ordner "Funktionsbausteine") mit dem Funktionsbaustein SLACT (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor im Variableneditor erstellt werden.

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

F-Eingänge	Typ
F_ACK	BOOL
F_DONE	BOOL
F_BUSY	BOOL
F_STATUS	DWORD

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

F-Ausgänge	Typ
F_REQ	BOOL
F_ID	DWORD
F_MODE	INT

So erstellen Sie den Funktionsbaustein SLACT im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [Funktionsbausteine] / [Neu] öffnen.
2. Funktionsbaustein SLACT wählen.
3. Rechtsklick auf Funktionsbaustein SLACT und Menüpunkt [Edit] wählen. Die Variablenzuweisung zum Funktionsbaustein wird geöffnet.
4. Die Eingänge des Funktionsbausteins SLACT im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins SLACT im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	DWORD

Eingänge	Typ
MODE	INT
REQ	BOOL

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

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Funktionsbaustein SLACT bedienen

So bedienen Sie den Funktionsbaustein SLACT:

1. Im Anwenderprogramm den Eingang *A_Mode* auf den gewünschten Zustand setzen.
2. Im Anwenderprogramm den Identifier mit der IO-Device-Adresse am Eingang *A_ID* setzen.
3. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen. Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

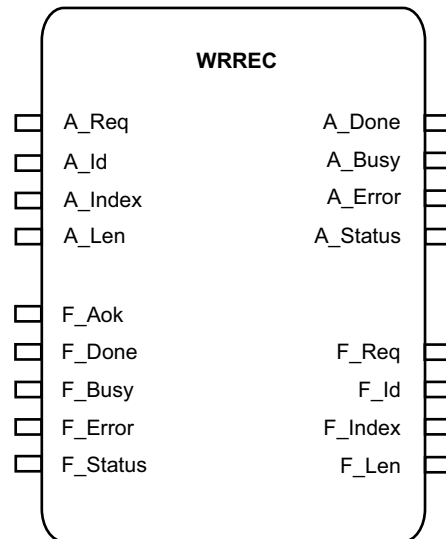
Der Ausgang *A_Busy* ist TRUE, bis der SLACT-Befehl abgearbeitet ist. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Done* auf TRUE.

Am Ausgang *A_Status* wird der Slave-Mode ausgegeben, wenn der Slave-Mode gesetzt werden konnte.

Am Ausgang *A_Status* wird ein Fehlercode ausgegeben, wenn der Slave-Mode nicht gesetzt werden konnte.

5.10.5 Funktionsbaustein WRREC

Das folgende Bild zeigt den Aufbau des Funktionsbausteins WRREC mit allen Eingängen (links) und Ausgängen (rechts).



7436372875

Der Funktionsbaustein WRREC dient zum azyklischen Schreiben eines mit *A_Index* adressierten Datensatzes an ein IO-Device. Welche Daten geschrieben werden können, muss der Betriebsanleitung des IO-Device entnommen werden.

Im PROFINET-IO-Controller können gleichzeitig bis zu 32 RDREC- und / oder WRREC-Bausteine aktiv sein.

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

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 die Anforderung zum Schreiben eines Datensatzes.	BOOL
A_ID	Identifikationsnummer des IO-Device (siehe Kap. "PROFINET-IO-Hilfsbausteine").	DWORD
A_Index	Datensatznummer des zu schreibenden Datensatzes. Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.	INT
A_Len	Länge des zu schreibenden Datensatzes in Bytes.	INT
A-Ausgänge	Beschreibung	Typ
A_Done	TRUE: Das PROFINET-IO-Device wurde in den am Eingang <i>A_Mode</i> definierten Zustand gesetzt.	BOOL

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Das Setzen des PROFINET-IO-Device ist noch nicht beendet.	BOOL
A_Status	Status oder Fehlercode (siehe Kapitel "Fehlercodes der Funktionsbausteine").	DWORD
A_Status	Status oder Fehlercode (siehe Kapitel "Fehlercodes der Funktionsbausteine").	DWORD

Ein- und Ausgänge des Funktionsbausteins mit dem Präfix "F"

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

HINWEIS



Die Verbindung des Funktionsbausteins WRREC im Strukturbaum (im Ordner "Funktionsbausteine") mit dem Funktionsbaustein WRREC (im Anwenderprogramm) erfolgt über gemeinsame Variablen. Diese müssen zuvor im Variableneditor erstellt werden.

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

F-Eingänge	Typ
F_ACK	BOOL
F_DONE	BOOL
F_ERROR	BOOL
F_BUSY	BOOL
F_STATUS	DWORD

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

F-Ausgänge	Typ
F_REQ	BOOL
F_ID	DWORD
F_INDEX	INT
F_LEN	INT

So erstellen Sie den Funktionsbaustein WRREC im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET-IO-Controller] / [Funktionsbausteine] / [Neu] öffnen.
2. Funktionsbaustein WRREC wählen.
3. Rechtsklick auf Funktionsbaustein WRREC und Menüpunkt [Edit] wählen. Die Variablenzuweisung zum Funktionsbaustein wird geöffnet.
4. Die Eingänge des Funktionsbausteins WRREC im Strukturbaum mit den gleichen Variablen verbinden, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins WRREC im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	DWORD
INDEX	INT
LEN	INT
REQ	BOOL

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

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
DONE	BOOL
ERROR	BOOL
STATUS	DWORD

Daten	Beschreibung
Es sind keine Variablen vorgegeben.	Im Register "Prozessvariablen" kann eine beliebige Datenstruktur definiert werden, die allerdings auf die Struktur des Datensatzes passen muss. Die Struktur des Datensatzes muss aus der Bedienungsanleitung des Herstellers des IO-Device entnommen werden.

Funktionsbaustein WRREC bedienen

So bedienen Sie den Funktionsbaustein WRREC:

1. Im Anwenderprogramm die IO-Device-Adresse an den Eingang *A_ID* setzen.
2. Im Anwenderprogramm den IO-Device-spezifischen Index für den Datensatz (siehe Handbuch des Herstellers) am Eingang *A_Index* setzen.
3. Im Anwenderprogramm die Länge des zu schreibenden Datensatzes am Eingang *A_Len* setzen.
4. Im Anwenderprogramm den Datensatz, wie im Register "Daten" definiert, einstellen.
5. Im Anwenderprogramm den Eingang *A_Req* auf TRUE setzen. Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis der Datensatz geschrieben ist. Danach wechseln die Ausgänge *A_Busy* auf FALSE und *A_Done* auf TRUE.

Konnte der Datensatz nicht erfolgreich geschrieben werden, wechselt der Ausgang *A_Error* auf TRUE und ein Fehlercode wird am Ausgang *A_Status* ausgegeben.

5.11 Konfiguration der Funktionsbausteine

Die Feldbus-Protokolle und die zugehörigen Funktionsbausteine laufen auf dem COM-Modul der Sicherheitssteuerung. 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, 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.

5.11.1 Hinzufügen der Funktionsbausteinbibliotheken

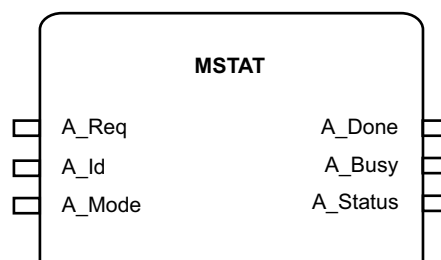
Die Funktionsbausteinbibliotheken für PROFINET und Send & Receive TCP (PROFIlib.A3 und *TCPIlib.A3*) müssen über die Funktion „Wiederherstellen...“ (Kontextmenü des Projekts) dem Projekt hinzugefügt werden. Die Funktionsbausteinbibliotheken sind auf der SILworX®-Installations-CD (unter „Binaries/Communic-Libs“) vorhanden. Von dort können sie per "Wiederherstellen" eingefügt werden.

5.11.2 Konfiguration der Funktionsbausteine im Anwenderprogramm

Die benötigten Funktionsbausteine können per Drag-and-Drop in das Anwenderprogramm kopiert werden. Konfigurieren Sie die Eingänge und Ausgänge nach der Beschreibung des jeweiligen Funktionsbausteins.

Oberer Teil des Funktionsbausteins

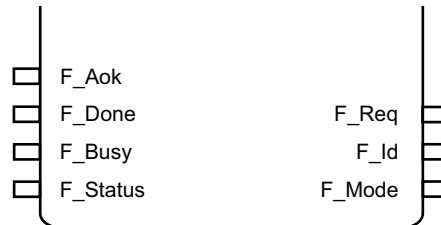
Der obere Teil des Funktionsbausteins entspricht der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird. Hier werden die Variablen verbunden, die im Anwenderprogramm verwendet werden. Das Präfix "A" steht für "Applikation".



7441757323

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".



7441760779

5.11.3 Konfiguration der Funktionsbausteine im Strukturbaum von SILworX®

So konfigurieren Sie den Funktionsbaustein im Strukturbaum von SILworX®:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle].
2. Klicken Sie mit der rechten Maustaste auf Funktionsbausteine und wählen Sie [Neu].
3. Wählen Sie den passenden Funktionsbaustein im Strukturbaum von SILworX® aus.



12843489675

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				
	Name	Datentyp	Eingangsvariable	Globale Variable
1	ACK	BOOL	<input checked="" type="checkbox"/>	
2	BUSY	BOOL	<input checked="" type="checkbox"/>	
3	DONE	BOOL	<input checked="" type="checkbox"/>	
4	ID	DWORD	<input type="checkbox"/>	
5	MODE	INT	<input type="checkbox"/>	
6	REQ	BOOL	<input type="checkbox"/>	
7	STATUS	DWORD	<input checked="" type="checkbox"/>	

12845617035

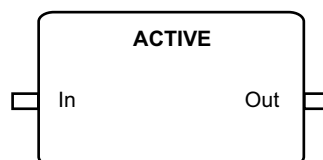
5.12 PROFINET-IO-Hilfsfunktionsbausteine

Die Hilfsfunktionsbausteine werden zur Parametrierung und Auswertung der Ein- und Ausgänge der Funktionsbausteine verwendet. Folgende Hilfsfunktionsbausteine stehen zur Verfügung.

Hilfsfunktionsbaustein	Beschreibung der Funktion
ACTIVE	Prüft, ob das PROFINET-IO-Device aktiv oder inaktiv ist.
ALARM	Dekodieren der Alarmedaten.
DEID	Identifikationsnummer dekodieren.
ID	Generiert aus vier Bytes einen Identifier.
NSLOT	Fortlaufende Identifikationsnummer für die Slots erstellen.
SLOT	SLOT-Identifikationsnummer mit Slot-Nummer erstellen.
STDDIAG	Standarddiagnose eines Slaves dekodieren.
LATCH	Wird nur innerhalb anderer Funktionsbausteine verwendet.
PIG	Wird nur innerhalb anderer Funktionsbausteine verwendet.
PIGII	Wird nur innerhalb anderer Funktionsbausteine verwendet.

5.12.1 Hilfsfunktionsbaustein ACTIVE

Der Hilfsfunktionsbaustein ACTIVE ermittelt aus der Standarddiagnose eines PROFINET-IO-Device, ob das Device gerade aktiv oder inaktiv ist.



7441573387



HINWEIS

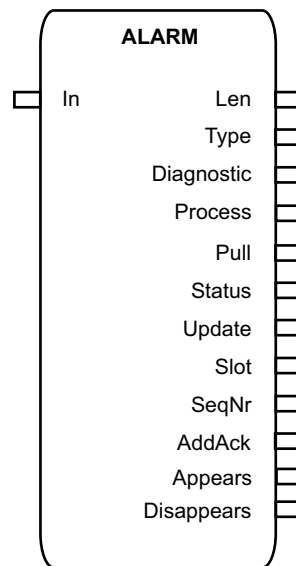
Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

Eingang	Beschreibung	Typ
IN	Standarddiagnose des Slave.	DWORD

Ausgang	Beschreibung	Typ
OUT	TRUE: Slave ist aktiv FALSE: Slave ist inaktiv	BOOL

5.12.2 Hilfsfunktionsbaustein ALARM

Der Hilfsfunktionsbaustein ALARM dekodiert die Standard-Alarmdaten eines PROFINET-IO-Devices.



7440806667



HINWEIS

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

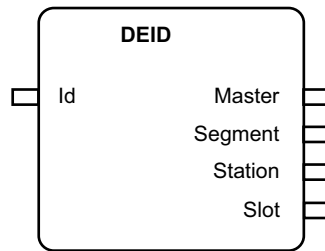
Eingang	Beschreibung	Typ
IN	Standardalarm	DWORD

Ausgänge	Beschreibung	Typ
LEN	Länge der gesamten Alarmmeldung.	SINT

Ausgänge	Beschreibung			Typ
Type	1: Diagnosealarm 2: Prozessalarm 3: Ziehenalarm 4: Steckenalarm 5: Statusalarm 6: Update-Alarm Andere Nummern sind entweder reserviert oder herstellerspezifisch. Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.			SINT
Diagnostic	True = Diagnosealarm			BOOL
Process	True = Prozessalarm			BOOL
Pull	True = Modul wurde gezogen			BOOL
Plug	True = Modul wurde wieder gesteckt			BOOL
Status	True = Status-Alarm			BOOL
Update	True = Update-Alarm			BOOL
Slot	Alarmauslösendes Modul			BYTE
SeqNr	Alarm-Sequenznummer			SINT
AddAck	TRUE bedeutet, dass der Slave, der diesen Alarm ausgelöst hat, eine zusätzliche Bestätigung durch die Anwendung erwartet. Welche genau, muss dem Slave-Handbuch des Herstellers entnommen werden.			BOOL
Appears Disap- pears	Aus- gang	Wert	Beschreibung	BOOL
	Appears	TRUE	Sind beide FALSE, dann ist bis zu diesem Zeitpunkt kein Fehler aufgetreten.	BOOL
	Disap- pears	FALSE		
	Appears	TRUE	Ein Fehler ist aufgetreten und steht noch an.	
	Disap- pears	FALSE		
	Appears	TRUE	Ein Fehler war aufgetreten und verschwindet gerade.	
	Disap- pears	FALSE		
	Appears	TRUE	Sind beide TRUE, dann verschwindet der Fehler, der Slave ist aber weiterhin gestört.	
	Disap- pears	FALSE		

5.12.3 Hilfsfunktionsbaustein DEID

Der Hilfsfunktionsbaustein DEID dekodiert die Identifikationsnummer und zerlegt diese in ihre vier Bestandteile.



7440815371

HINWEIS



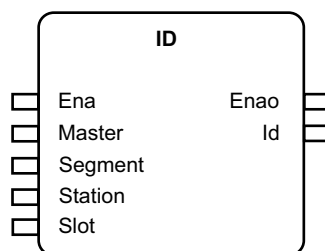
Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

Eingang	Beschreibung	Typ
ID	Identifikationsnummer des Slaves.	DWORD

Ausgänge	Beschreibung	Typ
Master	Busadresse des Master.	BYTE
Segment	Segment	BYTE
Station	Busadresse des Slave	BYTE
Slot	Slot- oder Moudl-Nummer	BYTE

5.12.4 Hilfsfunktionsbaustein ID

Der Hilfsfunktionsbaustein ID generiert aus vier Bytes einen Identifier (Identifikationsnummer), die von anderen Hilfsfunktionsbausteinen genutzt wird.



7441561227

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

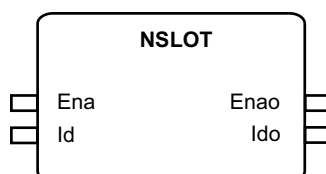
Eingänge	Beschreibung	Typ
Ena	Nicht genutzt.	BOOL
Master	Busadresse.	BYTE

Eingänge	Beschreibung	Typ
Segment	Segment.	BYTE
Station	Busadresse des Slave.	BYTE
Slot	Slot- oder Modulnummer.	BYTE

Ausgänge	Beschreibung	Typ
Enao	Nicht genutzt.	BOOL
ID	Identifikationsnummer des Slave.	DWORD

5.12.5 Hilfsfunktionsbaustein NSLOT

Der Hilfsfunktionsbaustein NSLOT generiert aus einem Identifier einen neuen Identifier, der den nächsten Slot im gleichen Slave adressiert. Der Eingang *Ena* muss auf TRUE gesetzt werden, damit der Hilfsfunktionsbaustein läuft. Der Ausgang *Enao* ist TRUE, wenn das Ergebnis am Ausgang *Ido* gültig ist.



7441569931

HINWEIS



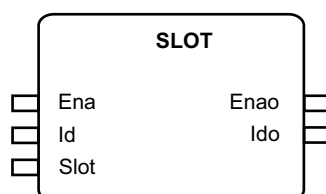
Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

Eingänge	Beschreibung	Typ
Ena	Solange TRUE anliegt, läuft der Baustein.	BOOL
ID	Identifikationsnummer des Slaves.	DWORD

Ausgänge	Beschreibung	Typ
Enao	TRUE = Ergebnis gültig FALSE = Keine weiteren Slot-Nummern	BOOL
Ido	Identifikationsnummer des Slaves.	DWORD

5.12.6 Hilfsfunktionsbaustein SLOT

Der Hilfsfunktionsbaustein SLOT generiert aus einem Identifier und einer Slot-Nummer einen neuen Identifier, der den gleichen Slave adressiert wie der alte Identifier, jedoch mit der neuen Slot-Nummer.



7441564683

HINWEIS



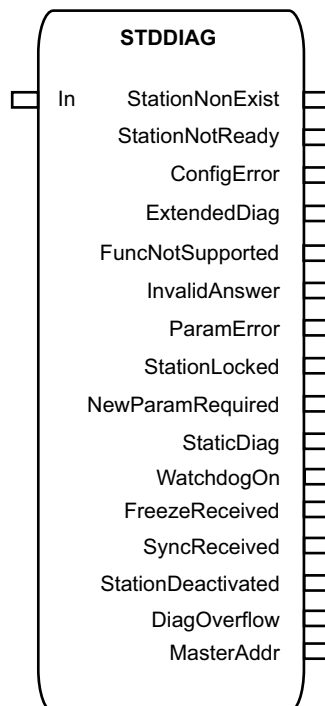
Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

Eingänge	Beschreibung	Typ
Ena	Nicht genutzt.	BOOL
ID	Logische Adresse der Slave-Komponente (Slave-ID und Slot-Nummer).	DWORD
Slot	Neue Slot- oder Modul Nummer.	BYTE

Ausgänge	Beschreibung	Typ
Enao	Nicht genutzt.	BOOL
Ido	Identifikationsnummer des Slaves.	DWORD

5.12.7 Hilfsfunktionsbaustein STDDIAG

Der Hilfsfunktionsbaustein Standard-Diagnose STDDIAG dekodiert die Standarddiagnose eines PROFINET-IO-Devices. Die Ausgänge vom Typ BOOL des Funktionsbaustein STDDIAG sind TRUE, wenn das dazugehörige Bit in der Standard-Diagnose gesetzt ist.



7440811915

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe Kapitel "Konfiguration der Funktionsbausteine").

Eingang	Beschreibung	Typ
IN	Standarddiagnose des Slave.	DWORD

Ausgänge	Beschreibung	Typ
StauionNonExist	Slave existiert nicht.	BOOL
StationNotReady	Slave nicht bereit.	BOOL
ConfigError	Konfigurationsfehler.	BOOL
ExtendedDiag	Erweiterte Diagnose folgt.	BOOL
FuncNotSupported	Funktion nicht unterstützt.	BOOL
InvalidAnswer	Ungültige Antwort vom Slave.	BOOL
ParamError	Parametrierfehler.	BOOL
StationLocked	Slave vom anderen Master gesperrt.	BOOL
NewParamRequired	Neue Parametrierdaten erforderlich.	BOOL
StaticDiag	Statische Diagnose.	BOOL
WatchdogOn	Watchdog aktiviert.	BOOL
FreezeReceived	Freeze-Kommando erhalten.	BOOL
SyncReceived	Sync-Kommando erhalten.	BOOI
StationDeactivated	Slave wurde deaktiviert.	BOOL
DiagOverflow	Diagnoseüberlauf.	BOOL
MasterAdd	Busadresse des Masters.	BYTE

So lesen Sie die Standard-Diagnose des PROFINET-IO-Device aus:

1. Wählen Sie im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [PROFINET IO Controller].
2. Klicken Sie mit der rechten Maustaste auf PROFINET IO Device und wählen Sie [Edit].
3. Ziehen Sie die globale Variable vom Typ DWORD auf das Feld *Standard-Diagnose*. Wählen Sie den passenden Funktionsbaustein im Strukturbaum von SILworX® aus.
4. Verbinden Sie die globale Variable mit dem Eingang des Funktionsbausteins STDDIAG.

5.13 Fehlercodes der Funktionsbausteine

Wenn ein Funktionsbaustein ein Kommando nicht korrekt ausführen konnte, wird am Ausgang *A_Status* ein Fehlercode ausgegeben. Die Bedeutung des Fehlercodes kann der folgenden Tabelle entnommen werden.

Fehlercode	Symbol	Beschreibung
16#40800800	TEMP_NOT_AVAIL	Dienst steht vorübergehend nicht zur Verfügung.
16#40801000	INVALID_PARA	Ungültiger Parameter.
16#40801100	WRONG_STATE	Slave unterstützt kein PROFINET.
16#40808000	FATAL_ERR	Fataler Programmfehler.

Fehlercode	Symbol	Beschreibung
16#40808100	BAD_CONFIG	Konfigurationsfehler im Datenbereich.
16#40808200	PLC_STOPPED	Steuerung wurde gestoppt.
16#4080A000	READ_ERR	Fehler beim Lesen eines Records.
16#4080A100	WRITE_ERR	Fehler beim Schreiben eines Records.
16#4080A200	MODULE_FAILURE	Fehler nicht näher spezifizierbar.
16#4080B000	INVALID_INDEX	Index ist ungültig.
16#4080B100	WRITE_LENGTH	Falsche Länge beim Schreiben.
16#4080B200	INVALID_SLOT	Slot-Nummer ist ungültig.
16#4080B300	TYPE_CONFLICT	Falscher Typ.
16#4080B400	INVALID_AREA	Falscher Lese- oder Schreibbereich.
16#4080B500	STATE_CONFLICT	Master im falschen Zustand.
16#4080B600	ACCESS_DENIED	Slave nicht aktiv (oder ähnliches).
16#4080B700	INVALID_RANGE	Falscher Lese- oder Schreibbereich.
16#4080B800	INVALID_PARAMETER	Falscher Parameterwert.
16#4080B900	INVALID_TYPE	Falscher Parametertyp.
16#4080C300	NO_RESOURCE	Slave nicht vorhanden.
16#4080BA00	BAD_VALUE	Ungültiger Wert.
16#4080BB00	BUS_ERROR	Busfehler.
16#4080BC00	INVALID_SLAVE	Ungültige Slave-ID.
16#4080BD00	TIMEOUT	Timeout aufgetreten.
16#4080C000	READ_CONSTRAIN	Lesebeschränkung.
16#4080C100	WRITE_CONSTRAIN	Schreibbeschränkung.
16#4080C200	BUSY	Ein Baustein dieser Art ist bereits aktiv.
16#4080C300	NO_RESOURCE	Slave nicht aktiv.

6 Modbus TCP/UDP

6.1 Modbus-Master

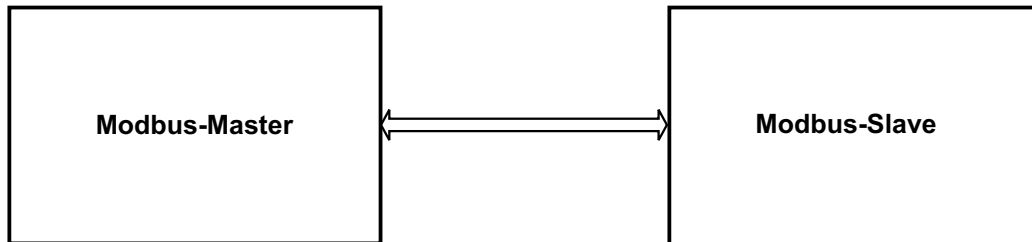
Die Datenübertragung zwischen dem Modbus-Master und den Modbus-Slaves erfolgt über TCP/UDP (Ethernet).

Die folgende Tabelle zeigt Ihnen die Eigenschaften des Modbus-Masters:

Eigenschaft	Beschreibung
Modbus Master	Es kann pro COM-Modul / Steuerung ein Modbus-Master konfiguriert werden. Der Modbus-Master kann simultan mit TCP/UDP-Slaves Daten austauschen.
Max. Anzahl Modbus Slaves	Ein Modbus-Master kann bis zu 247 Slaves bedienen. <ul style="list-style-type: none"> • 64 TCP Slaves über TCP/IP-Verbindung • 247 UDP Slaves über UDP/IP-Verbindung Die maximale Anzahl UDP-Slaves ist limitiert, da die Slaves auf der Masterseite verwaltet werden müssen.
Max. Anzahl Anforderungstelegramme	Es können bis zu 988 Anforderungstelegramme pro Modbus-Master konfiguriert werden.
Max. Prozessdatenlänge pro Anforderungstelegramm	Die Prozessdatenlänge beträgt bei SEW-spezifischen Anforderungstelegrammen 1100 Byte, siehe Kapitel "SEW spezifische Funktionscodes".
Max. Größe der Sendedaten	64 kB senden 64 kB empfangen
Max. Größe der Empfangsdaten	Hinweis: Die Statusbytes des Masters und die Statusbytes von jedem zugeordneten Slave müssen von der max. Größe der Sendedaten subtrahiert werden.
Darstellungsformat der Modbus-Daten	Die Sicherheitssteuerung verwendet das Big-Endian-Format. Beispiel: 32 Bit Daten (z. B. DWORD, DINT):
	32 Bit Daten (hex)0x12345678
	Speicher-Offset0123
	Big Endian12345678
	Middle Endian56781234
	Little Endian78563412

6.1.1 Modbus-Beispiel

In diesem Beispiel tauscht ein Modbus-Master Daten mit einem Modbus-Slave über Modbus TCP aus. Die beiden Steuerungen werden über die Ethernet-Schnittstellen der Kommunikationsmodule verbunden.



5452807563

HINWEIS



Befinden sich der Modbus-Master und der Modbus-Slave in verschiedenen Subnetzen, müssen in der Routing-Tabelle die entsprechenden benutzerdefinierten Routen eingetragen werden.

Folgende Globale Variablen müssen Sie für dieses Beispiel in SILworX® anlegen.

Globale Variablen	Typ
Master->Slave_BOOL_00	BOOL
Master->Slave_BOOL_01	BOOL
Master->Slave_BOOL_02	BOOL
Master->Slave_WORD_00	WORD
Master->Slave_WORD_01	WORD
Slave->Master_WORD_00	WORD
Slave->Master_WORD_01	WORD

Konfiguration des Modbus-TCP-Slave

So legen Sie einen neuen Modbus-Slave an:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] öffnen.
2. Um einen neuen Modbus-Slave-Set hinzuzufügen, wählen Sie im Kontextmenü von Protokolle den Menüpunkt [Neu] / [Modbus-Slave-Set].
3. Im Kontextmenü des Modbus-Slave-Set [Edit] wählen und [Eigenschaften Modbus-Slave-Set] öffnen, Standardwerte beibehalten.

Register "Modbus-Slave" wählen und folgende Einstellungen vornehmen:

- [COM Modul] wählen
- [TCP aktivieren] aktivieren
- Die restlichen Parameter behalten die Standardwerte.

*Bit-Eingangsvariablen des Modbus-Slave konfigurieren***HINWEIS**

Im Register "Bitvariablen" sind die Booleschen Variablen einzutragen, die der Master bitweise adressiert (Funktionscode 1, 2, 5, 15).

So konfigurieren Sie die Bit-Eingangsvariablen des Modbus-Slave:

1. Im Kontextmenü des Modbus-Slave [Edit] / [Bitvariablen] wählen.
2. In der Objektauswahl die folgenden globalen Variablen auswählen und diese per Drag-and-Drop in den Bereich "Bit Eingänge" ziehen.

Bit Adresse	Bitvariable	Typ
0	Master->Slave_BOOL_00	BOOL
1	Master->Slave_BOOL_01	BOOL
2	Master->Slave_BOOL_02	BOOL

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

Register-Eingangsvariablen des Modbus-Slave konfigurieren

So konfigurieren Sie die Register Eingangsvariablen des Modbus-Slave:

HINWEIS

Im Register „Registervariablen“ sind die Variablen einzutragen, die der Master registerweise adressiert (Funktionscode 3, 4, 6, 16 ,23).

1. Im Kontextmenü des Modbus-Slave [Edit] / [Registervariablen] wählen.
2. In der Objektauswahl die folgenden Variablen auswählen und diese per Drag-and-Drop in den Bereich "Register Eingänge" ziehen.

Bit Adresse	Registervariablen	Typ
0	Master->Slave_WORD_00	WORD
1	Master->Slave_WORD_01	WORD

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

Register-Ausgangsvariablen des Modbus-Slave konfigurieren

So konfigurieren Sie die Register Ausgangsvariablen des Modbus-Slave:

HINWEIS

Im Register „Registervariablen“ sind die Variablen einzutragen, die der Master registerweise adressiert (Funktionscode 3, 4, 6, 16 ,23).

1. Im Kontextmenü des Modbus-Slave [Edit] / [Registervariablen] wählen.
2. In der Objektauswahl die folgenden Variablen auswählen und diese per Drag-and-Drop in den Bereich "Register Ausgänge" ziehen.

Bit Adresse	Registervariablen	Typ
0	Slave → Master_WORD_00	WORD
1	Slave → Master_WORD_01	WORD

3. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Register Ausgänge" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Modbus-TCP-Slave-Konfiguration prüfen

So prüfen Sie die Modbus-TCP-Slave-Konfiguration:

1. Kontextmenü des Modbus-TCP-Slave öffnen und [Verifikation] wählen.
2. Einträge im Logbuch sorgfältig überprüfen, gegebenenfalls korrigieren.

Konfiguration des Modbus-TCP-Master

So legen Sie den Modbus Master an:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] öffnen.
2. Im Kontextmenü von Protokolle [Neu] / [Modbus Master] wählen um einen neuen Modbus Master hinzuzufügen.
3. Im Kontextmenü vom Modbus-Master [Eigenschaften] / [Allgemein] wählen.
4. [COM Modul] auswählen.

Die restlichen Parameter behalten die Standardwerte.

Verbindung vom Modbus-Master zum Modbus-TCP-Slave erstellen

So erstellen Sie im Modbus-Master die Verbindung zum Modbus-TCP-Slave:

1. Im Strukturbaum [Ressource] / [Protokolle] / [Modbus Master] / [Ethernet Slaves] öffnen.
2. Rechtsklick auf [Ethernet Slaves] und im Kontextmenü [Neu] wählen.
3. Aus der Liste "TCP/UDP-Slave" wählen und mit [OK] bestätigen.

Konfiguration des TCP/UDP-Slave im Modbus-Master:

- [Edit] zum Zuweisen der Systemvariablen wählen, siehe Kapitel "Systemvariablen Gateway-Slave".
- [Eigenschaften] zum Konfigurieren der Eigenschaften wählen, siehe Kapitel "Eigenschaften Gateway-Slave".

In den Eigenschaften des Slaves die IP Adresse des TCP/UDP-Slaves eintragen.

Die restlichen Parameter behalten die Standardwerte.

Konfigurieren des Anforderungstelegramms zum Schreiben der Bit-Ausgangsvariablen

So konfigurieren Sie das Anforderungstelegramm zum Schreiben der Bit Ausgangsvariablen:

1. Rechtsklick auf TCP/UDP-Slave und im Kontextmenü [Neu] wählen.
2. Aus der Liste das Anforderungstelegramm "Write Multiple Coils (15)" wählen.
3. Rechtsklick auf das Anforderungstelegramm "Write Multiple Coils (15)" und im Kontextmenü [Eigenschaften wählen].

Als Startadresse des Schreibbereichs "0" eintragen.

4. Rechtsklick auf das Anforderungstelegramm "Write Multiple Coils (15)" und im Kontextmenü "Edit" wählen.
5. In der Objektauswahl die folgenden Variablen auswählen und diese per Drag-and-Drop in das Register "Ausgangsvariablen" ziehen.

Offset	Bit Variable	Typ
0	Master->Slave_BOOL_00	BOOL
1	Master->Slave_BOOL_01	BOOL
2	Master->Slave_BOOL_02	BOOL

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Konfigurieren des Anforderungstelegramms zum Schreiben der Register-Ausgangsvariablen

So konfigurieren Sie das Anforderungstelegramm zum Schreiben der Register Ausgangsvariablen:

1. Rechtsklick auf TCP/UDP-Slave und im Kontextmenü [Neu] wählen.
2. Aus der Liste das Anforderungstelegramm "Write Multiple Registers (16)" wählen.
3. Rechtsklick auf das Anforderungstelegramm "Write Multiple Registers (16)" und im Kontextmenü [Eigenschaften] wählen.

Als Startadresse des Lesebereichs "0" eintragen.

4. Rechtsklick auf das Anforderungstelegramm "Write Multiple Registers (16)" und im Kontextmenü [Edit] wählen.
5. In der Objektauswahl die folgenden Variablen auswählen und diese per Drag-and-Drop in das Register "Ausgangsvariablen" ziehen.

Offset	Register Variablen	Typ
0	Master->Slave_WORD_00	WORD
1	Master->Slave_WORD_01	WORD

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Definieren des Anforderungstelegramms im Modbus-Master zum Lesen der Eingangsvariablen

So definieren Sie im Modbus-Master das Anforderungstelegramm zum Lesen der Eingangsvariablen:

1. Rechtsklick auf TCP/UDP-Slave und im Kontextmenü [Neu] wählen.
2. Aus der Liste das Anforderungstelegramm "Read Holding Register (03)" wählen.
3. Rechtsklick auf das Anforderungstelegramm "Read Holding Register (03)" und im Kontextmenü [Eigenschaften] wählen.

Als Startadresse des Lesebereichs "0" eintragen.

4. Rechtsklick auf das Anforderungstelegramm "Read Holding Register (03)" und im Kontextmenü [Edit] wählen.
5. In der Objektauswahl die folgenden Variablen auswählen und diese per Drag & Drop in das Register "Eingangsvariablen" ziehen.

Offset	Register Variablen	Typ
0	Slave->Master_WORD_00	WORD
1	Slave->Master_WORD_01	WORD

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Eingangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Prüfen der Modbus-Master-Konfiguration

So prüfen Sie die Modbus-TCP-Master-Konfiguration:

1. Kontextmenü des Modbus-TCP-Masters öffnen und [Verifikation] wählen.
2. Einträge im "Logbuch" sorgfältig überprüfen, gegebenenfalls korrigieren.

Erstellen des Codes für die Steuerungen

So erstellen Sie den Code für die Steuerungen:

1. Code Generator der Master- und der Slave-Ressource starten.
2. Sicherstellen, dass die Codes fehlerfrei generiert wurden.
3. Jeweilige Codes in die Master- und Slave-Steuerungen laden.

6.1.2 Beispiel zur alternativen Register/Bit-Adressierung

In diesem Beispiel wird die Konfiguration aus dem Kapitel "Modbus Beispiel" um 16 Boolesche Variablen im Registerbereich erweitert. Die 16 Booleschen Variablen werden mit dem Anforderungstelegramm Write Multiple Coils (15) ausgelesen.

Konfigurieren der Eingangsvariablen im Modbus-Slave

So konfigurieren Sie die Eingangsvariablen im Modbus-Slave:

1. Im Kontextmenü des Modbus-Slave [Edit] / [Registervariablen] wählen.
2. Wählen Sie in der Objektauswahl die 16 neuen Booleschen Variablen aus und ziehen Sie sie per Drag-and-Drop in den Bereich "Register Eingänge".

Register Adresse	Register Variablen	Typ
0	Master -> Slave_WORD_00	WORD
1	Master -> Slave_WORD_01	WORD
2	Master -> Slave_WORD_03 bis _18	BOOL

3. Öffnen Sie das Kontextmenü durch Rechtsklick auf eine freie Stelle im Bereich "Register Eingänge" und wählen Sie den Menüpunkt "Neue Offsets", um die Offsets der Variablen neu zu nummerieren.

Alternative Register/Bit-Adressierung im Modbus-Slave konfigurieren

So konfigurieren Sie im Modbus-Slave die alternative Register/Bit-Adressierung:

- Im Kontextmenü des Modbus-Slave Edit, Offsets wählen und Alternative Register/Bit-Adressierung verwenden aktivieren.
- Verwenden Sie für dieses Beispiel die folgenden Offsets für die alternativen Bereiche:
 - Register Bereich Offset Bits Input: 1000
 - Register Bereich Offset Bits Output: 1000
 - Bit Bereich Offset Register Input: 8000
 - Bit Bereich Offset Register Output: 8000

HINWEIS



Um mit dem Modbus Anforderungstelegramm "Write Multiple Coils (15)" auf Boolesche Variablen im Bereich "Register Variablen" zuzugreifen, müssen die Variablen in den Bereich "Bitvariablen" gespiegelt sein.

Konfiguration des Anforderungstelegramms zum Schreiben der Ausgangsvariablen

So konfigurieren Sie im Modbus-Master das Anforderungstelegramm zum Schreiben der Ausgangsvariablen (BOOL):

1. Rechtsklick auf TCP/UDP-Slave und im Kontextmenü [Neu] wählen.
2. Aus der Liste das Anforderungstelegramm "Write Multiple Coils (15)" wählen.
3. Rechtsklick auf das Anforderungstelegramm "Write Multiple Coils (15)" und im Kontextmenü [Eigenschaften] wählen.
Startadresse des Schreibbereichs, 8032 eintragen.
4. Rechtsklick auf das Anforderungstelegramm "Write Multiple Coils (15)" und im Kontextmenü [Edit] wählen.
5. In der Objektauswahl die folgenden Variablen auswählen und diese per Drag-and-Drop in das Register "Ausgangsvariablen" ziehen.

Offset	Gespiegelte Register Variable	Typ
0 bis 15	Master->Slave_BOOL_03..._18	BOOL

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

6.1.3 Menüfunktionen des Modbus-Master

Edit

Das Dialogfenster "Edit" des Modbus-Masters enthält die folgende Registerkarte:

Systemvariablen

Die Registerkarte "Systemvariablen" stellt Systemvariablen bereit, die es erlauben, den Zustand des Modbus-Masters im Anwenderprogramm auszuwerten und den Modbus-Master zu steuern.

Element	Beschreibung
Anzahl fehlerhafte Slave-Verbindungen	Anzahl der fehlerhaften Verbindungen mit Modbus-Slaves, die im Zustand aktiviert sind. Deaktivierte Modbus-Slaves werden hier nicht berücksichtigt.
Modbus-Master Aktivierungssteuerung	Hiermit kann der Modbus-Master vom Anwenderprogramm gestoppt oder gestartet werden. 0: Aktivieren 1: Deaktivieren(Flankengetriggert! Modbus-Master kann über PADT auch dann aktiviert werden wenn Modbus-Master Aktivierungssteuerung = 1.)
Modbus-Master Busfehler	Busfehler, z. B. Telegrammfehler (unbekannte Codes etc.), Längenfehler.
Modbus-Master Zustand	Der Modbus-Master Zustand zeigt den momentanen Protokollzustand an: 1: OPERATE 0: OFFLINE
Reset aller Slave-Fehler	Mit einem Wechsel von FALSE->TRUE werden alle Slave-Fehler und Busfehler zurückgesetzt.

Eigenschaften

Die Menüfunktion "Eigenschaften" aus dem Kontextmenü des Modbus-Master öffnet den Dialog Eigenschaften.

Der Dialog enthält die folgenden Register:

Allgemein

Im Register "Allgemein" werden der Name und die Beschreibung für den Modbus-Master eingegeben. Zudem werden hier die Parameter eingestellt, wenn der Modbus-Master zusätzlich als TCP und/oder UDP Gateway arbeiten soll.

Parameter	Beschreibung
Typ	Modbus-Master.
Name	Name für den Modbus-Master.
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</i> in [%] übernehmen. Deaktiviert: Kein Limit des μ P-Budget, für dieses Protokoll verwenden.
Max. μ P-Budget in [%]	Maximale μ P-Last des Moduls, welche bei der Abarbeitung des Protokolls produziert werden darf. Wertebereich: 1 – 100 % Standardwert: 30 %

Parameter	Beschreibung
Verhalten bei CPU/COM Verbindungsverlust	<p>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).</p> <p>Initialdaten annehmen: Eingangsvariablen werden auf die Initialwerte zurückgesetzt.</p> <p>Letzten Wert beibehalten: Eingangsvariablen behalten den letzten Wert.</p>
TCP-Gateway aktivieren	Diese Funktion darf nicht aktiviert werden, weil die RS485-Schnittstelle von der Com UserTask verwendet wird.
TCP-Server-Port	<p>Standard: 502</p> <p>Es können auch andere TCP-Ports konfiguriert werden. Dabei ist die Port-Belegung bei der <i>Internet Corporation for Assigned Names and Numbers (ICANN)</i> zu beachten.</p>
Maximale Anzahl TCP-Verbindungen als Server	<p>Maximale Anzahl gleichzeitig offener TCP-Verbindungen als Server. Wertebereich: 1 – 64</p> <p>Standardwert: 5</p>
UDP-Gateway aktivieren	Diese Funktion darf nicht aktiviert werden, weil die RS485-Schnittstelle von der Com UserTask verwendet wird.
UDP-Port	<p>Standard: 502</p> <p>Es können auch andere TCP-Ports konfiguriert werden. Dabei ist die Port-Belegung bei der <i>Internet Corporation for Assigned Names and Numbers (ICANN)</i> zu beachten.</p>
Maximale Länge der Queue	<p>Länge der Gateway-Warteschlange für noch nicht beantwortete Anforderungstelegramme von anderen Mastern. Dies wird nur beachtet, wenn ein Gateway aktiviert ist.</p> <p>Wertebereich: 1 – 20</p> <p>Standardwert: 3</p>

CPU/COM

Die Standardwerte für die Parameter sorgen für den schnellstmöglichen Datenaustausch der Modbus-Daten zwischen dem COM-Modul und dem CPU-Modul in der Sicherheitssteuerung.

Diese Parameter sollten nur dann geändert werden, wenn eine Reduzierung der COM und/oder CPU-Auslastung für eine Anwendung erforderlich ist und der Prozess dies zulässt.

HINWEIS

Die Änderung der Parameter wird nur dem erfahrenen Programmierer empfohlen. Eine Erhöhung der COM- und CPU-Aktualisierungszeit bedeutet auch, dass die tatsächliche Aktualisierungszeit der Modbus-Daten erhöht wird.

- Die Zeitanforderungen der Anlage sind zu prüfen.

Parameter	Beschreibung
Aktualisierungsintervall der Prozessdaten [ms]	<p>Aktualisierungszeit in Millisekunden, mit der die Daten des Protokolls zwischen COM und CPU ausgetauscht werden. Ist das Aktualisierungsintervall der Prozessdaten Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <p>Wertebereich: $0 - (2^{31} - 1)$</p> <p>Standardwert: 0</p>
Prozessdaten-Konsistenz erzwingen	<p>Aktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU.</p> <p>Deaktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU Zyklen zu je 1100 Byte pro Datenrichtung. Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.</p> <p>Standardwert: Aktiviert</p>

6.1.4 Modbus Funktionscodes des Masters

Mit den Modbus Funktionscodes (Anforderungstelegrammen) haben Sie die Möglichkeit, Variablen in beide Richtungen zu schreiben oder zu lesen. Es können einzelne Variablen oder mehrere aufeinander folgende Variablen gelesen oder geschrieben werden.

So erstellen Sie ein neues Anforderungstelegramm für einen TCP/UDP Slave:

1. Im Strukturbaum [Ressource] / [Protokolle] / [Modbus Master] / [Ethernet-Slaves] einen TCP/UDP Slave wählen.
2. Rechtsklick auf TCP/UDP Slave und im Kontextmenü [Neu] wählen.
3. Aus dem Dialog "Neues Objekt" ein Anforderungstelegramm auswählen.

Modbus Standard Funktionscodes

Folgende Modbus Standard Funktionscodes werden vom Modbus Master unterstützt.

Element	Code	Typ	Bedeutung
READ COILS	01	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Slave.
READ DISCRETE INPUTS	02	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Slave.
READ HOLDING REGISTERS	03	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Slave.
READ INPUT REGISTERS	04	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Slave.
WRITE SINGLE COIL	05	BOOL	Schreiben einer einzelnen Variable (BOOL) in den Slave.
WRITE SINGLE REGISTER	06	WORD	Schreiben einer einzelnen Variable (WORD) in den Slave.

Element	Code	Typ	Bedeutung
WRITE MULTIPLE COILS	15	BOOL	Schreiben mehrerer Variablen (BOOL) in den Slave.
WRITE MULTIPLE REGISTERS	16	WORD	Schreiben mehrerer Variablen beliebigen Typs in den Slave.
READ WRITE HOLDING REGISTERS	23	WORD	Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Slave.

HINWEIS



Weitere Informationen zu Modbus finden Sie in der Spezifikation Modbus Application Protocol Specification www.modbus.org.

SEW-spezifische Funktionscodes

Die SEW-spezifischen Funktionscodes entsprechen den Standard Modbus Funktionscodes. Die zwei Unterschiede sind die maximal zulässige Prozessdatenlänge von 1100 Bytes und das Format von Request und Response-Header.

Element	Code	Typ	Bedeutung
Read Coils Extended	100 (0x64)	BOOL	Entspricht dem Functioncode 01. Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Discrete Inputs Extended	101 (0x65)	BOOL	Entspricht dem Functioncode 02. Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Holding Registers Extended	102 (0x66)	WORD	Entspricht dem Functioncode 03. Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Input Registers Extended	103 (0x67)	WORD	Entspricht dem Functioncode 04. Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Write Multiple Coils Extended	104 (0x68)	BOOL	Entspricht dem Functioncode 15. Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Write Multiple Registers Extended	105 (0x69)	WORD	Entspricht dem Functioncode 16. Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes

Element	Code	Typ	Bedeutung
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Entspricht dem Functioncode 23. Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes (Anforderungstelegramm vom Modbus Master) 1100 Bytes (Antwort an den Master).

6.1.5 Format der Request und Response Header

Die Request und Response Header der SEW-spezifischen Modbus-Funktionscodes sind wie folgt aufgebaut.

Code	Request	Response
100 (0x64)	1 Byte Functionscode 0x64 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x2260)	1 Byte Functionscode 0x64 2 Bytes Anzahl von Bytes= N N Bytes Coil-Daten (8 Coils werden in ein Byte gepackt)
101 (0x65)	1 Byte Functionscode 0x65 2 Bytes Startadresse 2 Bytes Anzahl von Discrete Inputs 1 – 8800(0x2260)	1 Byte Functionscode 0x65 2 Bytes Anzahl von Bytes = N N Bytes Discrete Inputs -Daten (8 Discrete Inputs werden in ein Byte gepackt)
102 (0x66)	1 Byte Functionscode 0x66 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 – 550(0x226)	1 Byte Functionscode 0x66 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
103 (0x67)	1 Bytes Functionscode 0x67 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 – 550(0x226)	1 Byte Functionscode 0x67 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
104 (0x68)	1 Byte Functionscode 0x68 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x2260) 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten	1 Byte Functionscode 0x68 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x2260)

Code	Request	Response
105 (0x69)	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 – 550(0x226) 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 – 550(0x226)
106 (0x6A)	1 Byte Funktionscode 0x6a 2 Bytes Lese-Startadresse 2 Bytes Anzahl von Leseregistern 1 – 550(0x226) 2 Bytes Schreib-Startadresse 2 Bytes Anzahl von Schreibregistern 1 – 550(0x226) 2 Bytes Anzahl von Bytes zum Schreiben=N N Bytes Register-Daten	1 Byte Funktionscode 0x6a 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten

6.1.6 Anforderungstelegramme zum Lesen

Mit den Read-Funktionscodes können Variablen aus dem Slave gelesen werden. Ein Anforderungstelegramm des Modbus-Master enthält neben der Modbus-Funktion die Startadresse des Lese-/Schreibbereichs.

Zum Lesen von Variablen sendet der Modbus-Master ein Anforderungstelegramm zum Lesen an den Modbus-Slave. Der Modbus-Slave sendet daraufhin ein Antworttelegramm mit den angeforderten Variablen an den Modbus-Master zurück.

So konfigurieren Sie ein Anforderungstelegramm zum Lesen:

1. Im Strukturbaum [Anforderungstelegramm] zum Konfigurieren auswählen.
2. Rechtsklick auf Anforderungstelegramm und im Kontextmenü [Edit] wählen.
3. In der Objektauswahl eine Globale Variable wählen, die als Modbus-Empfangsvariablen dienen soll und diese per Drag-and-Drop auf eine leere Stelle im Bereich Eingangssignale ziehen.
4. Diesen Schritt für jede weitere Modbus Empfangsvariable wiederholen.
5. Kontextmenü durch einen Rechtsklick auf eine leere Stelle im Bereich "Eingangssignale" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Die folgenden Anforderungstelegramme zum Lesen stehen zur Verfügung:

Read Coils (01) und Extended (100)

Lesen mehrerer Variablen (BOOL) aus dem Slave.

Element	Bedeutung
Typ	Modbus-Funktion Read Coils.
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.

Element	Bedeutung
Startadresse des Lesebereichs	0 – 65535.

Read Discrete Inputs (02) und Extended (101)

Lesen mehrerer Variablen (BOOL) aus dem Slave.

Element	Bedeutung
Typ	Modbus-Funktion Read Discrete Inputs.
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Lesebereichs	0 – 65535.

Read Holding Registers (03) und Extended (102)

Lesen mehrerer Variablen beliebigen Typs aus dem Slave.

Element	Bedeutung
Typ	Modbus-Funktion Read Holding Registers.
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Lesebereichs	0 – 65535.

Read Input Registers (04) und Extended (103)

Lesen mehrerer Variablen beliebigen Typs aus dem Slave

Element	Bedeutung
Typ	Modbus-Funktion Read Input Registers.
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Lesebereichs	0 – 65535.

6.1.7 Anforderungstelegramm zum Lesen und Schreiben

Zum Lesen und Schreiben von Variablen sendet der Modbus-Master ein Anforderungstelegramm zum Lesen und Schreiben an den Modbus-Slave.

Zuerst schreibt der Modbus-Master die definierten Schreibvariablen in den definierten Import-Bereich des Modbus-Slave.

Anschließend liest der Modbus-Master die definierten Lesevariablen aus dem definierten Export-Bereich des Modbus-Slave.



HINWEIS

Die Funktionen Schreiben und Lesen sind auch bei dem Anforderungstelegramm zum Lesen und Schreiben voneinander unabhängig, sie werden nur in einem gemeinsamen Anforderungstelegramm gesendet.

Eine häufige Anwendung für das Anforderungstelegramm zum Lesen und Schreiben ist jedoch, dass die geschriebenen Variablen des Modbus-Master wieder zurückgelesen werden. Damit wird überprüft, ob die gesendeten Variablen korrekt geschrieben wurden.

So konfigurieren Sie ein Anforderungstelegramm zum Lesen und Schreiben:

1. Im Strukturbaum [Anforderungstelegramm] zum konfigurieren auswählen.
2. Rechtsklick auf Anforderungstelegramm und im Kontextmenü [Edit] wählen.

Lesevariablen konfigurieren

So konfigurieren Sie die Variablen zum Lesen:

1. Wählen Sie in der Objektauswahl eine Globale Variable, die Sie mit der neuen Modbus Empfangsvariablen verbinden wollen und ziehen Sie diese per Drag-and-Drop in die Spalte "Globale Variable" der Modbus Empfangsvariablen.
2. Schritt 1 für jede weitere Modbus Empfangsvariable wiederholen.
3. Kontextmenü durch einen Rechtsklick auf eine leere Stelle im Bereich "Eingangssignale" öffnen und [Neue Offsets wählen], um die Offsets der Variablen neu zu nummerieren.

Schreibvariablen konfigurieren

So konfigurieren Sie die Variablen zum Schreiben:

1. Wählen Sie in der Objektauswahl eine Globale Variable, die Sie mit der neuen Modbus Sendevariablen verbinden wollen und ziehen Sie diese per Drag-and-Drop in die Spalte Globale Variable der Modbus Sendevariablen.
2. Schritt 1 für jede weitere Modbus SendevARIABLE wiederholen.
3. Kontextmenü durch einen Rechtsklick auf eine leere Stelle im Bereich "Ausgangssignale" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Read Write Holding Register (23) und Extended (106)

Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus-Funktion <i>Read Write Holding Registers</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Lesebereichs	0 – 65535.
Startadresse des Schreibbereichs	0 – 65535.

6.1.8 Anforderungstelegramm zum Schreiben

Mit den Write-Funktionscodes werden Variablen nur in den Importbereich eines Slaves geschrieben.

Ein Anforderungstelegramm des Modbus-Master enthält neben der Modbus-Funktion die Startadresse des Lese-/Schreibbereichs.

Zum Schreiben von Variablen sendet der Modbus-Master ein Anforderungstelegramm zum Schreiben an den Modbus-Slave. Der Modbus-Slave schreibt die empfangenen Variablen in seinen Import-Bereich.

Im Dialog „Variablen zuweisen“ eines Anforderungstelegramms zum Schreiben müssen die Variablen eingefügt werden, die der Modbus-Master zum Modbus-Slave schreibt.

So konfigurieren Sie ein Anforderungstelegramm zum Schreiben:

1. Im Strukturbaum [Anforderungstelegramm] zum Konfigurieren auswählen.
2. Rechtsklick auf Anforderungstelegramm und im Kontextmenü [Edit] wählen.
3. In der Objektauswahl eine Globale Variable wählen, die als Modbus-SendevARIABLE dienen soll und diese per Drag-and-Drop auf eine leere Stelle im Bereich "Sendesignale" ziehen.
4. Schritt 3 für jede weitere Modbus-SendevARIABLE wiederholen.
5. Kontextmenü durch einen Rechtsklick auf eine leere Stelle im Bereich "Sendesignale" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Die folgenden Anforderungstelegramme zum Schreiben stehen zur Verfügung:

Write Multiple Coils (15) und Extended (104)

Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Multiple Coils</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Schreibbereichs	0 – 65535.

Write Multiple Registers (16) und Extended (105)

Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Multiple Registers</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Schreibbereichs	0 – 65535.

Write Single Coil (05)

Schreiben einer einzelnen Variablen (BOOL) in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Single Coil</i> .

Element	Bedeutung
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Schreibbereichs	0 – 65535.

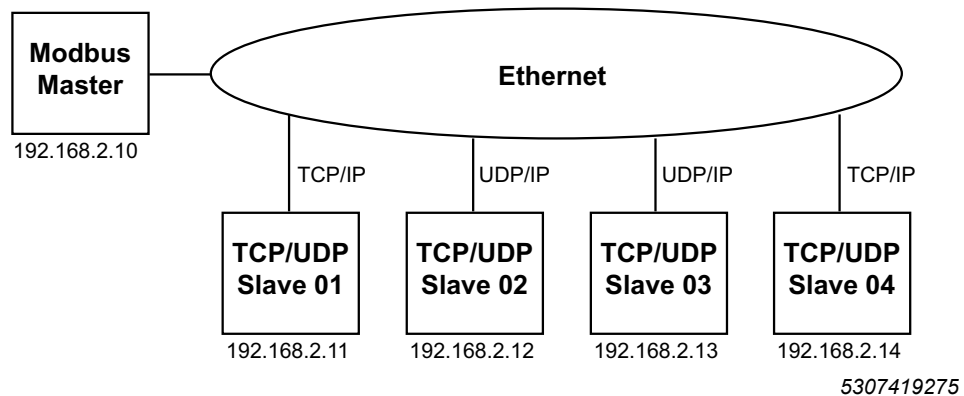
Write Single Register (06)

Schreiben einer einzelnen Variablen (WORD) in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Single Register</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Schreibbereichs	0 – 65535.

6.1.9 Ethernet-Slaves (TCP/UDP-Slaves)

Der Modbus-Master kann mit bis zu 64 TCP/IP und 247 UDP/IP-Slaves kommunizieren.



So erstellen Sie im Modbus-Master eine neue Verbindung zu einem TCP/UDP Slave:

1. Im Strukturbaum [Ressource] / [Protokolle] / [Modbus Master] / [Ethernet-Slaves] öffnen.
2. Rechtsklick auf Ethernet-Slaves und im Kontextmenü [Neu] wählen.
3. Aus der Liste "TCP/UDP-Slaves" wählen und mit [OK] bestätigen.
4. Konfiguration des TCP/UDP-Slave im Modbus-Master:

[Edit] zum Zuweisen der Systemvariablen, siehe Kapitel "Systemvariablen der TCP/UDP-Slaves"

[Eigenschaften] wählen zum Konfigurieren der Eigenschaften, siehe Kapitel "Eigenschaften TCP/UDP-Slaves".



HINWEIS

Befinden sich die TCP/UDP-Slaves und der Modbus-Master in verschiedenen Subnetzen, müssen in der Routing-Tabelle die entsprechenden benutzerdefinierten Routen eingetragen werden.

Der Modbus-TCP-Master sendet mit seinen Telegrammen an den Modbus- TCP-Slave zusätzlich zur IP-Adresse immer eine Modbus-Slave-Adresse (Unit Identifier) mit. Diese Adresse ist immer FF_{Hex} (255).

Systemvariablen der TCP/UDP-Slaves

Das Register Systemvariablen stellt Systemvariablen bereit, die es erlauben, den Zustand des TCP/UDP-Slave im Anwenderprogramm auszuwerten und zu steuern.

Der Status des TCP/UDP-Slave kann im Anwenderprogramm mit den folgenden Statusvariablen ausgewertet werden:

Element	Beschreibung	
Modbus-Slave Aktivierungssteuerung	Hiermit kann der TCP/UDP- Slave vom Anwenderprogramm deaktiviert oder aktiviert werden.	0: Aktivieren 1: Deaktivieren (Flankengetriggert! Modbus-Slave kann über PADT auch dann aktiviert werden wenn Modbus-Slave Aktivierungssteuerung = 1.)

Element	Beschreibung	
Modbus-Slave Fehler	Fehlercode	Die Fehlercodes 0x01 – 0x0b entsprechen den Exception Codes der Modbus-Protokollspezifikation. 0x00: Kein Fehler
	Exception Codes:	0x01: Ungültiger Funktionscode 0x02: Ungültige Adressierung 0x03: Ungültige Daten 0x04: (nicht verwendet) 0x05: (nicht verwendet) 0x06: Device Busy (nur Gateway, nicht unterstützt) 0x08: (nicht verwendet) 0x0a: (nicht verwendet) 0x0b: No Response from Slave (nur Gateway, nicht unterstützt)
	SEW-spezifische Codes	0x10: Defekter Frame empfangen 0x11: Frame mit falscher Transaktions ID empfangen 0x12: Unerwartete Antwort empfangen 0x13: Antwort über falsche Verbindung erhalten 0x14: Falsche Antwort auf einen Schreibauftrag 0xff: Slave Timeout
Modbus-Slave Zustand	Verbindungsstatus des TCP/UDP-Slave	0: Deaktiviert 1: Nicht verbunden 2: Verbunden

Eigenschaften TCP/UDP-Slaves

Zur Konfiguration der Verbindung zum TCP/UDP-Slave müssen im Modbus-Master die folgenden Parameter eingestellt werden:

Parameter	Beschreibung
Typ	TCP/UDP-Slave.
Name	Beliebiger eindeutiger Name für den TCP/UDP-Slave.
Beschreibung	Beliebige eindeutige Beschreibung für den TCP/UDP-Slave.

Parameter	Beschreibung
Master-Slave Datenaustausch [ms]	Intervall für den Datenaustausch mit diesem Slave 1 bis ($2^{31}-1$). Konnte der Slave nach <i>Maximale Anzahl Sendewiederholungen</i> nicht erreicht werden, wird das Intervall <i>Master-Slave Datenaustausch</i> um das Vierfache hochgesetzt.
TCP-Verbindung nur bei Bedarf	Wenn das Transportprotokoll TCP ist, wird hier eingestellt, ob die Verbindung zu diesem Slave nach jedem Datenaustausch automatisch abgebaut werden soll. TRUE: Die Verbindung abbauen. FALSE: Die Verbindung nicht abbauen. Standardwert: FALSE
Receive Timeout [ms]	Receive Timeout für diesen Slave [ms]. Nach dieser Zeit wird ein neuer Sendeversuch gestartet.
IP-Adresse	IP-Adresse des TCP/UDP-Slave.
Port	Standard: 502 Es können auch andere TCP/UDP-Ports konfiguriert werden. Dabei ist die Port-Belegung bei der <i>Internet Corporation for Assigned Names and Numbers (ICANN)</i> zu beachten.
Kommunikationsart IP-Protokoll	TCP oder UDP. Standardwert: TCP
Maximale Anzahl Sendewiederholungen	Maximale Anzahl an Sendewiederholungen, falls Slave nicht antwortet. Die Anzahl der Sendewiederholungen kann beliebig eingestellt werden (0 – 65535). Bei TCP/IP immer null, nicht änderbar. Empfohlen wird eine Anzahl von 0 bis 8 Sendewiederholungen.

6.1.10 Control-Panel (Modbus-Master)

Im Control-Panel kann der Anwender die Einstellungen des Modbus-Master überprüfen und steuern. Zudem werden aktuelle Statusinformationen (z. B. Master-Zustand usw.) des Masters angezeigt.

So öffnen Sie das Control-Panel zur Überwachung des Modbus-Master:

1. Im Strukturbaum [Hardware] und im Kontextmenü [Online] wählen.
2. Im System-Login, Zugangsdaten eingeben um die Online-Ansicht der Hardware zu öffnen.
3. Doppelklick auf "COM-Modul" und im Strukturbaum [Modbus Master] wählen.

Kontextmenü (Modbus-Master)

Aus dem Kontextmenü des selektierte Modbus-Master können die folgenden Kommandos gewählt werden:

Offline: Mit diesem Kommando wird der Modbus-Master gestoppt.

Operate: Mit diesem Kommando wird der Modbus-Master gestartet.

Statistik zurücksetzen: Setzt die statistischen Daten (z. B. Anzahl Busfehler, Zykluszeit min, max usw.) auf null zurück.

Anzeigefeld (Modbus-Master)

In dem Anzeigefeld werden die folgenden Werte des selektierten Modbus-Masters angezeigt.

Element	Beschreibung
Name	Name des Modbus-Masters.
Master-Zustand	Der Modbus-Masterzustand zeigt den momentanen Protokollzustand an: OPERATE OFFLINE
Anzahl Busfehler	Zähler Anzahl der Busfehler.
Gestörte Verbindungen	Zähler Anzahl der gestörten Verbindungen.
µP-Last (projektierte)	Siehe Eigenschaften im Kapitel "Menüfunktionen des Modbus-Master".
µP-Last (tatsächliche)	

6.1.11 Control-Panel (Modbus-Master → Slave)

Im Control-Panel kann der Anwender die Einstellungen der Kommunikationspartner des Modbus-Masters überprüfen und aktivieren oder deaktivieren.

Zudem werden aktuelle Statusinformationen (z. B. Slave-Zustand usw.) des Kommunikationspartners angezeigt.

So öffnen Sie das Control-Panel zur Überwachung der Modbus-Verbindung:

- Im Strukturbaum [Hardware] und im Kontextmenü [Online] wählen.
- Im System-Login Zugangsdaten eingeben um die Online-Ansicht der Hardware zu öffnen.
- Doppelklick auf "COM-Modul" und im Strukturbaum [Modbus Master] / [Slave] wählen.

6.2 Modbus-Slave

Der Modbus-Slave kann gleichzeitig über Ethernet (TCP/UDP) mehrere Modbus-Master bedienen.

Eigenschaft	Beschreibung
Modbus Slave	Es kann ein Modbus-Slave konfiguriert werden.
Anzahl Masterzugriffe	<ul style="list-style-type: none"> • TCP Maximal 20 Modbus-Master können auf den Slave zugreifen. • UDP Unbegrenzte Anzahl von Modbus-Master können auf den Slave zugreifen.

Eigenschaft	Beschreibung				
Max. Größe der Sendedaten	64 kB senden 64 kB empfangen				
Max. Größe der Empfangsdaten	Hinweis: Die Statusbytes des Masters und die Statusbytes von jedem zugeordneten Slave müssen von der maximalen Größe der Sendedaten subtrahiert werden.				
Darstellungsformat der Modbus-Daten	Die Sicherheitssteuerung verwendet das Big-Endian-Format. Beispiel: 32 Bit Daten (z. B. DWORD, DINT):				
	32 Bit Daten (hex)		0x12345678		
	Speicher-Offset	0	1	2	3
	Big Endian	12	34	56	78
	Middle Endian	56	78	12	34
	Little Endian	78	56	34	12

6.2.1 Konfiguration des Modbus-TCP-Slave

So legen Sie einen neuen Modbus-Slave an:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] öffnen.
2. Im Kontextmenü von Protokolle [Neu] / [Modbus-Slave-Set] wählen, um einen neuen Modbus-Slave-Set hinzuzufügen.
3. Im Kontextmenü des Modbus-Slave-Set [Edit] wählen und [Eigenschaften Modbus-Slave-Set öffnen], Standardwerte beibehalten.

Register Modbus-Slave wählen und folgende Einstellungen vornehmen:

- COM Modul wählen.
- "TCP aktivieren" aktivieren.
- Die restlichen Parameter behalten die Standardwerte.

HINWEIS



Ein Konfigurationsbeispiel zur Verbindung eines Modbus-TCP-Slave mit einem Modbus-TCP-Master ist im Kapitel "Modbus-Beispiel" beschrieben.

6.2.2 Menüfunktionen des Modbus-Slave-Set

Der Menüpunkt [Edit] aus dem Kontextmenü des Modbus-Slave-Set öffnet den Dialog "Eigenschaften Modbus-Slave-Set". Das Dialogfenster enthält die folgenden Register.

Eigenschaften Modbus-Slave-Set

Im Register "Eigenschaften Modbus-Slave-Set" werden die folgenden Parameter für den Modbus Slave eingestellt.

Element	Beschreibung
Name	Name des Modbus-Slave-Set.

Element	Beschreibung
Maximales μ P-Budget aktivieren	<p>Aktiviert: Limit des μP-Budget aus dem Feld <i>Max. μP-Budget in [%]</i> übernehmen.</p> <p>Deaktiviert: Kein Limit des μP-Budget für dieses Protokoll verwenden.</p>
Maximales μ P-Budget in [%]	<p>Maximale μP-Last des COM-Moduls, die bei der Abarbeitung des Protokolls produziert werden darf.</p> <p>Wertebereich: 1 – 100 %</p> <p>Standardwert: 30 %</p>
Redundanzbetrieb aktivieren	<p>Aktiviert: Redundanzbetrieb (nicht unterstützt)</p> <p>Deaktiviert: Monobetrieb</p> <p>Standardwert: Deaktiviert</p>
Maximale Antwortzeit [ms]	<p>Zeitspanne nach dem Empfang einer Anfrage, innerhalb der der Modbus-Slave noch antworten darf.</p> <p>Dieser Wert muss bei der Sicherheitssteuerung MOVISAFE® HM31 auf 0 ms gesetzt werden.</p> <p>Wertebereich: 0 – $(2^{31}-1)$ ms</p> <p>Standardwert: 5000 ms (0 = keine Limitierung)</p>
Bereich zum Lesen der Funktionscodes 1, 3, 100, 102	<p>Der Parameter bestimmt, von welchem Datenbereich die Daten für den Funktionscode 1, 3, 100, 102 gelesen werden sollen.</p> <p>Wertebereich:</p> <ul style="list-style-type: none"> • Importbereich • Exportbereich
Bereich zum Lesen bei Funktionscode 23, 106	<p>Hier kann der Anwender den Bereich des Modbus-Slave festlegen, aus dem der Funktionscode 23 lesen soll.</p> <p>Importbereich: Der Master greift auf den Importbereich des Slaves lesend und schreibend zu.</p> <p>Exportbereich: Der Master liest vom Exportbereich des Slaves und schreibt auf den Importbereich des Slaves.</p> <p>Hinweis: Das Lesen und Schreiben erfolgt innerhalb eines CPU-Zyklus. Das heißt, die gelesenen Daten wurden vom letzten CPU-Zyklus bereitgestellt.</p>
COM: Werte bei Verbindungsverlust zu Master	<p>Bei Verbindungsverlust des Kommunikationsmoduls zum Modbus-Master werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert an das Prozessormodul weitergeleitet.</p> <p>Initialdaten annehmen: Eingangsvariablen werden auf die Initialwerte zurückgesetzt.</p> <p>Letzten Wert beibehalten: Eingangsvariablen behalten den letzten Wert.</p>

Element	Beschreibung
CPU: Werte bei Verbindungsverlust zu COM	<p>Bei Verbindungsverlust des Prozessormoduls zum Kommunikationsmodul werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert im Prozessormodul verwendet.</p> <p>Gleiches Verhalten wie COM zu Master:</p> <p>Siehe Einstellungen in Parameter "COM: Werte bei Verbindungsverlust zu Master".</p> <p>Letzten Wert beibehalten:</p> <p>Eingangsvariablen behalten den letzten Wert.</p> <p>Standardwert:</p> <p>Eingangsvariablen behalten den letzten Wert.</p>
Alternative Register/Bit Adressierung verwenden	<p>Aktiviert: Alternative Adressierung verwenden</p> <p>Deaktiviert: Alternative Adressierung nicht verwenden</p> <p>Standardwert: Deaktiviert, siehe Kapitel "Offsets für alternative Modbus-Adressierung"</p>
Register Bereich Offset Bits Input	<p>Wertebereich: 0 – 65535</p> <p>Standardwert: 0</p>
Register Bereich Offset Bits Output	
Bit Bereich Offset Register Input	
Bit Bereich Offset Register Output	
Aktualisierungsintervall der Prozessdaten [ms]	<p>Aktualisierungszeit in Millisekunden, mit der die Daten des Protokolls zwischen COM und CPU ausgetauscht werden. Ist die <i>Refresh Rate</i> Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <p>Wertebereich: 0 – ($2^{31}-1$)</p> <p>Standardwert: 0</p>
Prozessdaten-Konsistenz erzwingen	<p>Aktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU.</p> <p>Deaktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU-Zyklen zu je 1100 Byte pro Datenrichtung. Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.</p> <p>Standardwert: Aktiviert</p>

Registervariable (Register Zugriff)

Im Register "Registervariablen" sind die Variablen einzutragen, die der Master 16-bit-weise adressiert (Funktionscode 3, 4, 6, 16, 23, 102, 103, 105, 106).

Bitvariablen (Bit-/Coil-Zugriff)

Im Register "Bitvariablen" sind die Variablen einzutragen, die der Master 1-bitweise adressiert (Funktionscode 1, 2, 5, 15, 100, 101, 104).

6.2.3 Sende-/Empfangsvariablen zuweisen

Im Register "Eingänge" werden alle Variablen zugewiesen, die der Modbus-Slave vom Modbus-Master empfängt..

So konfigurieren Sie die Sendevariablen des Modbus-Slave:

1. Im Strukturbaum den Modbus-Slave wählen, den Sie konfigurieren wollen.
2. Rechtsklicken auf Modbus-Slave und [Edit] wählen.
3. Register "Registervariablen" oder "Bitvariablen" wählen.
4. In der Objektauswahl eine Variable wählen und diese per Drag-and-Drop in den Bereich "Register Ausgänge" ziehen.
5. Schritt 4 für jede weitere Variable die Sie als Sendevariable für den Modbus-Slave definieren wollen wiederholen.
6. Rechtsklick in den Bereich "Register Ausgänge" und [Neue Offsets] wählen.

Empfangsvariablen des Modbus-Slave konfigurieren

So konfigurieren Sie die Empfangsvariablen des Modbus-Slave:

1. Im Strukturbaum den Modbus-Slave wählen, den Sie konfigurieren wollen.
2. Klicken Sie mit der rechten Maustaste auf Modbus-Slave und wählen Sie [Edit].
3. Register "Registervariablen" oder "Bitvariablen" wählen.
4. In der Objektauswahl eine Variable wählen und diese per Drag-and-Drop in den Bereich "Register Eingänge" ziehen.
5. Schritt 4 für jede weitere Variable wiederholen die Sie als Empfangsvariable für den Modbus-Slave definieren wollen.
6. Rechtsklick in den Bereich "Register Eingänge" und [Neue Offsets] wählen.

6.2.4 Systemvariablen Modbus-Slave-Set

Die Registerkarte "Systemvariablen Modbus-Slave-Set" stellt die folgende Systemvariable bereit.

Element	Beschreibung
Redundanz-Zustand (nicht unterstützt)	<p>Dieser Parameter beschreibt den Redundanz-Zustand des redundanten Modbus-Slave Kommunikationsmodul-Paars.</p> <p>0: Redundante Modbus-Slave COM-Module aktiv 1: Erstes Modbus-Slave COM-Modul nicht aktiv 2: Redundantes Modbus-Slave COM-Modul nicht aktiv 3: Beide Modbus-Slave COM-Module nicht aktiv</p>

Innerhalb der Registerkarte "Modbus-Slave" befinden sich die beiden Registerkarten "Eigenschaften" und "Systemvariablen".

Element	Beschreibung
Modul	Auswahl des COM-Moduls, auf dem dieses Protokoll abgearbeitet wird.
Master Überwachungszeit [ms]	Zeitspanne nach dem Empfang einer Anfrage, innerhalb der der Modbus-Slave reagieren muss. Bei Verbindungsverlust des Kommunikationsmoduls zum Modbus-Master werden in Abhängigkeit des Parameters <i>COM: Werte bei Verbindungsverlust zu Master</i> die Eingangsvariablen entweder initialisiert oder unverändert an das Prozessormodul weitergeleitet siehe Kapitel "Eigenschaften Modbus-Slave-Set" Wertebereich $1 - (2^{31}-1)$ [ms] Standardwert: 0 ms (keine Limitierung)
TCP aktivieren	Aktiviert: TCP/IP-Verbindung aktiviert Deaktiviert: TCP/IP-Verbindung deaktiviert Standardwert: Deaktiviert
TCP-Port	Standardwert: 502
Maximale Anzahl TCP-Verbindungen	Maximale Anzahl gleichzeitig offener TCP-Verbindungen als Server. Wertebereich: 1 – 20. Standardwert: 3
UDP aktivieren	Aktiviert: UDP/IP-Verbindung aktiviert Deaktiviert: UDP/IP-Verbindung deaktiviert Standardwert: Deaktiviert
UDP Port	Standardwert: 502.

Das Register "Systemvariablen" stellt Systemvariablen bereit, die es erlauben, den Zustand des Modbus-Slave im Anwenderprogramm auszuwerten und den Modbus-Slave zu steuern.

Element	Beschreibung
Durchschnittlicher Pufferfüllstand für Anfragen	Durchschnittliche Anzahl gleichzeitiger Master-Anfragen.
Gültige Master-Anfragen	Anzahl der gültigen Master-Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Master-Anfragen	Gesamtanzahl aller Master-Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Master-Überwachungszeit [ms]	Zeitspanne nach dem Empfang einer Anfrage, innerhalb der der Modbus-Slave reagieren muss, siehe Kapitel 7.3.7.
Master-Verbindungszustand	FALSE: nicht verbunden TRUE: Verbunden
Maximaler Pufferfüllstand für Anfragen	Maximale Anzahl gleichzeitiger Master-Anfragen.

Element	Beschreibung
Rücksetzen aller Zähler	Mit dieser Systemvariable können über das Anwenderprogramm alle Zähler zurückgesetzt werden. Ein Wechsel von 0 auf 1 löst die Reset-Funktion aus, Werte > 1 werden als 1 behandelt.
Ungültige Master-Anfragen	Anzahl ungültiger Master-Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten. Ungültige Anfragen sind solche, die der Modbus-Slave mit einem Fehlercode an den Modbus-Master beantwortet. Fehlerhafte Sendungen, die bereits auf Treiberebene erkannt und ausgefiltert werden (Framing Errors, CRC-Fehler, Längenfehler) sind hier nicht mit enthalten, sondern werden nur über die Diagnose gemeldet.
Verworfenne Anfragen	Anzahl der verworfenen Master-Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Zeitüberschreitungen bei Antworten	Anzahl der <i>Zeitüberschreitungen bei Antworten</i> seit dem letzten Rücksetzen aller Zähler oder Einschalten. Die <i>Zeitüberschreitungen bei Antworten</i> ist die maximale Zeit, die bis zur Empfangsbestätigung einer Nachricht beim Absender vergehen darf.

6.2.5 Modbus-Funktionscodes des Modbus-Slaves

Folgende Modbus-Funktionscodes werden vom Modbus-Slave unterstützt.

Element	Code	Typ	Bedeutung
READ COILS	01	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Import- oder Exportbereich des Slaves. Maximale Länge der Prozessdaten: 251 Bytes ¹⁾
READ DISCRETE INPUT	02	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 251 Bytes
READ HOLDING REGISTER	03	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Import- oder Exportbereich des Slaves. Maximale Länge der Prozessdaten: 250 Bytes ¹⁾
READ INPUT REGISTER	04	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 250 Bytes
WRITE SINGLE COIL	05	BOOL	Schreiben eines einzelnen Signals (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1 Byte
WRITE SINGLE REGISTER	06	WORD	Schreiben eines einzelnen Signals (WORD) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 2 Bytes

Element	Code	Typ	Bedeutung
WRITE MULTIPLE COILS	15	BOOL	Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 247 Bytes
WRITE MULTIPLE REGISTER	16	WORD	Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 246 Bytes
READ WRITE MULTIPLE REGISTER	23	WORD	Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 242 Bytes (Anforderungstelegramm vom Modbus-Master) 250 Bytes (Antwort an den Master).
Read Device Identification	43	Any	Liefern die Identifikationsdaten des Slaves an den Master. Details entnehmen Sie dem Hinweis zur Modbus-Funktion: Read Device Identification (43)

1) Export Bereich kann nur bei MOVISAFE® HM31 als Slave gewählt werden.

Die Funktionscodes 03, 04, 16 und 23 unterstützen neben dem Datentyp WORD (2 Byte) auch beliebige andere Datentypen.

Für jeden Request muss die Startadresse der ersten zu übertragenden Variable und die Anzahl der Register/Bits der zu übertragenden Variablen eingetragen werden.

Fehlercodes:

- Sendet der Master ein Telegramm mit unbekanntem Funktionscode, so antwortet die Steuerung mit Fehlercode 1 (Invalid Code).
- Wenn die Länge des Anfragetelegramms nicht zur Variablengrenze passt, so antwortet der Slave mit Fehlercode 2 (Invalid Data).
- Sendet der Master ein Telegramm mit fehlerhaften Werten (z. B. Längenfeld), so antwortet der Slave mit Fehlercode 3 (Invalid Value).

Die Kommunikation erfolgt nur im Zustand RUN des COM-Moduls. Master-Anfragen in allen anderen Betriebszuständen des COM-Moduls werden nicht beantwortet.

Hinweis zur Modbus-Funktion: Read Device Identification (43)

Der Modbus-Slave liefert die Identifikationsdaten an den Master und unterstützt die folgenden Object-Ids:

Basic:

```
0x00 VendorName "SEW-EURODRIVE GmbH + Co KG"
0x01 ProductCode "<Modul Seriennummer>"
0x02 MajorMinorRevision "<COM Vx.y CRC>"
```

Regular:

```
0x03 VendorUrl "http://www.sew-eurodrive.de"
0x04 ProductName "SEW F-PLC"
0x05 ModelName "PFF-HM31A1"
0x06 UserApplicationName "-----[S.R.S]"
```

Extended:

```

0x80 leer "-----"
0x81 leer "-----"
0x82 leer "-----"
0x83 leer "-----"
0x84 leer "-----"
0x85 leer "-----"
0x86 CRC der Datei modbus.config "<0x234adcef>"

```

(Konfigurationsdatei des Modbus-Slave-Protokolls im Dateisystem der CPU. Zu vergleichen mit den Angaben in SILworX unter Online/Versionsvergleich).

Es werden folgende ReadDevice ID Codes unterstützt:

- (1) Read Basic device identification (stream access)
- (2) Read regular device identification (stream access)
- (3) Read extended device identification (stream access)
- (4) Read one specific identification object (inidividual access)

Weitere Informationen zu Modbus finden Sie in der Spezifikation Modbus „Application-Protocol Specification“ www.modbus.org

6.2.6 SEW-spezifische Funktionscodes

Die SEW-spezifischen Funktionscodes entsprechen den Standard Modbus-Funktionscodes. Die einzigen Unterschiede sind die maximal zulässige Prozessdatenlänge von 1100 Bytes und das Format von Request und Responce-Header:

Element	Code	Typ	Bedeutung
Read Coils Extended	100 (0x64)	BOOL	Entspricht dem Functioncode 01 Lesen mehrerer Variablen (BOOL) aus dem Import- oder Exportbereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes ¹⁾)
Read Discrete Inputs Extended	101 (0x65)	BOOL	Entspricht dem Functioncode 02 Lesen mehrerer Variablen (BOOL) aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Holding Registers Extended	102 (0x66)	WORD	Entspricht dem Functioncode 03 Lesen mehrerer Variablen beliebigen Typs aus dem Import- oder Export)-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Input Registers Extended	103 (0x67)	WORD	Entspricht dem Functioncode 04 Lesen mehrerer Variablen beliebigen Typs aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Write Multiple Coils Extended	104 (0x68)	BOOL	Entspricht dem Functioncode 15 Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes

Element	Code	Typ	Bedeutung
Write Multiple Registers Extended	105 (0x69)	WORD	Entspricht dem Functioncode 16 Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Entspricht dem Functioncode 23 Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes (Anforderungstelegramm vom Modbus-Master) 1100 Bytes (Antwort an den Master).

1) Export-Bereich kann nur bei PFF-HM31A als Slave gewählt werden

Format der Request und Response Header

Die Request und Response Header der SEW-spezifischen Modbus-Funktionscodes sind wie folgt aufgebaut.

Code	Request	Response
100 (0x64)	1 Byte Functionscode 0x64 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x2260)	1 Byte Functionscode 0x64 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten (8 Coils werden in ein Byte gepackt)
101 (0x65)	1 Byte Functionscode 0x65 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x226)	1 Byte Functionscode 0x65 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten (8 Coils werden in ein Byte gepackt)
102 (0x66)	1 Byte Functionscode 0x66 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 – 550(0x226)	1 Byte Functionscode 0x66 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
103 (0x67)	1 Bytes Functionscode 0x67 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 – 550(0x226)	1 Byte Functionscode 0x67 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
104 (0x68)	1 Byte Functionscode 0x68 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x2260) 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten	1 Byte Functionscode 0x66 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 – 8800(0x2260)

Code	Request	Response
105 (0x69)	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 – 550(0x226) 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 – 550(0x226)
106 (0x6A)	1 Byte Funktionscode 0x6a 2 Bytes Lese-Startadresse 2 Bytes Anzahl von Leseregistern 1 – 550(0x226) 2 Bytes Schreib-Startadresse 2 Bytes Anzahl von Schreibregistern 1 – 550(0x226) 2 Bytes Anzahl von Bytes zum Schreiben=N N bytes Register-Daten	1 Byte Funktionscode 0x6a 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten

6.2.7 Modbus-Adressierung durch Bit und Register

Dieser Adressierungsmodus entspricht dem Standard der Modbus-Adressierung und kennt nur die zwei Datenlängen Bit (1 Bit) und Register (16 Bit) mit denen alle zugelassenen Datentypen übertragen werden können.

Im Modbus-Slave gibt es einen "Register-Bereich" (Ein- und Ausgänge) und einen "Bit-Bereich" (Ein- und Ausgänge). Beide Bereiche sind voneinander getrennt und können alle zugelassenen Datentypen aufnehmen.

Der Unterschied dieser Bereiche besteht in den erlaubten Modbus-Funktionscodes, mit denen auf diese Bereiche zugegriffen werden kann.

HINWEIS



Die Modbus-Adressierung durch Bit und Register garantiert keine Variablen-Integrität, d. h. mit diesem Zugriff können beliebige Teile von Variablen gelesen / geschrieben werden. Variablen vom Typ BOOL werden gepackt abgelegt, d. h. jede Variable vom Typ BOOL ist als Bit innerhalb eines Bytes abgelegt.

Register-Bereich

Innerhalb der Registerkarte "Registervariablen" werden die Variablen im Register-Bereich angelegt. Siehe auch Kapitel "Sende/Empfangsvariablen zuweisen".

HINWEIS



Um mit den Modbus-Funktionscodes 1, 2, 5, 15 auf die Variablen im Register-Bereich zuzugreifen, müssen die Variablen in den Bit-Bereich gespiegelt werden, siehe Kapitel "Zugriff auf die Register Variablen im Bit-Bereich des Modbus-Slaves".

Auf die Variablen im Register-Bereich kann nur über die Modbus-Funktionscodes 3, 4, 6, 16, 23 zugegriffen werden. Dazu muss in den Eigenschaften des Funktionscodes die Startadresse der ersten Variable eingetragen werden.

Beispiel: Zugriff auf Variablen im Register-Bereich des Modbus-Slave

Register-Variablen	Register.Bit	Bit
00_Register_Bereich_WORD	0.0	0
01_Register_Bereich_SINT	1.8	16
02_Register_Bereich_SINT	1.0	24
03_Register_Bereich_REAL	2.0	32
04_Register_Bereich_BOOL	4.8	64
05_Register_Bereich_BOOL	4.9	65
06_Register_Bereich_BOOL	4.10	66
07_Register_Bereich_BOOL	4.11	67
08_Register_Bereich_BOOL	4.12	68
09_Register_Bereich_BOOL	4.13	69
10_Register_Bereich_BOOL	4.14	70
11_Register_Bereich_BOOL	4.15	71
12_Register_Bereich_BOOL	4.0	72
13_Register_Bereich_BOOL	4.1	73
14_Register_Bereich_BOOL	4.2	74
15_Register_Bereich_BOOL	4.3	75
16_Register_Bereich_BOOL	4.4	76
17_Register_Bereich_BOOL	4.5	77
18_Register_Bereich_BOOL	4.6	78
19_Register_Bereich_BOOL	4.7	79

Modbus-Master-Konfiguration des Anforderungstelegramms

So lesen Sie im Modbus-Master die Variablen *01_Register_Bereich_SINT* bis *03_Register_Bereich_REAL* ein:

1. Rechtsklick auf "TCP/UDP-Slave" und im Kontextmenü [Neu] wählen.
2. Aus der Liste "Read Holding Registers (3)" wählen.
3. Rechtsklick auf "Read Holding Registers (3)" und [Eigenschaften] wählen.
Startadresse des Lesebereichs, 1 eintragen.
4. Rechtsklick auf "Read Holding Registers (3)" und [Edit] wählen.
5. Aus der Objektauswahl die folgenden Variablen in das Register "Eingangsvariablen" ziehen.

Register-Variablen	Offset
01_Register_Bereich_SINT	0
02_Register_Bereich_SINT	1
03_Register_Bereich_REAL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Bit-Bereich

Innerhalb der Registerkarte "Bitvariablen" werden die Variablen im Bit-Bereich angelegt. Siehe auch Kapitel "Sende/Empfangsvariablen zuweisen".

HINWEIS



Um mit Modbus-Funktionscodes 3, 4, 6, 16, 23 auf die Variablen im Bit-Bereich zuzugreifen, müssen die Variablen in den Register-Bereich gespiegelt werden, siehe Kapitel "Zugriff auf die Bit-Variablen im Register-Bereich des Modbus-Slaves".

Auf die Variablen im Bit-Bereich kann nur über die Modbus-Funktionscodes 1, 2, 5, 15 zugegriffen werden. Dazu muss in den Eigenschaften des Funktionscodes die Startadresse der ersten Variable eingetragen werden.

Beispiel: Zugriff auf Variablen im Bit-Bereich des Modbus-Slave

Bit-Variablen	Bit	Register-Bit
00_BIT_Bereich_WORD	0	0.0
01_BIT_Bereich_SINT	16	1.8
02_BIT_Bereich_SINT	24	1.0
03_BIT_Bereich_REAL	32	2.0
04_BIT_Bereich_BOOL	64	4.8
05_BIT_Bereich_BOOL	65	4.9
06_BIT_Bereich_BOOL	66	4.10
07_BIT_Bereich_BOOL	67	4.11
08_BIT_Bereich_BOOL	68	4.12
09_BIT_Bereich_BOOL	69	4.13
10_BIT_Bereich_BOOL	70	4.14
11_BIT_Bereich_BOOL	71	4.15
12_BIT_Bereich_BOOL	72	4.0
13_BIT_Bereich_BOOL	73	4.1
14_BIT_Bereich_BOOL	74	4.2
15_BIT_Bereich_BOOL	75	4.3
16_BIT_Bereich_BOOL	76	4.4
17_BIT_Bereich_BOOL	77	4.5
18_BIT_Bereich_BOOL	78	4.6
19_BIT_Bereich_BOOL	79	4.7

Modbus-Master-Konfiguration des Anforderungstelegramms

So lesen Sie im Modbus-Master die Variablen *04_BIT_Bereich_BOOL* bis *06_BIT_Bereich_BOOL* ein:

1. Rechtsklick auf "TCP/UDP-Slave" und im Kontextmenü [Neu] wählen.
2. Aus der Liste "Read Coils (1)" wählen.
3. Rechtsklick auf "Read Coils (1)" und [Eigenschaften] wählen.
Startadresse des Lesebereichs, 64 eintragen.
4. Rechtsklick auf "Read Coils (1)" und [Edit] wählen.

5. Aus der Objektauswahl die folgenden Variablen in das Register "Eingangsvariablen" ziehen.

Bit-Variablen	Offset
04_BIT_Bereich_BOOL	0
05_BIT_Bereich_BOOL	1
06_BIT_Bereich_BOOL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

6.2.8 Offsets für alternative Modbus-Adressierung

Um mit den Modbus-Funktionscodes (Typ Register) auf Variablen im Bit-Bereich und mit den Modbus-Funktionscodes (Typ Bit) auf Variablen im Register-Bereich zuzugreifen, müssen die Variablen in den jeweils anderen Bereich gespiegelt werden. Innerhalb der Registerkarte "Eigenschaften/Offsets" werden die Offsets der gespiegelten Variablen eingetragen.

So spiegeln Sie die Variablen in den Bit- und Register-Bereich:

1. Im Kontextmenü des Modbus-Slave [Edit] / [Offsets] wählen und "Alternative Register/Bit- Adressierung verwenden" aktivieren.

Damit werden die Variablen in den jeweils anderen Bereich gespiegelt.

2. Offset für die gespiegelten Variablen im Bit- und Register-Bereich eintragen.

HINWEIS



Die gespiegelten Variablen im Bit-/ Register-Bereich und die vorhandenen Variablen im Bit-/ Register-Bereich dürfen sich bzgl. der Modbus-Adressen nicht überlappen.

Element	Beschreibung/ Wertebereich
Alternative Register/Bit-Adressierung verwenden	Aktiviert: Alternative Adressierung verwenden Deaktiviert: Alternative Adressierung nicht verwenden Standardwert: Deaktiviert
Register-Bereich Offset/Bit Eingänge	0 – 65535
Register-Bereich Offset/Bit Ausgänge	
Bit-Bereich Offset/Register Eingänge	
Bit-Bereich Offset/Register Ausgänge	

Zugriff auf die Register-Variablen im Bit-Bereich des Modbus-Slave

Um mit den Modbus-Funktionscodes (Typ Bit) 1, 2, 5, 15 auf die Register-Variablen zuzugreifen, müssen die Register-Variablen in den Bit-Bereich gespiegelt werden. Innerhalb der Registerkarte "Eigenschaften/Offsets" müssen die Offsets der gespiegelten Register-Variablen eingetragen werden.

Beispiel:

Bit-Bereich Offset/Register Eingänge: 8000

Bit-Bereich Offset/Register Ausgänge: 8000

Hier liegen die aus dem Register-Bereich in den Bit-Bereich gespiegelten Variablen ab Bit-Adresse 8000.

Gespiegelte Register-Variablen	Bit
00_Register_Bereich_WORD	8000
01_Register_Bereich_SINT	8016
02_Register_Bereich_SINT	8024
03_Register_Bereich_REAL	8032
04_Register_Bereich_BOOL	8064
05_Register_Bereich_BOOL	8065
06_Register_Bereich_BOOL	8066
07_Register_Bereich_BOOL	8067
08_Register_Bereich_BOOL	8068
09_Register_Bereich_BOOL	8069
10_Register_Bereich_BOOL	8070
11_Register_Bereich_BOOL	8071
12_Register_Bereich_BOOL	8072
13_Register_Bereich_BOOL	8073
14_Register_Bereich_BOOL	8074
15_Register_Bereich_BOOL	8075
16_Register_Bereich_BOOL	8076
17_Register_Bereich_BOOL	8077
18_Register_Bereich_BOOL	8078
19_Register_Bereich_BOOL	8079

Modbus-Master Konfiguration des Anforderungstelegramms

So lesen Sie im Modbus-Master die Variablen *04_Register_Bereich_BOOL* bis *06_Register_Bereich_BOOL* ein:

1. Rechtsklick auf "TCP/UDP-Slave" und im Kontextmenü [Neu] wählen.
2. Aus der Liste "Read Coils (1)" wählen.
3. Rechtsklick auf "Read Coils (1)" und [Eigenschaften] wählen.
Startadresse des Lesebereichs, 8064 eintragen.
4. Rechtsklick auf "Read Coils (1)" und [Edit] wählen.
5. Aus der Objektauswahl die folgenden Variablen in das Register "Eingangsvariablen" ziehen.

Bit-Variablen	Offset
04_Register_Bereich_BOOL	0
05_Register_Bereich_BOOL	1
06_Register_Bereich_BOOL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

Zugriff auf die Bit-Variablen im Registerbereich des Modbus-Slave

Um mit den Modbus-Funktionscodes (Typ Register) 3, 4, 6, 16, 23 auf die Bit-Variablen zuzugreifen, müssen die Bit-Variablen in den Register-Bereich gespiegelt werden. Innerhalb der Registerkarte "Eigenschaften/Offsets" müssen die Offsets der gespiegelten Bit-Variablen eingetragen werden.

Beispiel:

Register-Bereich Offset/Bit Eingänge: 1000

Register-Bereich Offset/Bit Ausgänge: 1000

Hier liegen die aus dem Bit-Bereich in den Register Bereich gespiegelten Variablen ab Register-Adresse 1000.

Gespiegelte Bit-Variablen	Register.Bit
00_BIT_Bereich_WORD	1000.0
01_BIT_Bereich_SINT	1001.8
02_BIT_Bereich_SINT	1001.0
03_BIT_Bereich_REAL	1002.0
04_BIT_Bereich_BOOL	1004.8
05_BIT_Bereich_BOOL	1004.9
06_BIT_Bereich_BOOL	1004.10
07_BIT_Bereich_BOOL	1004.11
08_BIT_Bereich_BOOL	1004.12
09_BIT_Bereich_BOOL	1004.13
10_BIT_Bereich_BOOL	1004.14
11_BIT_Bereich_BOOL	1004.15
12_BIT_Bereich_BOOL	1004.0
13_BIT_Bereich_BOOL	1004.1
14_BIT_Bereich_BOOL	1004.2
15_BIT_Bereich_BOOL	1004.3
16_BIT_Bereich_BOOL	1004.4
17_BIT_Bereich_BOOL	1004.5
18_BIT_Bereich_BOOL	1004.6
19_BIT_Bereich_BOOL	1004.7

Modbus-Master Konfiguration des Anforderungstelegramms

So lesen Sie im Modbus-Master die Variablen *01_BIT_Bereich_SINT* bis *03_BIT_Bereich_REAL* ein:

1. Rechtsklick auf "TCP/UDP-Slave" und im Kontextmenü [Neu] wählen.
2. Aus der Liste "Read Holding Register (3)" wählen.
3. Rechtsklick auf "Read Holding Register (3)" und [Eigenschaften] wählen.

Startadresse des Lesebereichs, 1001 eintragen.

4. Rechtsklick auf "Read Holding Register (3)" und [Edit] wählen.
5. Aus der Objektauswahl die folgenden Variablen in das Register "Eingangsvariablen" ziehen.

Bit-Variablen	Offset
01_BIT_Bereich_SINT	0
02_BIT_Bereich_SINT	1
03_BIT_Bereich_REAL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich "Ausgangsvariablen" öffnen und [Neue Offsets] wählen, um die Offsets der Variablen neu zu nummerieren.

6.2.9 Control-Panel (Modbus-Slave)

Im Control-Panel kann der Anwender die Einstellungen des Modbus-Slave überprüfen und steuern. Zudem werden aktuelle Statusinformationen (z. B. Master-Zustand usw.) des Slaves angezeigt.

So öffnen Sie das Control-Panel zur Überwachung des Modbus-Slave:

1. Im Strukturbaum [Hardware] und im Kontextmenü [Online] wählen.
2. Im System-Login Zugangsdaten eingeben um die Online-Ansicht der Hardware zu öffnen.
3. Doppelklick auf COM-Modul und im Strukturbaum [Modbus Slave] wählen.

Kontextmenü (Modbus-Slave)

Aus dem Kontextmenü des selektierte Modbus-Slave kann das folgende Kommando gewählt werden:

Zurücksetzen Statistik: Setzt die statistischen Daten (Zykluszeit min, max usw.) auf null zurück.

Anzeigefeld (Modbus-Slave)

Im Anzeigefeld werden die folgenden Werte des selektierten Modbus-Slaves angezeigt.

Element	Beschreibung
Name	Name des Modbus-Slave.
Projektiertes μ P-Budget [%]	Siehe Kapitel "Menüfunktionen des Modbus-Slave-Set".
Aktuelles μ P-Budget [%]	
SRS redundante Baugruppe (nicht unterstützt)	SRS des redundanten COM-Moduls.
Antwortzeit [ms]	Zeitspanne nach dem Empfang einer Anfrage, innerhalb welcher der Modbus-Slave antwortet.

Anzeigefeld (Masterdaten)

In dem Anzeigefeld Masterdaten werden die folgenden Werte angezeigt.

Element	Beschreibung
Name	Name der Masterdaten.
Anfragen	Gesamtanzahl aller Master Anfragen seit dem letzten Zähler Reset.
Gültige Anfragen	Anzahl gültiger Master-Anfragen seit dem letzten Zähler-Reset.
Ungültige Anfragen	Anzahl ungültiger Master-Anfragen seit dem letzten Zähler-Reset. Zu ungültigen Anfragen zählen nur Anfragen, die vom Master quittiert wurden. Fehlerhaft empfangene Anfragen mit CRC-Fehler werden automatisch verworfen.
Master-Timeout [ms]	Timeoutzeit. Innerhalb der Timeoutzeit muss der Slave mindestens einen Request von seinem Master erhalten haben. Empfängt der Slave innerhalb der Timeoutzeit keinen Request, so wird der <i>Master Verbindungsstatus</i> auf "nicht verbunden" gesetzt.
Verbindungszustand	0 = Nicht überwacht (Master Request Timeout ist Null) 1 = Nicht verbunden 2 = Verbunden
Zeitüberschreitungen bei Antworten	Anzahl der <i>Zeitüberschreitungen bei Antworten</i> seit dem letzten Rücksetzen aller Zähler oder Einschalten. Die <i>Zeitüberschreitungen bei Antworten</i> ist die maximale Zeit, die bis zur Empfangsbestätigung einer Nachricht beim Absender vergehen darf.
Verworfenne Anfragen	Anzahl der verworfenen Master-Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Maximaler Pufferfüllstand für Anfragen	Maximale Anzahl gleichzeitiger Master-Anfragen.
Durchschnittlicher Pufferfüllstand für Anfragen	Durchschnittliche Anzahl gleichzeitiger Master-Anfragen.

6.2.10 Fehlercodes der Modbus TCP/IP-Verbindung

Die Fehlercodes der Modbus TCP/IP-Verbindung werden im Dialogfenster "Diagnose" angezeigt.

Fehlercode	Beschreibung
35	Operation ist blockiert.
48	Port ist bereits in Verwendung.
50	Netzwerk läuft nicht.
53	Verbindung durch Software abgebrochen.
54	Verbindung durch Partner abgebrochen.
55	Kein Pufferspeicher mehr verfügbar.

Fehlercode	Beschreibung
60	Timeout aufgetreten. Verbindung geschlossen.
61	Verbindung abgewiesen (durch Partner).
65	Kein Routing-Eintrag zum Partner vorhanden.

7 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 Send & Receive TCP-Protokoll unterstützt die Steuerung nahezu jedes Fremdsystem sowie PCs mit vorhandener Socket-Schnittstelle (z. B. Winsock.dll) zu TCP/IP.

Send & Receive TCP ist kompatibel zur 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).

7.1 Systemanforderungen

Benötigte Ausstattung und Systemanforderungen.

Element	Beschreibung
Steuerung	MOVISAFE® HM31 ab CPU BS V7 und COM BS V12.
CPU-Modul	Die Ethernet-Schnittstellen des Prozessormoduls können für Send & Receive TCP nicht verwendet werden.
COM-Modul	Ethernet 10/100BaseT. Es kann für jedes COM-Modul ein Send & Receive TCP-Protokoll konfiguriert werden.
Aktivierung	Die Freischaltung erfolgt per Software-Freischaltcode (siehe Kapitel "Registrierung und Aktivierung der Protokolle")

Eigenschaften des Send & Receive-TCP-Protokolls.

Element	Beschreibung
Sicherheitsgerichtet	Nein.
Datenaustausch	Zyklischer und azyklischer Datenaustausch über TCP/IP.
Funktionsbausteine	Die Send & Receive-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	Hinweis: 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.
Max. Größe der Empfangsdaten	

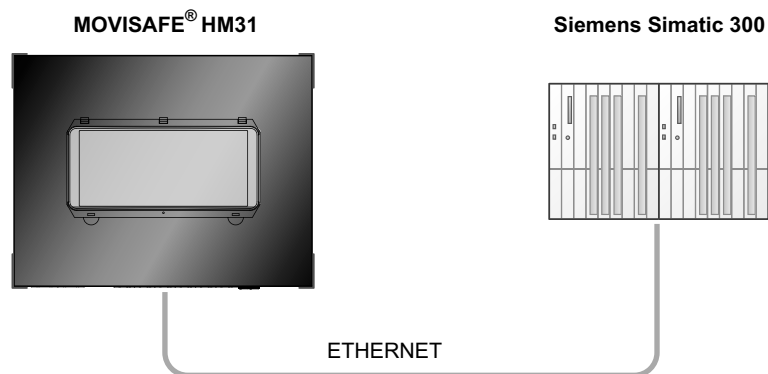
7.1.1 Anlegen eines Send & Receive-TCP-Protokolls

So legen Sie eine neues Send & Receive-TCP-Protokoll an:

1. Öffnen Sie im Strukturbaum das Menü [Konfiguration] / [Ressource] / [Protokolle].

2. Um ein neues Send & Receive TCP-Protokoll hinzuzufügen, wählen Sie im Kontextmenü "Protokolle" den Menüpunkt [Neu] / [Send/Receive over TCP].
3. Im Kontextmenü des Send/Receive-over-TCP-Protokoll den Menüpunkt [Eigenschaften] / [Allgemein] wählen und dort das COM-Modul auswählen.

7.2 Beispiel Send & Receive TCP-Konfiguration



12757488907

In diesem Beispiel wird das Protokoll Send & Receive (S&R) over TCP in MOVISAFE[®] eingerichtet. MOVISAFE[®] HM31 soll zyklisch über S&R TCP mit einer Siemens Steuerung (z. B. SIMATIC 300) kommunizieren.

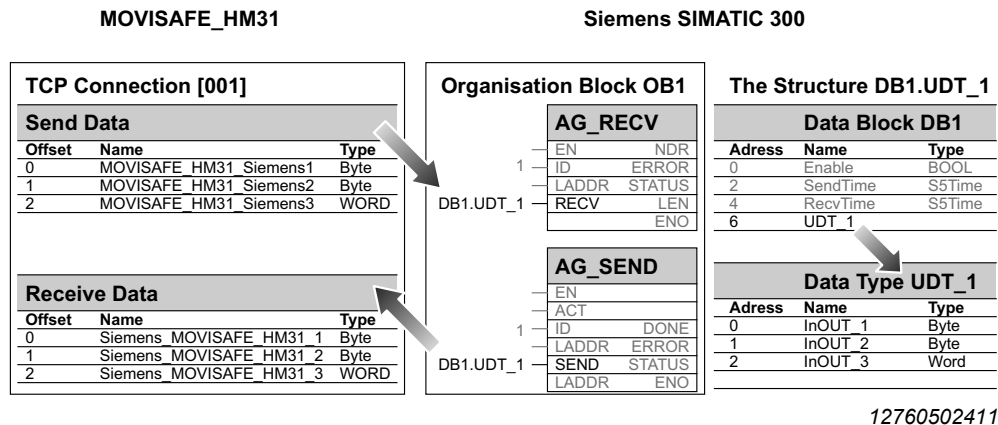
Dabei ist MOVISAFE[®] HM31 (Client) die aktive Station, die die TCP-Verbindung zur passiven Siemens SIMATIC 300 (Server) aufbaut. Nach dem Verbindungsaufbau sind aber beide Steuerungen gleichberechtigt und können jederzeit senden und empfangen.

Bei der Zusammenschaltung von MOVISAFE[®] HM31 und der Siemens SIMATIC 300 ist folgendes zu beachten:

- Für MOVISAFE[®] HM31 gelten die im Kapitel "Systemanforderungen" beschriebenen Anforderungen.
- MOVISAFE[®] HM31 und die Siemens SIMATIC 300 werden über ihre Ethernet-Schnittstellen miteinander verbunden.
- Die Steuerungen MOVISAFE[®] HM31 und Siemens SIMATIC 300 müssen sich im gleichen Subnetz befinden oder bei Verwendung eines Routers die entsprechenden Routing-Einträge besitzen.

In diesem Beispiel sollen zwei BYTES und ein WORD von der Sicherheitssteuerung MOVISAFE[®] HM31 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 Sicherheitssteuerung MOVISAFE[®] HM31 zurück.

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



Konfigurationsbeschreibung MOVISAFE® HM31:

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 Sicherheitssteuerung MOVISAFE® HM31 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 Sicherheitssteuerung MOVISAFE® HM31 müssen mit der Adresse und den Typen der Variablen im Datentyp <i>UDT_1</i> der SIMATIC 300 übereinstimmen

Konfigurationsbeschreibung 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.
AG_RECV (FC6)	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.
AG_SEND (FC5)	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 DB1	Der Datentyp <i>UDT_1</i> wird im Datenbaustein <i>DB1</i> definiert.
Datentyp UDT_1	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.

7.2.1 Send & Receive TCP-Konfiguration der Steuerung SIMATIC 300

HINWEIS



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.

So erstellen Sie den Send & Receive 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.

Datentyp UDT1 erstellen

So erstellen Sie den 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* erstellen (siehe folgende Variablenliste).

Adresse	Name	Typ	Anfangswert
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	

HINWEIS



Beim zyklischen 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 Steuerung MOVISAFE® HM31 eingefügt werden (siehe Kapitel "Fremdsysteme mit Pad Bytes").

Datenbaustein DB1 für die Funktionsbausteine FC5 und FC6 erstellen

So erstellen Sie den Datenbaustein DB1 für die Funktionsbausteine FC5 und FC6:

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 parametrieren (siehe folgende Variablenliste).

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	Enable	BOOL	FALSE
+2.0	SendTime	SSTIME	SST#100MS
+4.0	RecvTime	SSTIME	SST#10MS
+6.0	UDT_1	„UDT_1“	
=10.0		END_STRUCT	

Symbole im Symboleditor erstellen

So erstellen Sie im Symboleditor die folgenden Symbole:

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 ergänzen (siehe folgende Übersicht).

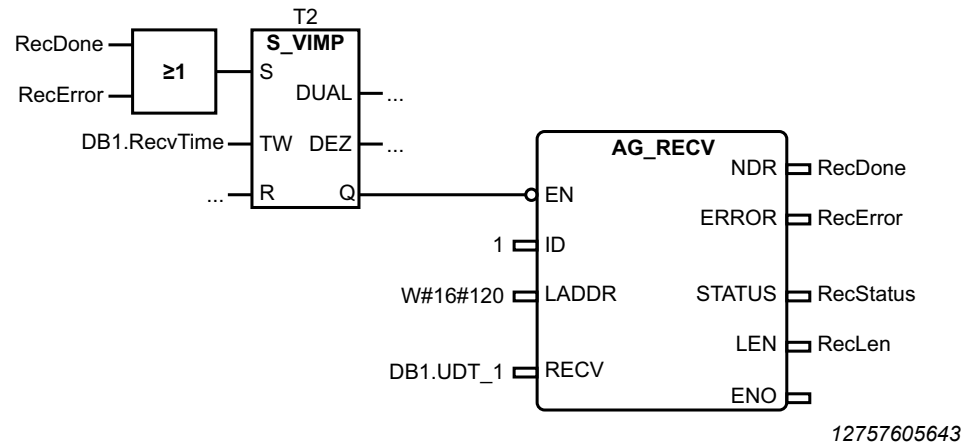
Symbol	Adresse	Typ
RecDone	M 1.0	BOOL
RecError	M 1.1	BOOL
SendDone	M 1.2	BOOL
SendError	M 1.3	BOOL
RecStatus	MW 1	WORD
RecLen	MW 3	INT
SendStatus	MW 5	WORD

FC-Baustein AG_RECV (FC6) erstellen

So erstellen Sie den FC-Baustein AG_RECV (FC6):

1. Dialogfenster "KOP/AWL/FUP" öffnen.
 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.



FC-Baustein AG_SEND (FC 5) erstellen

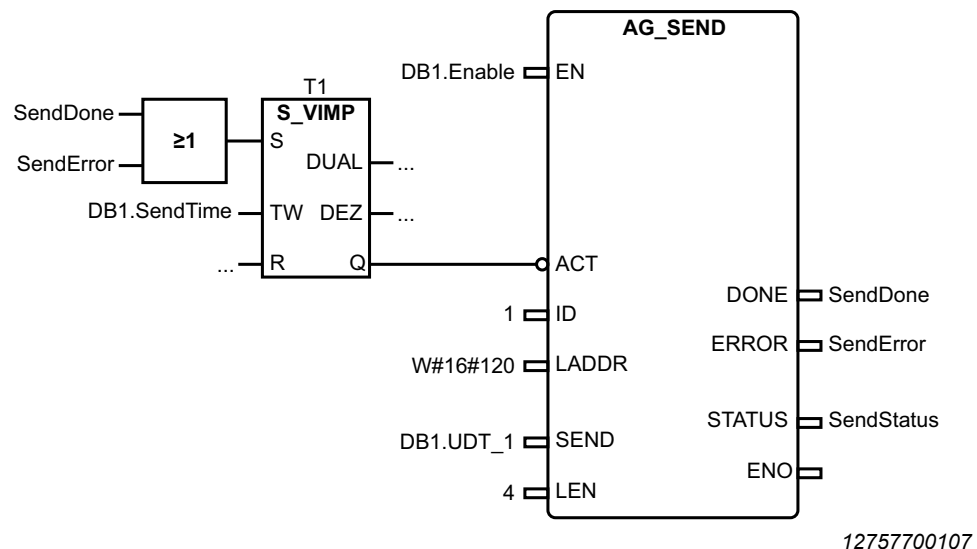
So erstellen Sie den FC-Baustein *AG_SEND* (FC5):

1. Dialogfenster "KOP/AWL/FUP" öffnen.

Folgende FC-Bausteine nacheinander aus der Struktur im linken Teil des Symatic-Managers auswählen:

- 1 Oder-Gatter
- 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.



Code in die Steuerung SIMATIC-300 laden

So laden Sie den Code in die SIMATIC-300-Steuerung:

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

7.2.2 Send & Receive TCP-Konfiguration von MOVISAFE® HM31

Für die Konfiguration der Steuerung MOVISAFE® HM31 und den Umgang mit dem Programmierwerkzeug SILworX® wird das Handbuch "Erste Schritte SILworX®" empfohlen.

So erstellen Sie die folgenden Globalen Variablen im Variableneditor:

1. Im Strukturbaum [Konfiguration] / [Globale Variablen] wählen.
2. Wählen Sie im Kontextmenü den Menüpunkt "Edit" wählen.
3. Globale Variablen erstellen (siehe folgende Tabelle).

Name	Typ
Siemens_PFF-HM31	Byte
Siemens_PFF-HM31	Byte
Siemens_PFF-HM31	WORD
PFF-HM31_Siemens1	Byte
PFF-HM31_Siemens2	Byte
PFF-HM31_Siemens3	WORD

Send & Receive TCP-Protokoll in der Ressource erstellen

So erstellen Sie das Send & Receive TCP-Protokoll in der Ressource:

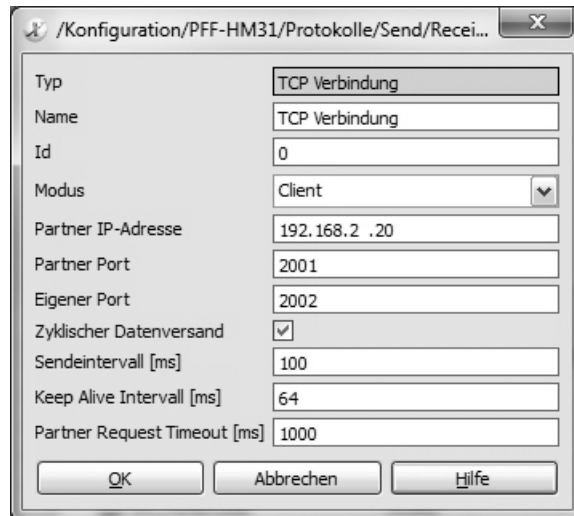
1. Im Strukturbaum [Konfiguration] / [Ressource] öffnen.
2. Rufen Sie im Menü "Protokolle" das Kontextmenü "Neu" auf. Wählen Sie dort den Menüpunkt "Send/Receive over TCP" und tragen Sie einen Protokollnamen ein.
3. Um das neue Protokoll zu erstellen, bestätigen Sie Ihre Eingaben mit "OK".
4. Rufen Sie im Menüpunkt "Send/Receive over TCP" das Kontextmenü "Eigenschaften" auf und wählen Sie hier den Eintrag "COM Modul" aus. Die restlichen Parameter behalten die Standardwerte.

TCP-Verbindung erstellen

So erstellen Sie die TCP-Verbindung:

1. Rufen Sie im Menüpunkt "Send/Receive over TCP" das Kontextmenü "Neu" auf. wählen Sie den Eintrag "TCP-Verbindung".

- Rufen Sie im Menüpunkt "TCP-Verbindung" das Kontextmenü "Eigenschaften" auf. Konfigurieren Sie die Eigenschaften wie im folgenden Bild dargestellt.



12846254859

HINWEIS



Wenn ein zyklischer Datenaustausch zwischen zwei Steuerungen parametrieren werden soll, muss im Dialog "Eigenschaften der TCP-Verbindung" das Kontrollfeld "Zyklischer Datenversand" markiert sein.

Eingangsdaten der Steuerung MOVISAFE® HM31 konfigurieren

So konfigurieren Sie die Empfangsdaten der MOVISAFE® HM31:

- Rufen Sie im Menüpunkt "TCP-Verbindung" das Kontextmenü "Edit" auf und wählen Sie die Registerkarte "Prozessvariablen".
- In der Objektauswahl die folgenden Globalen Variablen auswählen und diese per Drag & Drop in den Bereich "Eingangssignale" ziehen.

Globale Variable	Typ
Siemens_PFF-HM31	Byte
Siemens_PFF-HM31	Byte
Siemens_PFF-HM31	WORD

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

HINWEIS



Beachten Sie, dass die Offsets der Variablen in MOVISAFE® HM31 mit den Adressen der Variablen im Datentyp *UDT_1* der SIMATIC 300 übereinstimmen müssen.

Sendedaten der Steuerung MOVISAFE® HM31 konfigurieren

So konfigurieren Sie die Sendedaten der Steuerung MOVISAFE®:

- Rufen Sie im Menüpunkt "TCP-Verbindung" das Kontextmenü "Edit" auf und wählen Sie die Registerkarte "Prozessvariablen".

2. In der Objektauswahl die folgenden Globalen Variablen auswählen und diese per Drag & Drop in den Bereich "Eingangssignale" ziehen.

Globale Variable	Typ
PFF-HM31_Siemens1	Byte
PFF-HM31_Siemens2	Byte
PFF-HM31_Siemens3	WORD

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

HINWEIS



Beachten Sie, dass die Offsets der Variablen in MOVISAFE® HM31 mit den Adressen der Variablen im Datentyp *UDT_1* der SIMATIC 300 übereinstimmen müssen.

Send & Receive TCP-Konfiguration verifizieren

So verifizieren Sie die Send & Receive TCP-Konfiguration:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [Send/Receive over TCP] öffnen.
2. Auf Schaltfläche [Verifikation] in Action Bar klicken und mit "OK" bestätigen.

7.3 Menüfunktionen im Send & Receive TCP-Protokoll

7.3.1 Edit

Das Dialogfenster "Edit" des S&R-TCP-Protokolls enthält das folgende Register.

Systemvariablen

Mit den Systemvariablen kann der Zustand des Send & Receive TCP-Protokolls im Anwenderprogramm ausgewertet werden.

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

7.3.2 Eigenschaften

Der Datenaustausch über eine TCP-Verbindung erfolgt entweder zyklisch oder azyklisch. Für den azyklischen Datenaustausch werden die Send & Receive-TCP-Funktionsbausteine benötigt.

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

Allgemeine Eigenschaften

Name	Beschreibung
Typ	Send/Receive over TCP.
Name	Name für das Send/Receive-over-TCP-Protokoll. Maximal 31 Zeichen.
Modul	Auswahl des COM-Moduls, auf dem dieses Protokoll abgearbeitet wird.
Max. µP-Budget aktivieren	<ul style="list-style-type: none"> • Aktiviert: Limit des µP-Budget aus dem Feld <i>Max. µP-Budget</i> in % übernehmen. • Deaktiviert: Kein Limit des µP-Budget für dieses Protokoll verwenden.
Max. µP-Budget in %	<p>Maximales µP-Budget des Moduls, das bei der Abarbeitung des Protokolls produziert werden darf.</p> <p>Wertebereich: 1 – 100 %</p> <p>Standardwert: 30 %</p>
Verhalten bei CPU/COM-Verbindungsverlust	<p>Bei Verbindungsverlust des Prozessormoduls zum Kommunikationsmodul werden, in Abhängigkeit dieses Parameters, die Eingangsvariablen entweder initialisiert oder unverändert im Prozessormodul verwendet.</p> <ul style="list-style-type: none"> • Initialdaten annehmen: Eingangsvariablen werden auf die Initialwerte zurückgesetzt. • Letzten Wert beibehalten: Eingangsvariablen behalten den letzten Wert.

CPU/COM

Die vorgegebenen Parameter sorgen für den schnellstmöglichen Datenaustausch der Send & Receive 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.

HINWEIS

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

Name	Beschreibung
Aktualisierungsintervall der Prozessdaten in ms	<p>Aktualisierungszeit in Millisekunden, mit der die Daten im Send & Receive-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.</p> <p>Wertebereich: 0 bis $(2^{31} - 1)$</p> <p>Standardwert: 0</p>
Prozessdatenkonsistenz erzwingen	<ul style="list-style-type: none"> Aktiviert: Transfer der Send & Receive TCP-Daten von der CPU zur COM innerhalb eines Zyklus der CPU. Deaktiviert: Transfer der Send & Receive TCP-Daten (maximal 1100 Byte pro Datenrichtung) von der CPU zur COM über mehrere Zyklen der CPU.

7.4 Menüfunktionen TCP-Verbindung

7.4.1 Edit

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

Prozessvariablen

- Eingangsvariable
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.

7.4.2 Systemvariablen

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

Name	Beschreibung
Bytes empfangen	Anzahl Bytes, die bisher empfangen wurden.
Bytes versenden	Anzahl Bytes, die bisher gesendet wurden.
Errorcode	Fehlercode der TCP-Verbindung.
Errorcode Zeitstempel in ms	Millisekunden-Anteil des Zeitstempels. Zeitpunkt des Fehlerintritts.

Name	Beschreibung
Errorcode Zeitstempel in s	Sekunden-Anteil des Zeitstempels. Zeitpunkt des Fehlereintritts.
Partner Request Timeout	Bei zyklischer Datenübertragung: Timeoutzeit, innerhalb der nach Datenversand mindestens eine Datensendung vom Kommunikationspartner empfangen werden muss. 0 : Aus 1 : 1 bis ($2^{31} - 1$) ms
Partner Verbindungszustand	Wird keine Datensendung innerhalb der Timeoutzeit empfangen, so wird der Status "Partner Verbindungszustand" auf nicht verbunden gesetzt und die Verbindung neu aufgesetzt. 0: Keine Verbindung 1 : Verbindung OK
Status	Verbindungsstatus der TCP-Verbindung.

7.4.3 Eigenschaften

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

Name	Beschreibung
Typ	TCP-Verbindung.
Name	Beliebiger, eindeutiger Name für eine TCP-Verbindung. Maximal 31 Zeichen.
ID	Beliebige, aber eindeutige Identifikationsnummer ID für jede TCP-Verbindung. Die ID wird auch als Referenz in den Send & Receive TCP-Funktionsbausteinen benötigt. Wertebereich: 0 – 255 Standardwert: 0

Name	Beschreibung
Modus	<ul style="list-style-type: none"> • 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 der 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	<p>IP-Adresse des Kommunikationspartners. 0.0.0.0: beliebige IP-Adresse ist erlaubt Gültiger Bereich: 1.0.0.0 bis 223.255.255.255, außer: 127.x.x.x Standardwert: 0</p>
Partner Port	<p>Port des Kommunikationspartners. 0: beliebiger Port. Reservierte oder bereits belegte Ports (1 – 1024) werden von dem COM-BS abgelehnt. Wertebereich 0 – 65535 Standardwert: 0</p>
Eigener Port	<p>Eigener Port. 0: beliebiger Port Reservierte oder bereits belegte Ports (1 – 1024) werden vom COM-BS abgelehnt. Wertebereich 0 – 65535 Standardwert: 0</p>

Name	Beschreibung
Zyklischer Datenversand	<ul style="list-style-type: none"> 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	<p>Nur editierbar bei zyklischem Datenversand. Hier wird das Sendeintervall eingestellt.</p> <p>Wertebereich 10 – 2 147 483 647 ms (Kleinere Werte werden auf 10 ms aufgerundet)</p> <p>Standardwert: 0</p>
KeepAlive Intervall in s	<p>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).</p>
Partner Request Timeout in ms	<p>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 "Partner Verbindungszustand" auf nicht verbunden 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 Sekunden, wenn "PartnerRequestTimeout" = 0 ist, neu auf. Die passive Seite öffnet den Port bereits nach der halben Zeit.</p>

7.5 Datenaustausch

Send & Receive 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.

Send & Receive 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.

Send & Receive 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).

Zudem werden fünf Send & Receive-TCP-Funktionsbausteine bereitgestellt, mit denen die Kommunikation über das Anwenderprogramm gesteuert und individuell angepasst werden kann. Mit den Send & Receive TCP-Funktionsbausteinen können beliebige Protokolle (z. B. Modbus), die über TCP übertragen werden, gesendet und empfangen werden.

7.5.1 TCP-Verbindungen

Für jede Verbindung über Send & Receive TCP mit einem Kommunikationspartner muss mindestens eine TCP-Verbindung in der Steuerung MOVISAFE® HM31 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 Steuerung MOVISAFE® HM31 erstellt werden. Die erstellten TCP-Verbindungen müssen unterschiedliche Identifikationsnummern und unterschiedliche Adressen / Ports besitzen.

So erstellen Sie eine neue TCP-Verbindung:

HINWEIS



Die Steuerung MOVISAFE® HM31 und das Fremdsystem müssen sich im gleichen Subnetz befinden oder bei Verwendung eines Routers die entsprechenden Routingeinträge besitzen.

1. Im Strukturbaum [Konfiguration] / [Ressource] öffnen.
2. Rufen Sie im Menü "Protokolle" das Kontextmenü "Neu" auf und wählen Sie den Menüpunkt "Send/receive over TCP". Geben Sie einen Namen für das Protokoll ein.
3. Um ein neues Protokoll zu erstellen, bestätigen Sie mit "OK".
4. Rufen Sie im Menüpunkt "Send/Receive over TCP" das Kontextmenü Eigenschaften auf und wählen Sie dort "COM Modul" aus.
5. Die restlichen Parameter behalten die Standardwerte.

7.5.2 Zyklischer Datenaustausch

Wird zyklischer Datenaustausch verwendet, dann muss ein Sendeintervall in der Steuerung MOVISAFE® HM31 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 "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.

HINWEIS



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.

7.5.3 Azyklischer Datenaustausch mit Funktionsbausteinen

Der azyklische Datenaustausch wird in MOVISAFE® HM31 vom Anwenderprogramm über die Send & Receive TCP-Funktionsbausteine gesteuert. Somit ist es möglich, mit einem Timer oder einen mechanischen Schalter an einem physikalischen Eingang der Steuerung MOVISAFE® HM31, den Datenaustausch zu steuern.

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

HINWEIS



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.

7.5.4 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.

7.5.5 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 ≥ 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.

7.6 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 Steuerung MOVISAFE® HM31 müssen für die Pad-Bytes Dummy-Bytes an den entsprechenden Stellen eingefügt werden.

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	InOut_1	BYTE	B#16#0
+2.0	InOut_2	WORD	W#16#0
=4.0		END_STRUCT	

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

	Name	Datentyp	Offset	Globale Variable
1	Dummy	BYTE	1	Dummy
2	InOut_1	BYTE	0	InOut_1
3	InOut_3	WORD	2	InOut_3

12846249227

In der Steuerung MOVISAFE® HM31 muss ein Dummy-Byte eingefügt werden, damit die Variable *InOut_3* den gleichen Offset wie in der Siemens Steuerung hat.

7.7 Send & Receive TCP-Funktionsbausteine

Wenn die zyklische Datenübertragung zu unflexibel ist, können Daten auch mittels der Send & Receive TCP-Funktionsbausteine gesendet und empfangen werden. Die Option "Zyklischer Datenversand" muss in der verwendeten TCP-Verbindung deaktiviert werden.

Mit den Send & Receive TCP-Funktionsbausteinen kann der Anwender die Datenübertragung über TCP/IP den Erfordernissen seines Projekts optimal anpassen.

Die Funktionsbausteine werden im Anwenderprogramm parametrierbar. So können die Funktionen (Senden, Empfangen, Reset) der Steuerung MOVISAFE® HM31 im Anwenderprogramm gesetzt und ausgewertet werden.

Send & Receive TCP-Funktionsbausteine werden nur für den azyklischen Datenaustausch benötigt. Für den zyklischen Datenaustausch zwischen Server und Client sind diese Funktionsbausteine nicht erforderlich.

HINWEIS

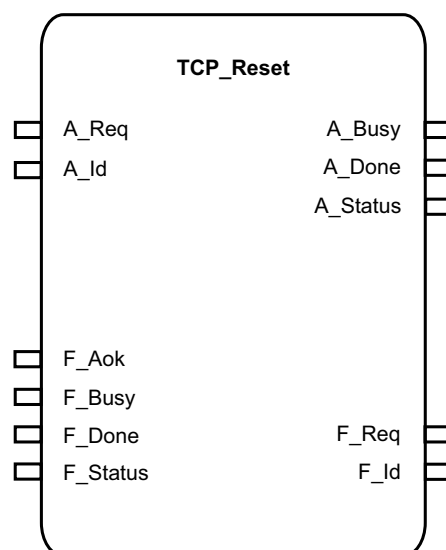


- Die Send & Receive TCP-Funktionsbausteine müssen über die Bibliothek *TCPLib.A3* eingebunden werden. Die Bibliothek *TCPLib.A3* finden Sie auf der SIL-worX®-Installations-CD.
- Die Konfiguration der Send & Receive TCP-Funktionsbausteine wird im Kapitel „Konfiguration der Funktionsbausteine“ beschrieben.

Es stehen die folgenden Funktionsbausteine zur Verfügung.

Funktionsbaustein	Beschreibung der Funktion
TCP_Reset (siehe Kapitel "TCP_Reset")	Rücksetzen einer TCP-Verbindung.
TCP_Send (siehe Kapitel TCP_Send")	Senden von Daten.
TCP_Receive (siehe Kapitel "TCP_Receive")	Empfangen von Datenpaketen fester Länge
TCP_ReceiveLine (siehe Kapitel "TCP_ReceiveLine")	Empfang einer ASCII-Zeile.
TCP_ReceiveVar (siehe Kapitel "TCP_ReceiveVar")	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.

7.7.1 TCP_Reset



12757709579

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).

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe auch Kapitel „Konfiguration der Funktionsbausteine“).

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

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> werden Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD

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".

HINWEIS



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.

Verbinden Sie die F-Eingänge des Funktionsbausteins "TCP_Reset" im Anwenderprogramm mit den gleichen Variablen, 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

Verbinden Sie die F-Ausgänge des Funktionsbausteins "TCP_Reset" im Anwenderprogramm mit den gleichen Variablen, mit denen später auch die Eingänge des Funktionsbausteins "Reset" im Strukturbaum verbunden werden.

F-Ausgänge	Typ
F_Req	BOOL
F_Id	INT

Funktionsbaustein „Reset“ im Strukturbaum erstellen

So erstellen Sie einen zugehörigen Funktionsbaustein "Reset" im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [Send Receive over TCP] / [Funktionsbausteine] / [Neu] öffnen.
2. Wählen Sie den Funktionsbaustein "Reset" aus. Markieren Sie den Funktionsbaustein "Reset" und rufen Sie im Kontextmenü den Menüpunkt "Edit" auf.
3. Die Variablenzuordnung des Funktionsbausteins wird geöffnet

Verbinden Sie die Eingänge des Funktionsbausteins "Reset" im Strukturbaum mit den gleichen Variablen, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins "TCP_Reset" im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	INT
REQ	BOOL

4. Verbinden Sie die Ausgänge des Funktionsbausteins "Reset" im Strukturbaum mit den gleichen Variablen, 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

Bedienung des Funktionsbausteins „TCP_Reset“

Zur Bedienung des Funktionsbausteins "TCP_Reset" sind die folgenden Schritte erforderlich:

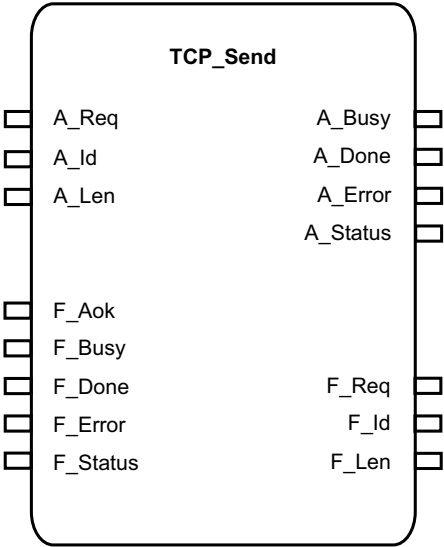
1. Setzen Sie im Anwenderprogramm am Eingang *A_ID* die Identifikationsnummer der gestörten TCP-Verbindung.
2. Setzen Sie im Anwenderprogramm den Eingang *A_Req* auf TRUE.

HINWEIS

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

3. 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.

7.7.2 TCP_Send



12757780619

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.



HINWEIS

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe auch Kapitel „Konfiguration der Funktionsbausteine“).

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 dem die Daten gesendet werden sollen.	INT
A_Len	Anzahl der zu sendenden Variablen in Bytes. <i>A_Len</i> muss größer als Null sein und darf nicht innerhalb einer Variablen enden.	INT

A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Sendevorgang ist noch nicht beendet.	BOOL
A_Done	TRUE: Der Sendevorgang wurde erfolgreich beendet.	BOOL
A_Error	<ul style="list-style-type: none">TRUE: Ein Fehler ist aufgetretenFALSE: Kein Fehler	BOOL
A:Status	Am Ausgang <i>A_Status</i> werden Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD

21233780 / DE – 07/2014

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

HINWEIS



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.

Verbinden Sie die F-Eingänge des Funktionsbausteins "TCP_Send" im Anwenderprogramm mit den gleichen Variablen, 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

Verbinden Sie die F-Ausgänge des Funktionsbausteins "TCP_Send" im Anwenderprogramm mit den gleichen Variablen, mit denen später auch die Eingänge des Funktionsbausteins "Senden" im Strukturbaum verbunden werden.

F-Ausgänge	Typ
F_Id	INT
F_Len	INT
F_Req	BOOL

Funktionsbaustein „Senden“ im Strukturbaum erstellen

So erstellen Sie einen zugehörigen Funktionsbaustein "Senden" im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [Send Receive over TCP] / [Funktionsbausteine] / [Neu] öffnen.
2. Wählen Sie den Funktionsbaustein "Senden" aus. Markieren Sie den Funktionsbaustein "Reset" und rufen Sie im Kontextmenü den Menüpunkt "Edit" auf.
3. Die Variablenzuordnung des Funktionsbausteins wird geöffnet

Verbinden Sie die Eingänge des Funktionsbausteins "Senden" im Strukturbaum mit den gleichen Variablen, mit denen zuvor auch die F-Ausgänge des Funktionsbausteins "TCP_Send" im Anwenderprogramm verbunden wurden.

Eingänge	Typ
ID	INT
LEN	INT
REQ	BOOL

4. Verbinden Sie die Ausgänge des Funktionsbausteins "Senden" im Strukturbaum mit den gleichen Variablen, 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

Daten	Beschreibung
Sendedaten	Im Register "Prozessvariablen" können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen identisch mit den Offsets und den Typen der Variablen des Kommunikationspartners sein.

Bedienung des Funktionsbausteins „TCP_Send“

Zur Bedienung des Funktionsbausteins "TCP_Send" sind die folgenden Schritte erforderlich:

HINWEIS



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. Setzen Sie im Anwenderprogramm am Eingang *A_ID* die TCP-Verbindungs-ID.
2. Setzen Sie im Anwenderprogramm die Länge der zu sendenden Variablen am Eingang *A_Req* in Byte.
3. Setzen Sie im Anwenderprogramm den Eingang *A_Req* auf TRUE.

HINWEIS

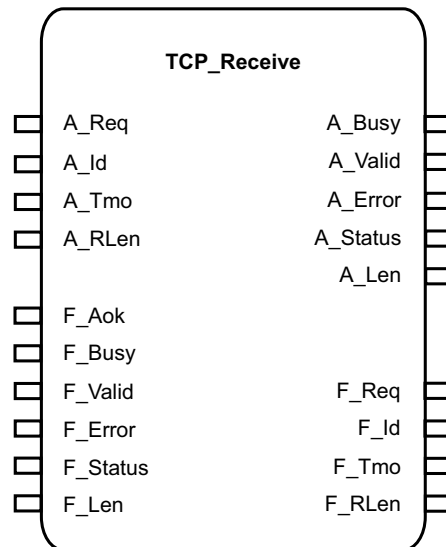


Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis die Variablen gesendet wurden. Danach wechseln 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.

7.7.3 TCP_Receive



12757827339

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.

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe auch Kapitel „Konfiguration der Funktionsbausteine“).

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
A-Ausgänge	Beschreibung	Typ
A_Busy	TRUE: Der Empfang der Daten noch nicht beendet.	BOOL

21233780 / DE – 07/2014

A-Ausgänge	Beschreibung	Typ
A_Done	TRUE: Der Empfang der Daten wurde fehlerfrei beendet.	BOOL
A_Error	<ul style="list-style-type: none"> 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

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".

HINWEIS



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.

Verbinden Sie die F-Eingänge des Funktionsbausteins "TCP_Receive" im Anwenderprogramm mit den gleichen Variablen, 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

Verbinden Sie die F-Ausgänge des Funktionsbausteins "TCP_Receive" im Anwenderprogramm mit den gleichen Variablen, mit denen später auch die Eingänge des Funktionsbausteins "Empfangen" im Strukturbaum verbunden werden.

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

Zugehörigen Funktionsbaustein „Empfangen“ im Strukturbaum erstellen

So erstellen Sie einen zugehörigen Funktionsbaustein "Empfangen" im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [Send Receive over TCP] / [Funktionsbausteine] / [Neu] öffnen.

- Wählen Sie den Funktionsbaustein "Empfangen" aus. Markieren Sie den Funktionsbaustein "Empfangen" und rufen Sie im Kontextmenü den Menüpunkt "Edit" auf.
- Die Variablenzuordnung des Funktionsbausteins wird geöffnet
Verbinden Sie die Eingänge des Funktionsbausteins "Empfangen" im Strukturbaum mit den gleichen Variablen, 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

- Verbinden Sie die Ausgänge des Funktionsbausteins "Empfangen" im Strukturbaum mit den gleichen Variablen, 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

Daten	Beschreibung
Empfangsdaten	Im Register "Prozessvariablen" können beliebige Variablen angelegt werden. Die Offsets und Typen der Variablen müssen identisch mit den Offsets und den Typen der Variablen des Kommunikationspartners sein.

Bedienung des Funktionsbausteins „TCP_Receive“

Zur Bedienung des Funktionsbausteins "TCP_Receive" sind die folgenden Schritte erforderlich:

HINWEIS



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 Variablen des Kommunikationspartners sein.

- Setzen Sie im Anwenderprogramm am Eingang *A_ID* die Identifikationsnummer der TCP-Verbindung.
- Setzen Sie im Anwenderprogramm den Empfangs-Timeout am Eingang *A_Tmo*.
- Setzen Sie im Anwenderprogramm die erwartete Länge der zu empfangenden Variablen am Eingang *A_RLen*.

4. Setzen Sie im Anwenderprogramm den Eingang *A_Req* auf TRUE.

HINWEIS



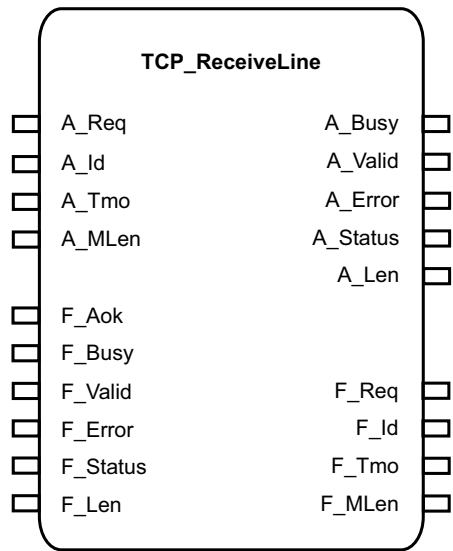
Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

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.

7.7.4 TCP_ReceiveLine



12757834891

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

HINWEIS



Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe auch Kapitel „Konfiguration der Funktionsbausteine“).

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

21233780 / DE – 07/2014

A-Eingänge	Beschreibung	Typ
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	A_MLen ist die maximale Länge einer zu empfangenden Zeile in Bytes. Die Empfangsvariablen müssen im Register Daten im Com-Funktionsbaustein angelegt werden. Übertragene Bytes = Min (A_MLen, Zeilenlänge, Länge des Datenbereichs).	INT

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	<ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten FALSE: Kein Fehler 	BOOL
A_Status	Am Ausgang A_Status werden Status und Fehlercode des Funktionsbausteins und der TCP-Verbindung ausgegeben.	DWORD
A_Len	Anzahl der empfangenen Bytes.	INT

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".

HINWEIS



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.

Verbinden Sie die F-Eingänge des Funktionsbausteins "TCP_ReveiveLine" im Anwenderprogramm mit den gleichen Variablen, 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

Verbinden Sie die F-Ausgänge des Funktionsbausteins "TCP_ReveiveLine" im Anwenderprogramm mit den gleichen Variablen, 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

Zugehörigen Funktionsbaustein „Zeilenweises Empfangen“ im Strukturbaumn erstellen

So erstellen Sie einen zugehörigen Funktionsbaustein "Zeilenweises Empfangen" im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [Send Receive over TCP] / [Funktionsbausteine] / [Neu] öffnen.
2. Wählen Sie den Funktionsbaustein "Zeilenweises Empfangen" aus. Markieren Sie den Funktionsbaustein "Zeilenweises Empfangen" und rufen Sie im Kontextmenü den Menüpunkt "Edit" auf.
3. Die Variablenzuordnung des Funktionsbausteins wird geöffnet

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

Eingänge	Typ
ID	INT
REQ	BOOL
MLEN	INT
TIMEOUT	TIME

4. Verbinden Sie die Ausgänge des Funktionsbausteins "Zeilenweises Empfangen" im Strukturbaum mit den gleichen Variablen, 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

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

Bedienung des Funktionsbausteins „TCP_ReceiveLine“

Zur Bedienung des Funktionsbausteins "TCP_ReceiveLine" sind die folgenden Schritte erforderlich:

HINWEIS



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 Sendevariablen des Kommunikationspartners sein.

1. Setzen Sie im Anwenderprogramm am Eingang *A_ID* die Identifikationsnummer der TCP-Verbindung.
2. Setzen Sie im Anwenderprogramm den Empfangs-Timeout am Eingang *A_Tmo*.
3. Setzen Sie im Anwenderprogramm die maximale Länge der zu empfangenden Zeile am Eingang *A_MLen*.

HINWEIS



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. Setzen Sie im Anwenderprogramm den Eingang *A_Req* auf TRUE.

HINWEIS



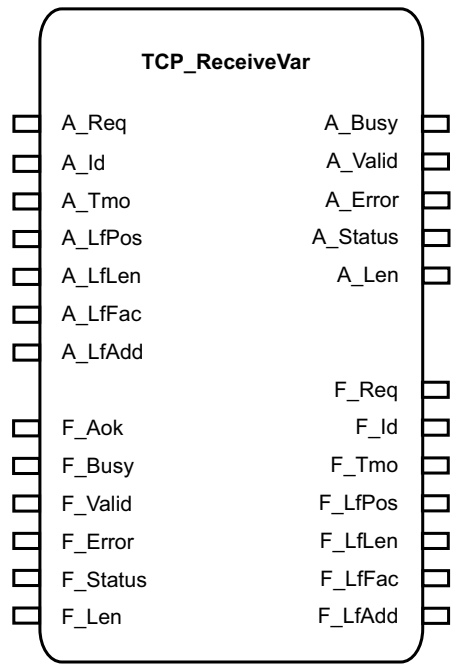
Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

Der Ausgang *A_Busy* ist TRUE, bis der Empfangspuffer voll ist oder das Zeilenende der LineFeed empfangen wurde, 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 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.

7.7.5 TCP_ReceiveVar



12757935755

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

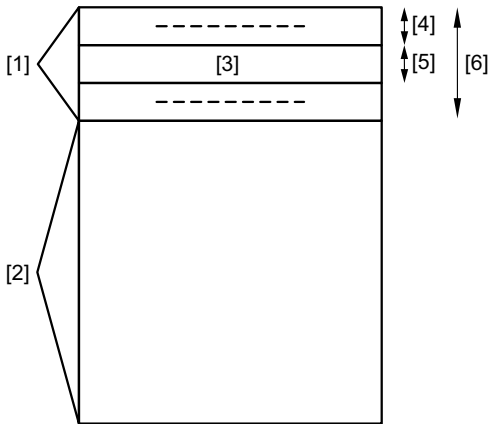


HINWEIS

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag-and-Drop aus der Bausteinbibliothek in das Anwenderprogramm (siehe auch Kapitel „Konfiguration der Funktionsbausteine“).

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*.



12757940619

- | | | | |
|-----|------------------|-----|---------|
| [1] | Kopfbereich | [4] | A_LfPos |
| [2] | Nutzdatenbereich | [5] | A_LfLen |
| [3] | Längenfeld | [6] | A_LfAdd |

Das empfangene Datenpaket besteht aus einem Kopf- und einem Nutzdatenbereich. Der Kopfbereich enthält Daten wie Teilnehmeradresse, 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).

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.	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_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 Kopffelds in Bytes.	USINT

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	<ul style="list-style-type: none"> TRUE: Bei Lesen trat Ein Fehler auf 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

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".



HINWEIS

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.

Verbinden Sie die F-Eingänge des Funktionsbausteins "TCP_ReveiveVar" im Anwenderprogramm mit den gleichen Variablen, 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

Verbinden Sie die F-Ausgänge des Funktionsbausteins "TCP_ReveiveVar" im Anwenderprogramm mit den gleichen Variablen, 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

Zugehörigen Funktionsbaustein „Variabel Empfangen“ im Strukturbaum erstellen

So erstellen Sie einen zugehörigen Funktionsbaustein "Variabel Empfangen" im Strukturbaum:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [Send Receive over TCP] / [Funktionsbausteine] / [Neu] öffnen.
2. Wählen Sie den Funktionsbaustein "Variabel Empfangen" aus. Markieren Sie den Funktionsbaustein "Variabel Empfangen" und rufen Sie im Kontextmenü den Menüpunkt "Edit" auf.
3. Die Variablenzuordnung des Funktionsbausteins wird geöffnet

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

Eingänge	Typ
ID	INT

Eingänge	Typ
LfAdd	USINT
LfFac	USINT
LfLen	USINT
LfPos	USINT
REQ	BOOL
TIMEOUT	TIME

4. Verbinden Sie die Ausgänge des Funktionsbausteins "Variabel Empfangen" im Strukturbaum mit den gleichen Variablen, 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

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

Bedienung des Funktionsbausteins "TCP_ReceiveVar"

Zur Bedienung des Funktionsbausteins "TCP_ReceiveVar" sind die folgenden Schritte erforderlich:

HINWEIS



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. Setzen Sie im Anwenderprogramm am Eingang *A_ID* die Identifikationsnummer der TCP-Verbindung.
2. Setzen Sie im Anwenderprogramm den Empfangs-Timeout am Eingang *A_Tmo*.
3. Setzen Sie im Anwenderprogramm die Parameter *A_LfPos*, *A_LfLen*, *A_LfFac* und *A_LfAdd*.
4. Setzen Sie im Anwenderprogramm den Eingang *A_Req* auf TRUE.

HINWEIS



Der Funktionsbaustein reagiert auf einen 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.

7.8 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.

So öffnen Sie das Control-Panel zur Überwachung des Send/Receive-Protokolls:

1. Im Strukturbaum [Ressource] wählen.
2. Aus der Action Bar "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.

7.8.1 Anzeigefeld allgemeine Parameter

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

Element	Beschreibung
Name	TCP SR Protokoll.
µP-Last (projektierte) in %	Siehe Kapitel "Lastbegrenzung".
µP-Last (tatsächliche) in %	Siehe Kapitel "Lastbegrenzung".
Ungestörte Verbindungen	Anzahl ungestörte Verbindungen.
Gestörte Verbindungen	Anzahl gestörte Verbindungen.

7.8.2 Anzeigefeld TCP-Verbindungen

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

Element	Beschreibung
Name	TCP-Verbindung.
Partner Timeout	<ul style="list-style-type: none"> • Ja: Partner Request Timeout abgelaufen • Nein: Partner Request Timeout nicht abgelaufen

Element	Beschreibung
Verbindungszustand	Aktueller Verbindungszustand dieser Verbindung. <ul style="list-style-type: none"> • 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 in ms	Ist die aktuelle Partner Request Timeout, innerhalb der nach Datenversand mindestens eine Datensendung vom Kommunikationspartner empfangen wurde.
Fehlercode	Siehe Kapitel "Fehlercode der TCP-Verbindung".
Zeitstempel Fehlercode in ms	Zeitstempel des letzten gemeldeten Fehlers. Wertebereich: Sekunden seit 1.1.1970 in Millisekunden
Empfangene Bytes	Anzahl empfangener Bytes dieser TCP-Verbindung.
Gesendete Bytes	Anzahl gesendeter Bytes dieser TCP-Verbindung.

7.8.3 Fehlercodes der TCP-Verbindung

Die Fehlercodes können aus der Variablen *Errorcode* gelesen werden. Pro konfigurierter Verbindung setzt sich der Verbindungsstatus 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.

Fehler-code (dezimal)	Fehlercode (hexadezi-mal)	Beschreibung
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 vom 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#36	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.

7.8.4 Zusätzliche Fehlercodetabelle der Funktionsbausteine

Die Fehlercodes der Funktionsbausteine werden nur an *A_Status* der Send & Receive-TCP-Funktionsbausteine ausgegeben.

Fehler-code (dezimal)	Fehlercode (Hexadezi-mal)	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

Fehler-code (dezimal)	Fehlercode (Hexadezi-mal)	Beschreibung
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.

7.8.5 Verbindungszustand

Fehlercode (hexadezi-mal)	Beschreibung
16#00	Verbindung OK.
16#01	Verbindung geschlossen.
16#02	Server wartet auf Verbindungsaufnahme.
16#04	Cleint versucht Verbindungsaufbau.
16#08	Verbindung ist blockiert.

7.8.6 Partner Verbindungszustand

Protokollzu-stand (dezimal)	Beschreibung
0	Keine Verbindung.
1	Verbindung OK.

8 Simple Network Time Protokoll (SNTP)

Mit dem SNTP-Protokoll (Simple Network Time Protocol) wird über Ethernet die Uhrzeit der SNTP-Clients durch den SNTP-Server synchronisiert. Die Sicherheitssteuerung kann als SNTP-Server und/oder als SNTP-Client konfiguriert und eingesetzt werden.

Es gilt der SNTP-Standard nach RFC 2030 (SNTP-Version 4) mit der Einschränkung, dass nur der Unicast-Modus unterstützt wird.

- Die Funktion ist standardgemäß freigeschaltet.
- Als Übertragungsstandard für das Netzwerk ist Ethernet 10/100/-BaseT erforderlich.

8.1 SNTP-Client

Der SNTP-Client benutzt zu seiner Zeitsynchronisation immer nur den erreichbaren SNTP-Server mit der höchsten Priorität.

In jeder Ressource kann ein SNTP-Client zur Zeitsynchronisation konfiguriert werden.

HINWEIS



Zeitsynchronisation einer Sicherheitssteuerung durch eine andere Sicherheitssteuerung.

Wird ein SNTP-Client auf einer Sicherheitssteuerung eingerichtet, so wird der interne SNTP-Server der Sicherheitssteuerung abgeschaltet.

8.1.1 Neuen SNTP-Client anlegen

So legen Sie einen neuen SNTP-Client an:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] öffnen.
2. Rechtsklick auf Protokolle und im Kontextmenü [Neu] / [SNTP-Client] wählen.
Ein neuer SNTP-Client wird hinzugefügt.
3. Im Kontextmenü von [SNTP-Client] / [Eigenschaften] das COM-Modul auswählen.

Das Dialogfenster des SNTP-Client enthält die folgenden Parameter:

Parameter	Beschreibung
Typ	SNTP-Client.
Name	Name für den SNTP-Client maximal 32 Zeichen.
Modul	Auswahl des CPU- oder COM-Moduls, auf dem dieses Protokoll abgearbeitet wird.
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). Initialdaten annehmen: Eingangsvariablen werden auf die Initialwerte zurückgesetzt. Letzten Wert beibehalten: Eingangsvariablen behalten den letzten Wert.

Parameter	Beschreibung
Max. µP-Budget aktivieren	Wird vom Betriebssystem des Moduls nicht berücksichtigt. Parameter wurde wegen der CRC- und Reload-Stabilität erhalten.
Max. µP-Budget in [%]	Wird vom Betriebssystem des Moduls nicht berücksichtigt. Parameter wurde wegen der CRC- und Reload-Stabilität erhalten.
Beschreibung	Beliebige eindeutige Beschreibung für den SNTP.
Aktuelle SNTP-Version	Anzeige der aktuellen SNTP Version.
Referenz Stratum	<p>Das Stratum eines SNTP-Clients gibt die Genauigkeit seiner lokalen Zeit wieder. Je niedriger das Stratum, desto genauer ist seine lokale Zeit. Null bedeutet ein unspezifiziertes oder nicht verfügbares Stratum (nicht gültig). Der aktuell verwendete SNTP-Server eines SNTP-Clients ist der, welcher erreichbar ist und die höchste Priorität besitzt.</p> <p>Ist das Stratum des aktuellen SNTP-Servers kleiner als das des SNTP-Clients, so übernimmt die Ressource die Zeit des aktuellen SNTP-Servers.</p> <p>Ist das Stratum des aktuellen SNTP-Servers größer als das des SNTP-Clients, so übernimmt die Ressource die Zeit des aktuellen SNTP-Servers nicht.</p> <p>Ist das Stratum des aktuellen SNTP-Servers gleich dem des SNTP-Clients, so sind zwei Fälle zu unterscheiden:</p> <ul style="list-style-type: none"> • Wenn der SNTP-Client (Ressource) ausschließlich als SNTP-Client arbeitet, so übernimmt die Ressource die Zeit des aktuellen SNTP-Servers. • Wenn der SNTP-Client (Ressource) gleichzeitig auch als SNTP-Server arbeitet, wird pro Anfrage des SNTP-Clients die Hälfte der Zeitdifferenz zum aktuellen SNTP-Server auf der Ressource übernommen (Zeit nähert sich langsam an). <p>Wertebereich: 16 s – 16384 s (Standardwert: 16 s)</p>
Client Zeitanfrage Intervall [s]	<p>Zeitintervall, in dem die Zeitsynchronisation durch den aktuellen SNTP-Server erfolgt. Das Client-Zeitanfrage-Intervall im SNTP-Client muss größer sein als das Timeout im SNTP-Server.</p> <p>Wertebereich: 16 s – 16384 s (Standardwert: 16 s)</p>

8.2 SNTP-Client (Server Info)

In der SNTP-Server Info wird die Verbindung zu einem SNTP-Server konfiguriert. Unterhalb eines SNTP-Clients können 1 bis 4 SNTP-Server Infos konfiguriert werden.

8.2.1 Neue SNTP-Server Info anlegen

So legen Sie eine neue SNTP-Server Info an:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] / [SNTP Client] öffnen.

2. Rechtsklick auf Protokolle und im Kontextmenü [Neu] / [SNTP-Server Info] wählen.
Eine neue SNTP-Server Info wird hinzugefügt.
3. Im Kontextmenü von [SNTP-Server Info] / [Eigenschaften] das COM-Modul auswählen.

Das Dialogfenster der SNTP-Server Info enthält die folgenden Parameter:

Parameter	Beschreibung
Typ	SNTP-Server Info.
Name	Name für die SNTP-Server Info. Maximal 31 Zeichen.
Beschreibung	Beschreibung für den SNTP-Server. Maximal 31 Zeichen.
IP-Adresse	IP-Adresse der Ressource oder des PCs, auf dem der SNTP-Server konfiguriert ist. Standardwert: 0.0.0.0
SNTP-ServerPriorität	Priorität mit welcher der SNTP-Client diesen SNTP-Server behandelt. Die für einen SNTP-Client konfigurierten SNTP-Server sollten unterschiedliche Prioritäten besitzen. Wertebereich: 0 (geringste Priorität) bis 4294967295 (höchste Priorität) Standardwert: 1s
SNTP-ServerTimeout [s]	Der Timeout im SNTP-Server muss kleiner eingestellt sein als das Zeitanfragenintervall im SNTP-Client. Wertebereich: 1 s – 16384 s Standardwert: 1 s

8.3 SNTP-Server

Der SNTP-Server nimmt die Anforderung von einem SNTP-Client entgegen und sendet seine aktuelle Zeit an den SNTP-Client zurück.

8.3.1 Neuen SNTP-Server anlegen

So legen Sie einen neuen SNTP-Server an:

1. Im Strukturbaum [Konfiguration] / [Ressource] / [Protokolle] öffnen.
2. Rechtsklick auf [Protokolle] und im Kontextmenü [Neu] / [SNTP-Server] wählen.
Ein neuer SNTP-Server wird hinzugefügt.
3. Im Kontextmenü von [SNTP Server] / [Eigenschaften] das COM-Modul auswählen.

Das Dialogfenster des SNTP-Client enthält die folgenden Parameter:

Parameter	Beschreibung
Typ	SNTP-Client.
Name	Name für den SNTP-Client maximal 31 Zeichen.
Modul	Auswahl des CPU- oder COM-Moduls, auf dem dieses Protokoll abgearbeitet wird.

Parameter	Beschreibung
Max. μ P-Budget aktivieren	<p>Aktiviert: Limit des μP-Budget aus dem Feld Max. μP-Budget in [%] übernehmen.</p> <p>Deaktiviert: Kein Limit des μP-Budget, für dieses Protokoll verwenden.</p>
Max. μ P-Budget in [%]	<p>Maximale μP-Last des Moduls, welche bei der Abarbeitung des Protokolls produziert werden darf.</p> <p>Wertebereich: 1 – 100 %</p> <p>Standardwert: 30 %</p>
Verhalten bei CPU/COM Verbindungsverlust	<p>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).</p> <p>Initialdaten annehmen: Eingangsvariablen werden auf die Initialwerte zurückgesetzt.</p> <p>Letzten Wert beibehalten: Eingangsvariablen behalten den letzten Wert.</p>
Beschreibung	Beliebige eindeutige Beschreibung für den SNTP.
Aktuelle SNTP-Version	Anzeige der aktuellen SNTP-Version.
Stratum des Zeitservers	<p>Das Stratum eines SNTP-Clients gibt die Genauigkeit seiner lokalen Zeit wieder. Je niedriger das Stratum, desto genauer ist seine lokale Zeit. Null bedeutet ein unspezifiziertes oder nicht verfügbares Stratum (nicht gültig).</p> <p>Das Stratum des SNTP-Servers muss niedriger oder gleich dem Stratum des anfragenden SNTP-Clients sein. Ansonsten wird die Zeit des SNTP-Servers vom SNTP-Client nicht übernommen.</p> <p>Wertebereich: 1 – 15</p> <p>Standardwert: 14</p>

9 Com-User Task (CUT)

Neben dem Anwenderprogramm, das mit SILworX® erstellt wird, kann zusätzlich ein C-Programm auf der Steuerung betrieben werden. Dieses nicht sichere C-Programm läuft als Com-User Task rückwirkungsfrei zum sicheren Prozessormodul auf dem Kommunikationsmodul der Steuerung.

Die Com-User Task hat einen eigenen Zyklus, der unabhängig vom Zyklus der CPU ist.

9.1 Eigenschaften der CUT

Die folgende Tabelle beschreibt die Eigenschaften der CUT

Element	Beschreibung
Com-User Task	Es kann für jede Sicherheitssteuerung eine Com-User Task konfiguriert werden.
Sicherheitsgerichtet	Nein

9.2 Voraussetzung

Um ein SILworX®-Programm mit einer Com-User Task zu erstellen, benötigen Sie Folgendes:

Bezeichnung	Sachnummer
SILworX® für MOVISAFE® HM31 <ul style="list-style-type: none">Hardware: SILworX® Lizenz DongleSoftware: SILworX® ab 4.64.0	19500114
Motion Library MOVISAFE® HM31 Bausteinbibliothek für Antriebssicherheitsfunktionen und sichere Wegmessung / Function block library for safety related position and velocity detection	17106400
Com-User Task MOVISAFE® HM31 Beachten Sie dazu das Handbuch „Com-User Task für MOVISAFE® HM31“.	28202430

- Software, die **nicht** im Lieferumfang enthalten ist:
Sie können diese Software zusammen mit der Dokumentation auf einem Datenträger (CD/DVD) von SEW-EURODRIVE unter folgenden Bestellangaben beziehen:
- Zur Diagnose der Com-User-Task-Anwendungen benötigen Sie als Software das MOVIVISION® Parameter- und Diagnosetool Version 2.0 (ebenfalls nicht im Lieferumfang enthalten).

10 Betriebssystem

Das Betriebssystem enthält alle Grundfunktionen der Sicherheitssteuerung

Welche Anwenderfunktionen das jeweilige PES ausführen soll, ist im Anwenderprogramm vorgegeben. Ein Codegenerator übersetzt das Anwenderprogramm in den Maschinencode. Das Programmierwerkzeug überträgt diesen Maschinencode in den Flash-Speicher der Steuerung.

10.1 Funktionen des Prozessor-Betriebssystems

Die wesentlichen Funktionen des Betriebssystems für das Prozessorsystem und die Verbindungen mit dem Anwenderprogramm sind in nachfolgender Tabelle aufgezeigt.

Funktionen des Betriebssystems	Verbindungen zum Anwenderprogramm
Zyklisches Abarbeiten des Anwenderprogramms.	Wirkt auf Variablen, Funktionsbausteine.
Konfiguration des Automatisierungsgeräts.	Festlegung durch Auswahl der Steuerung.
Prozessor-Tests.	-
Tests von E/A-Modulen.	-
Reaktionen im Fehlerfall.	Fest vorgegeben. Das Anwenderprogramm ist für Prozessreaktion verantwortlich.
Diagnose für Prozessorsystem und Ein-/Ausgänge.	Verwendung der Systemvariablen für Fehlermeldungen.
Sichere Kommunikation: • Peer-to-Peer Nicht sichere Kommunikation: • Modbus	Festlegung der Verwendung von Kommunikationsvariablen.
PADT-Schnittstelle: • Zulässige Aktionen	Festlegung im Programmierwerkzeug: • Konfiguration von Schutzfunktionen • Einloggen des Anwenders

Jedes Betriebssystem wird vom zuständigen TÜV geprüft und für den Betrieb mit der sicherheitsgerichteten Steuerung zugelassen. Die jeweils gültigen Versionen des Betriebssystems und die dazugehörigen Signaturen (CRCs) sind auf einer Liste dokumentiert, die SEW-EURODRIVE gemeinsam mit dem TÜV erstellt.

10.2 Verhalten bei Auftreten von Fehlern

Wichtig ist die Reaktion auf Fehler, die durch Tests festgestellt wurden. Zu unterscheiden sind folgende Arten von Fehlern.

- Permanente Fehler bei Eingängen und Ausgängen
- Vorübergehende Fehler bei Eingängen und Ausgängen
- Interne Fehler

10.2.1 Permanente Fehler bei Eingängen und Ausgängen

Ein Fehler, der in einem Eingangs- oder Ausgangskanal auftritt, beeinflusst die Steuerung nicht. Das Betriebssystem betrachtet nur den defekten Kanal als fehlerhaft und nicht die ganze Steuerung. Die übrigen Sicherheitsfunktionen werden davon nicht beeinflusst und bleiben aktiv.

Bei fehlerhaften Eingangskanälen gibt das Betriebssystem den sicheren Wert "0" oder den Initialwert an die Verarbeitung weiter.

Fehlerhafte Ausgangskanäle setzt das Betriebssystem in den energielosen Zustand. Ist es nicht möglich, nur einen Kanal abzuschalten, wird das ganze Ausgangsmodul als fehlerhaft betrachtet.

Das Betriebssystem setzt das Fehlerstatus-Signal und meldet dem Anwenderprogramm die Art des Fehlers.

Kann die Steuerung einen entsprechenden Ausgang nicht abschalten und wird auch der 2. Abschaltweg nicht wirksam, geht die Steuerung in STOPP. Der Watchdog des Prozessorsystems schaltet dann die Ausgänge ab.

Treten in den E/A-Modulen Fehler auf, die länger als 24 Stunden anstehen, schaltet die Steuerung nur die betroffenen E/A-Module permanent ab.

10.2.2 Vorübergehende Fehler bei Eingängen und Ausgängen

Tritt ein Fehler in einem Eingangs- oder Ausgangsmodul auf und verschwindet von selbst wieder, setzt das Betriebssystem den Fehlerstatus zurück und nimmt den normalen Betrieb wieder auf.

Das Betriebssystem wertet die Häufigkeit des Auftretens der Fehler statistisch aus. Es setzt den Status des Moduls ständig auf fehlerhaft, wenn die vorgegebene Fehlerhäufigkeit überschritten wird. Dadurch arbeitet das Modul auch nach Verschwinden des Fehlers nicht mehr. Die Freigabe des Moduls und die Löschung der Fehlerstatistik erfolgt mit dem Wechsel des Betriebszustandes der Steuerung von STOPP auf RUN. Diese Änderung quittiert den Fehler des Moduls.

10.2.3 Interne Fehler

HINWEIS



Sollte der seltene Fall auftreten, dass eine Sicherheitssteuerung einen internen Fehler feststellt, wird folgende Fehlerreaktion ausgeführt:

Die Sicherheitssteuerung läuft automatisch wieder hoch.

10.3 Das Prozessorsystem

Das Prozessorsystem ist die zentrale Komponente der Steuerung und kommuniziert mit den E/A-Modulen innerhalb der Steuerung über den E/A-Bus.

Das Prozessorsystem überwacht den Ablauf und die logisch korrekte Ausführung des Betriebssystems und des Anwenderprogramms. Folgende Funktionen werden zeitlich überwacht:

- Selbsttests für Hardware und Software des Prozessorsystems
- RUN-Zyklus des Prozessorsystems (einschließlich Anwenderprogramm)
- E/A-Tests und Verarbeitung der E/A-Signale

10.3.1 Betriebszustände des Prozessorsystems

LEDs auf der Frontplatte der Steuerung zeigen den Betriebszustand des Prozessorsystems an. Das Programmiergerät kann ihn ebenfalls anzeigen, zusammen mit anderen Parametern von Prozessormodul und Anwenderprogramm.

Der Stopp des Prozessors unterbricht die Ausführung des Anwenderprogramms und setzt die Ausgänge der Steuerung und aller Remote I/Os auf sichere Werte.

Das Setzen des Systemparameters NOTAUS auf TRUE mittels einer Programmlogik überführt das Prozessorsystem in den Zustand STOPP. Die wichtigsten Betriebszustände sind in der folgenden Tabelle zusammengefasst.

Betriebsart	Beschreibung
INIT	Sicherer Zustand des Prozessorsystems während der Initialisierung. Hardware- und Softwaretests werden durchgeführt.
STOPP/GÜLTIGE KONFIGURATION	Sicherer Zustand des Prozessorsystems ohne Ausführung eines Anwenderprogramms. Alle Ausgänge der Steuerung sind zurückgesetzt. Hardware- und Softwaretests werden durchgeführt.
STOPP/UNGÜLTIGE KONFIGURATION	Sicherer Zustand des Prozessorsystems ohne geladene Konfiguration oder nach einem Systemfehler. Alle Ausgänge der Steuerung sind zurückgesetzt, der Hardware-Watchdog wird nicht getriggert. Ein Reboot des Prozessorsystems kann nur über das PADT erfolgen.
RUN	Das Prozessorsystem ist aktiv: <ul style="list-style-type: none"> • Das Anwenderprogramm wird ausgeführt, E/A-Signale werden verarbeitet • Das Prozessorsystem führt sicherheitsgerichtete und nicht sicherheitsgerichtete Kommunikation durch (wenn parametrierung). • Hardware- und Softwaretests werden ausgeführt wie auch Tests für parametrierte E/A-Module.

10.3.2 Programmierung

Für die Programmierung der Steuerung dient ein PADT (Programmiergerät). Das Programmiergerät ist ein PC mit dem Programmierwerkzeug SILworX®.

SILworX® unterstützt folgende Programmiersprachen nach IEC 61131-3:

- Funktionsbausteinsprache (FBS)
- Ablaufsprache (AS)

Die Programmierwerkzeuge sind für die Erstellung sicherheitsgerichteter Programme und für die Bedienung der Steuerung geeignet. Weitere Informationen zu den Programmierwerkzeugen finden Sie in der Online-Hilfe zu SILworX®.

11 Anwenderprogramm

Die Erstellung des Anwenderprogramms für das PES und das Laden müssen mit einem Programmiergerät mit dem installierten Programmierwerkzeug SILworX nach den Erfordernissen der IEC 61131-3 erfolgen.

Zuerst ist mit dem Programmierwerkzeug das Anwenderprogramm zu erstellen und für den sicherheitsgerichteten Betrieb der Steuerung zu konfigurieren. Dabei sind die Vorgaben des Sicherheitshandbuchs "Dezentrale Sicherheitssteuerung MOVISAFE® HM31" zu beachten.

Nach dem anschließenden Kompilieren lädt das Programmiergerät Anwenderprogramm (Logik) und Konfiguration (Verbindungsparameter wie IP-Adresse, Subnet Mask und System-ID) in die Steuerung und startet diese.

Das Programmiergerät bietet folgende Möglichkeiten, während des Betriebs der Steuerung mit dieser zu arbeiten:

- Starten und Stoppen des Anwenderprogramms
- Anzeigen und Forcen von Variablen mit dem Force-Editor
- Im Testmodus Ausführen des Anwenderprogramms in Einzelschritten – nicht im sicherheitsgerichteten Betrieb
- Auslesen der Diagnosehistorie

Voraussetzung hierfür ist, dass das Programmiergerät dasselbe Anwenderprogramm enthält wie die Steuerung.

Für das Anwenderprogramm gibt es folgende optionale Funktionen:

- **Multitasking**

Multitasking bezeichnet die Fähigkeit der Sicherheitssteuerung, bis zu 32 Anwenderprogramme innerhalb des Prozessormoduls abzuarbeiten.

Dadurch lassen sich Teilfunktionen eines Projekts voneinander trennen. Die einzelnen Anwenderprogramme lassen sich unabhängig voneinander starten, stoppen und auch durch Reload laden.

- **Reload**

Wurden Änderungen an Anwenderprogrammen vorgenommen, dann können diese im laufenden Betrieb auf das PES übertragen werden. Das Betriebssystem prüft und aktiviert das geänderte Anwenderprogramm, das dann die Steuerungsaufgabe übernimmt.

HINWEIS



Die optionalen Funktionen können in MOVISAFE® HM31 ohne Aktivierung für 5000 Betriebsstunden zu Testzwecken verwendet werden. Bei der Verwendung der nicht aktivierten Funktionen leuchtet die System-LED "ERROR" dauerhaft rot.

Nach Ablauf der 5000 Betriebsstunden läuft die Steuerung nicht mehr an.

- Bestellen Sie rechtzeitig die Lizenzen zur Freischaltung der benötigten Funktionen.

11.1 Betriebsarten des Anwenderprogramms

In eine Steuerung kann nur jeweils ein Anwenderprogramm geladen werden. Für dieses Anwenderprogramm sind folgende Betriebsarten möglich:

Betriebsart	Beschreibung
RUN	Das Prozessorsystem ist in Betriebsart RUN. Das Anwenderprogramm wird zyklisch ausgeführt, E/A-Signale werden verarbeitet.
Testmodus (Einzelschritt)	Das Prozessorsystem ist in Betriebsart RUN. Das Anwenderprogramm wird auf manuelle Anforderung hin zyklusweise ausgeführt, E/A-Signale werden verarbeitet. Nicht zulässig für sicherheitsgerichteten Betrieb!
STOPP	Das Prozessorsystem ist in Betriebsart STOPP. Das Anwenderprogramm wird nicht (mehr) ausgeführt, die Ausgänge sind zurückgesetzt.
FEHLER	Ein geladenes Anwenderprogramm ist aufgrund eines Fehlers angehalten worden. Die Ausgänge sind zurückgesetzt. Hinweis: Ein Neustart des Programms ist nur durch das PADT möglich.

11.2 Multitasking

Multitasking bezeichnet die Fähigkeit der Sicherheitssteuerung, bis zu 32 Anwenderprogramme innerhalb des Prozessormoduls abzuarbeiten. Dadurch lassen sich Teilfunktionen eines Projekts voneinander trennen. Die einzelnen Anwenderprogramme lassen sich unabhängig voneinander starten, stoppen und auch durch Reload laden. SILworX® zeigt im Control-Panel die Zustände der einzelnen Anwenderprogramme an und ermöglicht deren Bedienung.

11.2.1 CPU-Zyklus ohne Multitasking

Der Zyklus des Prozessormoduls (CPU-Zyklus) für nur ein Anwenderprogramm besteht, vereinfacht dargestellt, aus folgenden Phasen:

1. Verarbeitung der Eingabedaten.
2. Abarbeitung des Anwenderprogramms.
3. Bereitstellung der Ausgabedaten.

Nicht dargestellt sind besondere Aufgaben (z. B. Reload), die gegebenenfalls innerhalb des CPU-Zyklus durchgeführt werden.

11.2.2 CPU-Zyklus mit Multitasking

Bei Multitasking ändert sich die zweite Phase, so dass ein CPU-Zyklus folgendermaßen abläuft:

1. Verarbeitung der Eingabedaten.
2. Abarbeitung aller Anwenderprogramme.

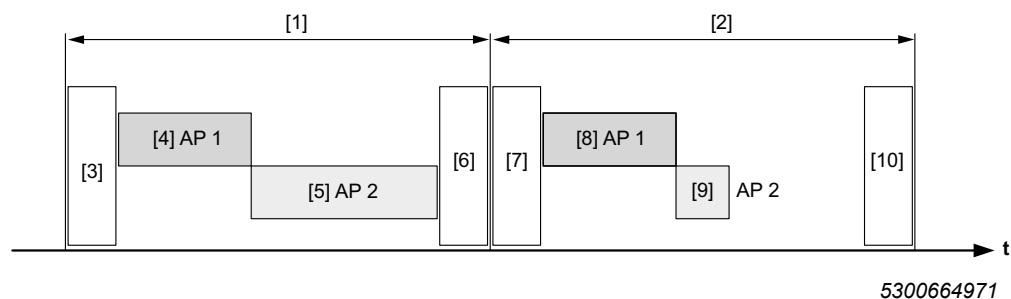
3. Bereitstellung der Ausgabedaten.

In der zweiten Phase kann die Sicherheitssteuerung bis zu 32 Anwenderprogramme abarbeiten. Dabei sind für jedes Anwenderprogramm zwei Fälle möglich:

- Innerhalb eines CPU-Zyklus wird ein vollständiger Zyklus des Anwenderprogramms abgearbeitet.
- Ein vollständiger Zyklus des Anwenderprogramms benötigt mehrere CPU-Zyklen zur Abarbeitung.

Diese beiden Fälle sind auch dann möglich, wenn es nur ein Anwenderprogramm gibt. Innerhalb eines CPU-Zyklus ist eine Übergabe von globalen Daten zwischen Anwenderprogrammen nicht möglich. Die von einem Anwenderprogramm geschriebenen Daten werden nach der vollständigen Ausführung des Anwenderprogramms unmittelbar vor Phase 3 verfügbar gemacht. Damit können diese Daten erst beim nächsten Start eines anderen Anwenderprogramms als Eingangswerte genutzt werden.

Das Beispiel im folgenden Bild zeigt beide Fälle in einem Projekt, das zwei Anwenderprogramme AP 1 und AP 2 enthält.



- [1] Erster betrachteter CPU-Zyklus
- [2] Zweiter betrachteter CPU-Zyklus
- [3] Eingangsverarbeitung im ersten CPU-Zyklus
- [4] Erster Teil des betrachteten Zyklus von Anwenderprogramm 1 (AP 1)
- [5] Erster Teil des betrachteten Zyklus von Anwenderprogramm 2 (AP 2)
- [6] Ausgangsverarbeitung im ersten CPU-Zyklus
- [7] Eingangsverarbeitung im zweiten CPU-Zyklus
- [8] Zweiter Teil des betrachteten Zyklus von Anwenderprogramm 1 (AP 1)
- [9] Zweiter Teil des betrachteten Zyklus von Anwenderprogramm 2 (AP 2)
- [10] Ausgangsverarbeitung im zweiten CPU-Zyklus

Jeder Zyklus des Anwenderprogramms AP 1 wird in jedem CPU-Zyklus vollständig abgearbeitet. AP 1 verarbeitet eine Eingabeänderung, die das System am Anfang des CPU-Zyklus [1] registriert hat, und liefert eine Reaktion am Ende dieses Zyklus.

Ein Zyklus des Anwenderprogramms AP 2 benötigt zu seiner Abarbeitung zwei CPU-Zyklen. AP 2 benötigt zur Verarbeitung einer Eingabeänderung, die das System am Anfang des ersten CPU-Zyklus [1] registriert hat, auch noch den zweiten CPU-Zyklus [2]. Aus diesem Grund steht die Reaktion auf diese Eingabeänderung erst am Ende des zweiten CPU-Zyklus [2] zur Verfügung. Die Reaktionszeit von AP 2 ist doppelt so groß wie die von AP 1.

Am Ende des ersten Teils [5] des betrachteten Zyklus von AP 2 wird die Abarbeitung von AP 2 **vollständig** unterbrochen und erst mit dem Beginn von [9] fortgesetzt. AP 2 verarbeitet während seines Zyklus die Daten, die das System zum Zeitpunkt [3] bereitgestellt hat. Die Ergebnisse von AP 2 stehen dem System zum Zeitpunkt [10] zur Verfügung (z. B. zur Ausgabe zum Prozess). Die Daten, die das Anwenderprogramm mit dem System austauscht, sind immer konsistent.

Die Verarbeitung der Programme ist durch eine Priorität steuerbar, die angibt, wie wichtig das jeweilige Anwenderprogramm im Verhältnis zu anderen ist (siehe Multitasking Mode 2).

Die Abarbeitung der Anwenderprogramme ist durch folgende Parameter bei Ressource und Programmen oder im Multitasking Editor festlegbar:

HINWEIS



Die Verwendung des Multitasking ist nur möglich mit einer Lizenzfreischaltung.

Parameter	Bedeutung	Einstellbar bei
Max. Dauer pro Zyklus [µs]	Zulässige Ausführungsdauer für ein Anwenderprogramm innerhalb eines CPU-Zyklus.	Anwenderprogramm, Multitasking Editor
Programm ID	ID für die Identifizierung des Programms bei der Anzeige in SILworX®	Anwenderprogramm, Multitasking Editor
Watchdog-Zeit	Watchdog-Zeit der Ressource	Ressource, Multitasking Editor
Sollzykluszeit [ms]	Gewünschte oder maximale Zykluszeit.	Ressource

Parameter	Bedeutung	Einstellbar bei
Multitasking Mode	<p>Verwendung der von Anwenderprogrammen nicht benötigten Ausführungsdauer, d. h. der Differenz zwischen der tatsächlichen Ausführungsdauer in einem CPU-Zyklus und der eingestellten <i>Max. Dauer pro Zyklus [μs]</i>.</p> <p>Mode 1: Die Länge eines Zyklus der CPU richtet sich nach der benötigten Ausführungsdauer aller Anwenderprogramme.</p> <p>Mode 2: Prozessor stellt von Anwenderprogrammen niedriger Priorität nicht benötigte Ausführungszeit den Anwenderprogrammen hoher Priorität zur Verfügung. Betriebsart für hohe Verfügbarkeit.</p> <p>Mode 3: Prozessor wartet nicht benötigte Ausführungszeit von Anwenderprogrammen ab und verlängert so den Zyklus.</p>	Ressource, Multitasking Editor
Sollzykluszeit-Modus	Verwendung der <i>Sollzykluszeit [ms]</i> .	Ressource
Priorität	Wichtigkeit eines Anwenderprogramms, höchste Priorität: 0.	Anwenderprogramm, Multitasking Editor
Maximale Zyklenanzahl	Maximale Anzahl CPU-Zyklen zur Abarbeitung eines Zyklus des Anwenderprogramms.	Multitasking Editor

Bei der Festlegung der Parameter sind folgende Regeln zu beachten:

- Ist die *Max. Dauer pro Zyklus [μs]* auf "0" gesetzt, so ist die Ausführungszeit des Anwenderprogramms nicht begrenzt, d. h. es wird immer vollständig ausgeführt. Daher darf die Anzahl Zyklen in diesem Fall nur "1" sein.
- Die Summe der Parameter *Max. Dauer pro Zyklus [μs]* aller Anwenderprogramme darf nicht größer als die Watchdog-Zeit der Ressource sein. Dabei ist auf eine ausreichende Reserve zur Bearbeitung der restlichen Aufgaben des Systems zu achten.

- Die Summe der Parameter *Max. Dauer pro Zyklus [μs]* aller Anwenderprogramme muss so groß sein, dass noch eine Reserve für die Einhaltung der Sollzykluszeit bleibt.
- Die *Programm-IDs* aller Anwenderprogramme müssen eindeutig sein.

SILworX® überwacht die Einhaltung dieser Regeln bei der Verifizierung und Codegenerierung. Bei der Online-Änderung von Parametern sind diese Regeln ebenfalls einzuhalten.

Aus diesen Parametern errechnet SILworX® die Watchdog-Zeit des Anwenderprogramms zu:

Watchdog-Zeit des Anwenderprogramms = *Watchdog-Zeit* × *Maximale Zyklenanzahl*

HINWEIS



Die Ablaufsteuerung zur Ausführung der Anwenderprogramme arbeitet in Schritten zu 250 μs. Aus diesem Grund können die parametrisierten Werte für *Max. Dauer pro Zyklus [μs]* um bis zu 250 μs über- oder unterschritten werden.

Die einzelnen Anwenderprogramme laufen generell rückwirkungsfrei voneinander ab. Gegenseitige Beeinflussung ist jedoch möglich durch:

- Verwendung derselben globalen Variablen in mehreren Anwenderprogrammen.
- Unvorhersehbar lange Laufzeiten bei einzelnen Anwenderprogrammen, falls keine parametrisierte Limitierung durch *Max Dauer pro Zyklus* erfolgt.

ACHTUNG



Gegenseitige Beeinflussung von Anwenderprogrammen möglich!

Verwendung derselben globalen Variablen in mehreren Anwenderprogrammen kann zu gegenseitiger Beeinflussung von Anwenderprogrammen mit unterschiedlichen Auswirkungen führen.

- Verwendung von globalen Variablen in mehreren Anwenderprogrammen genau planen.
- Querverweise in SILworX® nutzen, um die Verwendung globaler Daten zu prüfen. Globale Daten dürfen nur an einer Stelle mit Werten beschrieben werden, entweder in einem Anwenderprogramm oder von der Hardware.

HINWEIS



SEW-EURODRIVE empfiehlt, den Parameter *Max. Dauer pro Zyklus [μs]* auf einen geeigneten Wert ≠ 0 einzustellen. Dadurch wird das jeweilige Anwenderprogramm bei zu langer Laufzeit im aktuellen CPU-Zyklus beendet und im nächsten fortgesetzt, ohne die anderen Anwenderprogramme zu beeinträchtigen.

Andernfalls ist es möglich, dass eine ungewöhnlich lange Laufzeit eines oder mehrerer Anwenderprogramme zu einem Überschreiten der Sollzykluszeit oder gar der Watchdog-Zeit der Ressource und damit zum Fehlerstopp der Steuerung führt.

Das Betriebssystem legt die Ausführungsreihenfolge der Anwenderprogramme folgendermaßen fest:

- Das System arbeitet Anwenderprogramme mit niedrigerer Priorität vor Anwenderprogrammen mit höherer Priorität ab.
- Haben Anwenderprogramme die gleiche Priorität, arbeitet das System diese nach aufsteigenden *Programm IDs* ab.

Diese Reihenfolge gilt auch für das Starten und Stoppen der Anwenderprogramme beim Starten oder Stoppen des PES.

11.2.3 Multitasking-Mode

Für jede Ressource ist eine von drei Arbeitsweisen des Multitasking wählbar, die sich durch die Nutzung nicht benötigter Zeit der Ausführungsdauern pro CPU-Zyklus der Anwenderprogramme unterscheiden.

HINWEIS

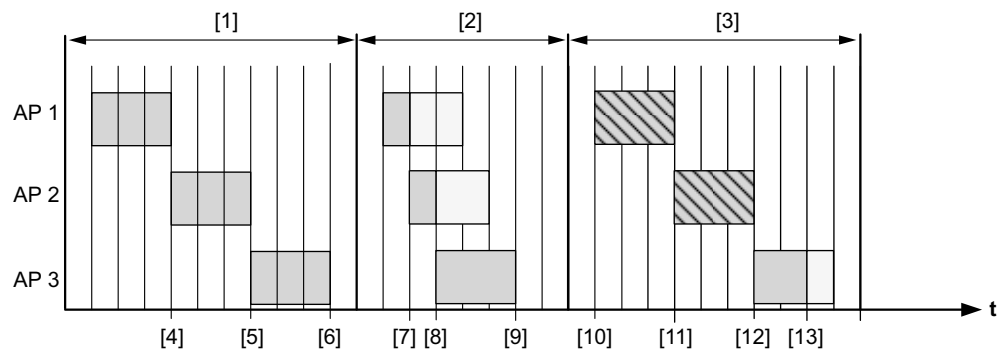


In den folgenden Beispielen für die Multitasking-Modes sind die Eingabe- und Ausgabeverarbeitung durch leere Bereiche am Anfang und Ende jedes CPU-Zyklus ange- deutet.

- Multitasking Mode 1

Multitasking Mode 1 nutzt die nicht benötigte Dauer zur Verringerung des CPU-Zyklus. Ist die Bearbeitung eines Anwenderprogramms abgeschlossen, wird sofort die Bearbeitung des nächsten Anwenderprogramms gestartet. Insgesamt ergibt sich dadurch ein kürzerer Zyklus.

Beispiel: 3 Anwenderprogramme AP 1, AP 2 und AP 3, bei denen ein Zyklus des Anwenderprogramms bis zu 3 CPU-Zyklen dauern darf.



5300668683

- [1] Erster betrachteter CPU-Zyklus
- [2] Zweiter betrachteter CPU-Zyklus
- [3] Dritter betrachteter CPU-Zyklus
- [4] *Max. Dauer pro Zyklus [μs]* von AP 1 abgelaufen, AP 2 startet
- [5] *Max. Dauer pro Zyklus [μs]* von AP 2 abgelaufen, AP 3 startet
- [6] *Max. Dauer pro Zyklus [μs]* von AP 3 abgelaufen, Ende des ersten CPU-Zyklus
- [7] Anwenderprogramm-Zyklus von AP 1 beendet, AP 2 wird fortgesetzt
- [8] Anwenderprogramm-Zyklus von AP 2 beendet, AP 3 wird fortgesetzt
- [9] *Max. Dauer pro Zyklus [μs]* von AP 3 abgelaufen, Ende des zweiten CPU-Zyklus
- [10] Nächster Anwenderprogramm-Zyklus von AP 1 beginnt
- [11] *Max. Dauer pro Zyklus [μs]* von AP 1 abgelaufen, nächster Anwenderprogramm-Zyklus von AP 2 beginnt
- [12] *Max. Dauer pro Zyklus [μs]* von AP 2 abgelaufen, AP 3 startet

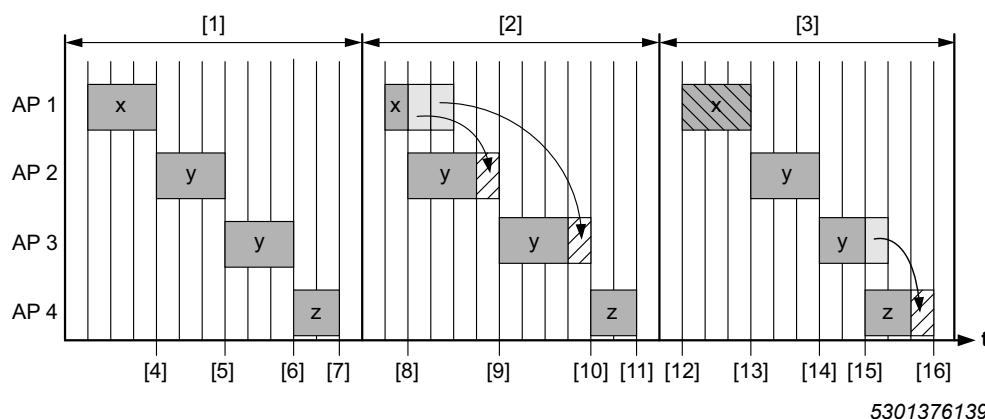
[13] Anwenderprogramm-Zyklus von AP 3 beendet

- Multitasking Mode 2

Multitasking Mode 2 verteilt nicht benötigte Dauer von Anwenderprogrammen niedriger Priorität auf Anwenderprogramme höherer Priorität. Dadurch stehen diesen außer ihrer eingestellten *Max. Dauer pro Zyklus* $[\mu s]$ noch die Anteile an der nicht benötigten Dauer zur Verfügung. Diese Arbeitsweise sorgt für hohe Verfügbarkeit.

Im folgenden Beispiel gibt es 4 Anwenderprogramme AP 1 – AP 4. Den Anwenderprogrammen sind folgende Prioritäten zugewiesen:

- AP 1 hat die niedrigste Priorität x
- AP 2 und AP 3 haben die mittlere Priorität y
- AP 4 hat die höchste Priorität z



[1] Erster betrachteter CPU-Zyklus.

[2] Zweiter betrachteter CPU-Zyklus.

[3] Dritter betrachteter CPU-Zyklus.

[4] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 1 abgelaufen, AP 2 startet.

[5] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 2 abgelaufen, AP 3 startet.

[6] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 3 abgelaufen, AP 4 startet.

[7] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 4 abgelaufen, Ende des ersten CPU-Zyklus.

[8] Anwenderprogramm-Zyklus von AP 1 beendet, AP 2 wird fortgesetzt. Restliche Dauer wird auf die *Max. Dauer pro Zyklus* $[\mu s]$ von AP 2 und AP 3 (höhere Priorität y) verteilt (Pfeile).

[9] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 2 plus anteilige Restdauer von AP 1 abgelaufen, AP 3 wird fortgesetzt.

[10] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 3 plus anteilige Restdauer von AP 1 abgelaufen, AP 4 startet.

[11] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 4 abgelaufen, Ende des zweiten CPU-Zyklus.

[12] Nächster Anwenderprogramm-Zyklus von AP 1 beginnt.

[13] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 1 abgelaufen, AP 2 wird fortgesetzt.

[14] *Max. Dauer pro Zyklus* $[\mu s]$ von AP 2 abgelaufen, AP 3 wird fortgesetzt.

[15] Anwenderprogramm-Zyklus von AP 3 abgelaufen, AP 4 wird fortgesetzt. Restliche Dauer wird AP 4 (höhere Priorität z) zugeschlagen.

[16] *Max. Dauer pro Zyklus [μs]* von AP 4 plus Restdauer von AP 3 abgelaufen, dritter Zyklus beendet.

HINWEIS



Die nicht verwendete Ausführungsdauer von Anwenderprogrammen, die nicht ausgeführt werden, steht nicht als Restzeit für andere Anwenderprogramme zur Verfügung.

Anwenderprogramme werden nicht ausgeführt, wenn sie sich in einem der folgenden Zustände befinden:

- STOPP
- ERROR
- TEST_MODE

Dies kann dazu führen, dass sich die Anzahl der CPU-Zyklen erhöht, die zur Abarbeitung des Zyklus eines anderen Anwenderprogramms benötigt werden.

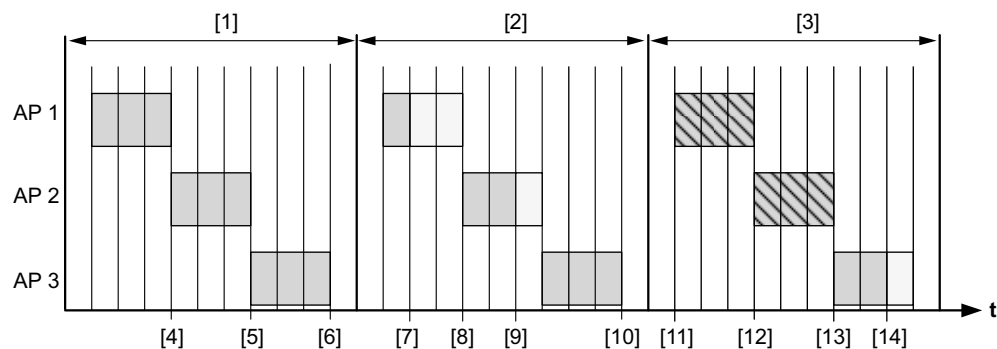
In diesem Fall kann zu niedrige Parametrierung der Maximalen Zyklusanzahl zur Überschreitung der maximalen Verarbeitungsdauer des Anwenderprogramms und zum Fehlerstopp führen!

Maximale Verarbeitungsdauer = *Max. Dauer pro Zyklus [μs]* × Maximale Zyklusanzahl-
Zur Prüfung der Parametrierung Multitasking Mode 3 verwenden!

- Multitasking Mode 3

Multitasking Mode 3 nutzt die nicht benötigte Dauer nicht für die Ausführung von Anwenderprogrammen, sondern wartet die Zeit bis zum Erreichen der *Max. Dauer pro Zyklus [μs]* des Anwenderprogramms und startet die Bearbeitung des nächsten Anwenderprogramms. Dieses Verhalten führt zu gleicher Dauer der CPU-Zyklen. Der Multitasking Mode 3 ist dazu gedacht, dass der Anwender prüfen kann, ob der Multitasking Mode 2 auch im ungünstigsten Fall eine ordnungsgemäße Programmausführung gewährleisten kann.

Beispiel:



5301379595

- [1] Erster betrachteter CPU-Zyklus
- [2] Zweiter betrachteter CPU-Zyklus
- [3] Dritter betrachteter CPU-Zyklus
- [4] *Max. Dauer pro Zyklus [μs]* von AP 1 abgelaufen, AP 2 startet
- [5] *Max. Dauer pro Zyklus [μs]* von AP 2 abgelaufen, AP 3 startet
- [6] *Max. Dauer pro Zyklus [μs]* von AP 3 abgelaufen, Ende des ersten CPU-Zyklus. Anwenderprogramm 1 (AP 1) wird fortgesetzt

- [7] Anwenderprogramm-Zyklus von AP 1 beendet, AP 2 wird fortgesetzt. Restliche Dauer wird gestartet.
- [8] *Max. Dauer pro Zyklus [μs]* von AP 1 abgelaufen, AP 2 wird fortgesetzt
- [9] Anwenderprogramm-Zyklus von AP 2 beendet. Restliche Dauer wird gewartet.
- [10] *Max. Dauer pro Zyklus [μs]* von AP 3 beendet, zweiter CPU-Zyklus beendet.
- [11] Nächster Anwenderprogramm-Zyklus von AP 1 beginnt
- [12] Nächster *Max. Dauer pro Zyklus [μs]* von AP 2 abgelaufen, nächster Anwenderprogramm-Zyklus von AP 2 startet.
- [13] *Max. Dauer pro Zyklus [μs]* von AP 2 abgelaufen, AP 3 wird fortgesetzt.
- [14] Anwenderprogramm-Zyklus von AP 3 beendet, Wartezeit bis zum Ende von *Max. Dauer pro Zyklus [μs]* von AP 3. Dritter CPU-Zyklus beendet.

11.3 Reload

Wurden Änderungen an Anwenderprogrammen vorgenommen, dann können diese im laufenden Betrieb auf das PES übertragen werden. Das Betriebssystem prüft und aktiviert das geänderte Anwenderprogramm, das dann die Steuerungsaufgabe übernimmt.

HINWEIS



Beim Reload von Schrittketten zu beachten:

Die Reload-Information für Schrittketten berücksichtigt nicht den aktuellen Status der Kette. Daher ist es möglich, durch Reload einer entsprechenden Änderung der Schrittkette diese in einen undefinierten Zustand zu versetzen. Die Verantwortung hierfür liegt beim Anwender.

- Löschen des aktiven Schritts. Danach hat kein Schritt der Schrittkette den Zustand *aktiv*.
- Umbenennen des Initialschritts, während ein anderer Schritt aktiv ist. Dies führt zu einer Schrittkette mit zwei aktiven Schritten!

HINWEIS



Beim Reload von Actions zu beachten:

Reload lädt Actions mit ihren kompletten Daten. Die Konsequenzen daraus sind vor dem Reload sorgfältig zu überdenken.

- Entfernen eines Timer-Bestimmungszeichens durch den Reload führt dazu, dass der Timer sofort abgelaufen ist. Dadurch kann der Ausgang Q in Abhängigkeit von der restlichen Belegung auf TRUE gehen.
- Entfernen des Bestimmungszeichens bei haftenden Elementen (z. B. Bestimmungszeichen S), die gesetzt waren, führt dazu, dass die Elemente gesetzt bleiben.
- Entfernen eines Bestimmungszeichens P0, das TRUE gesetzt war, löst den Trigger aus.

Vor der Ausführung eines Reload prüft das Betriebssystem, ob die notwendigen Zusatzaufgaben die Zykluszeit der laufenden Anwenderprogramme so stark erhöhen würde, dass die festgelegte Watchdog-Zeit überschritten würde. In diesem Fall wird der Reload mit einer Fehlermeldung abgebrochen, und die Steuerung läuft mit der bisherigen Projektkonfiguration weiter.

HINWEIS



Die Steuerung kann Reload abbrechen.

Um einen erfolgreichen Reload zu erreichen, ist bei der Festlegung der Watchdog-Zeit eine Reserve für den Reload einzuplanen oder die Watchdog-Zeit der Steuerung vorübergehend um eine Reserve erhöhen.

Die vorübergehende Erhöhung der Watchdog-Zeit ist mit der zuständigen Prüfstelle abzustimmen. Eine Überschreitung der Sollzykluszeit kann ebenfalls zum Abbruch eines Reload führen.

Reload ist nur möglich, wenn der Systemparameter *Reload erlaubt* auf "ON" und die Systemvariable *Reload-Deaktivierung* auf "OFF" eingestellt ist.

HINWEIS



Es liegt in der Verantwortung des Anwenders, bei der Bemessung der Watchdog-Zeit Reserven einzuplanen. Diese sollen die folgenden Situationen beherrschbar machen.

- Schwankungen bei der Zykluszeit des Anwenderprogramms
- Plötzliche, starke Belastungen des Zyklus, z. B. durch Kommunikation
- Ablauf von Zeitgrenzen bei der Kommunikation

Die globalen und lokalen Variablen erhalten beim Reload jeweils die Werte der gleichnamigen Variablen des vorhergehenden Projektstands. Namen lokaler Variable erhalten den Instanznamen der Programm-Organisations-Einheit (POE, gemäß IEC 61131).

Diese Vorgehensweise hat folgende Auswirkungen, wenn Namen geändert und mittels Reload auf das PES geladen werden:

- Umbenennen einer Variablen wirkt wie löschen und neu einfügen, d. h. führt zum Initialisieren, auch bei Retain-Variablen. Dadurch verlieren diese ihren aktuellen Wert.
- Umbenennen einer Funktionsbaustein-Instanz führt zum Initialisieren aller Variablen, auch der Retain-Variablen und aller enthaltenen Funktionsbaustein-Instanzen.
- Umbenennen eines Programms führt zum Initialisieren aller enthaltenen Variablen und Funktionsbaustein-Instanzen.

Dieses Verhalten kann unbeabsichtigte Auswirkungen auf ein oder mehrere Anwenderprogramme und damit auf die zu steuernde Anlage haben!

11.3.1 Bedingungen für die Verwendung von Reload

Die Verwendung von Reload ist nur mit einer Lizenz möglich.

Die folgenden Projektänderungen sind mit Reload in die Steuerung übertragbar:

- Änderungen an den Parametern des Anwenderprogramms
- Änderungen an der Logik in Programmen, Funktionsbausteinen, Funktionen
- Änderungen, bei denen gemäß folgender Tabelle Reload möglich ist

Änderungen bei	Art der Änderung			
	Hinzu- fügen	Löschen	Initialwert ändern	Andere Vari- able zuwei- sen
Zuweisungen globaler Variablen zu:				
• Anwenderprogrammen	X	X	X	X
• Systemvariablen	X	X	X	X
• E/A-Kanälen	X	X	X	X
• Kommunikationsprotokollen	-	-	-	-
• safeethernet	-	-	X	-
• SER	-	-		
Kommunikationsprotokollen	-	-	n. a.	n. a.
Anwenderprogrammen	X	X ¹⁾	n. a.	n. a.
System-ID, Rack-ID	-			
IP-Adressen	-			
Benutzerkonten und Lizenzen	X			

1) Reload möglich, aber in der Steuerung muss mindestens ein Anwenderprogramm verbleiben

X Reload möglich

- Reload nicht möglich

n. a. Nicht anwendbar

Reload ist nur nach Änderungen gemäß obigen Bedingungen möglich, andernfalls die Steuerung stoppen und Download verwenden.

HINWEIS



Auf folgende Weise lässt sich Reload in Fällen ermöglichen, in denen Zuweisungen globaler Variablen hinzugefügt werden:

Auf diese Weise ist es später nur notwendig, diese Zuweisungen zu ändern und nicht hinzuzufügen, so dass ein Reload möglich ist.

- Bereits beim Erstellen des Anwenderprogramms Kommunikationsprotokollen unbenutzte globale Variablen zuweisen.
- Den unbenutzten globalen Variablen einen sicheren Wert als Initialwert zuweisen.

Auf diese Weise ist es später nur notwendig, diese Zuweisungen zu ändern und nicht hinzuzufügen, so dass ein Reload möglich ist.

11.4 Allgemeines zum Forcen

Forcen bedeutet das Ersetzen des aktuellen Wertes einer Variablen durch einen Force-Wert. Eine Variable kann ihren aktuellen Wert aus folgenden Quellen erhalten:

- Durch einen physikalischen Eingang
- Durch die Kommunikation
- Durch eine logische Verknüpfung

Beim Forcen einer Variablen gibt der Anwender den Wert vor. Das Forcen wird in folgenden Fällen angewendet:

- Testen des Anwenderprogramms, besonders in Fällen, die selten auftreten und auf andere Weise nicht geprüft werden können.
- Simulation nicht verfügbarer Sensoren in Fällen, in denen der Initialwert nicht angemessen ist.

⚠ WARNUNG



Personenschäden durch geforcte Werte möglich!

Tod oder schwere Körperverletzung möglich.

- Werte nur nach Absprache mit der Prüfstelle für die Anlagenabnahme forcen.
- Einschränkungen des Forcens nur nach Absprache mit der Prüfstelle für die Anlagenabnahme aufheben.

Während des Forcens muss der Verantwortliche die sicherheitstechnisch ausreichende Überwachung des Prozesses durch andere technische und organisatorische Maßnahmen gewährleisten. SEW-EURODRIVE empfiehlt, das Forcen zeitlich zu begrenzen.

⚠ WARNUNG



Störung des sicherheitsgerichteten Betriebs durch geforcte Werte möglich!

Tod oder schwere Körperverletzung möglich.

- Geforcte Werte können zu falschen Ausgangswerten führen.
- Forcen verlängert die Zykluszeit. Dadurch kann die Watchdog-Zeit überschritten werden.

Grundlegende Informationen zum Forcen werden im Dokument „Maintenance Override“ des TÜV gegeben. Das Dokument ist auf folgender Homepage des TÜV bereitgestellt:

<http://www.tuv-fs.com> oder <http://www.tuvasi.com>.

11.5 Forcen

Forcen kann auf zwei Ebenen erfolgen:

- Globales Forcen
Globale Variablen werden für alle Verwendungen geforct.
- Lokales Forcen

Die Werte von lokalen Variablen eines Anwenderprogramms werden geforct.

Damit eine globale oder lokale Variable geforct wird, müssen folgende Bedingungen erfüllt sein:

- Der zugehörige Force-Schalter ist gesetzt
- Das Forcen wurde gestartet

Ist das Forcen gestartet, wirkt eine Änderung des Force-Schalters sofort. Ist das Forcen gestartet und der Force-Schalter gesetzt, wirkt eine Änderung des Force-Werts sofort. Das lokale Forcen lässt sich für jedes Anwenderprogramm getrennt starten und stoppen.

11.5.1 Zeitbegrenzung

Für das globale wie für das lokale Forcen sind unterschiedliche Zeitbegrenzungen einstellbar. Nach Ablauf der eingestellten Zeit beendet die Steuerung das Forcen. Das Verhalten der Sicherheitssteuerung nach dem Ablauf der Zeitbegrenzung ist einstellbar.

- Beim globalen Forcen sind folgende Einstellungen wählbar:
 - Die Ressource stoppt.
 - Die Ressource läuft weiter.
- Beim lokalen Forcen sind folgende Einstellungen wählbar:
 - Das Anwenderprogramm stoppt.
 - Das Anwenderprogramm läuft weiter.

Es ist auch möglich, ohne Zeitbegrenzung zu forcen. In diesem Fall ist das Forcen von Hand zu beenden. Nach dem Ende des Forcens einer Variablen gilt wieder der Prozesswert.

11.5.2 Force-Editor

Der Force-Editor von SILworX® zeigt alle Variablen an, für die Forcen möglich ist. Dabei werden die globalen und lokalen Variablen getrennt in unterschiedlichen Registern angezeigt. In den Registern ist das Einstellen von Force-Werten und Setzen von Force-Schaltern möglich.

11.5.3 Einschränkung des Forcens

Um eventuelle Störungen des sicherheitsgerichteten Betriebs durch unzulässiges Forcen zu vermeiden, können in der Konfiguration folgende Maßnahmen getroffen werden, die die Benutzung des Forcens einschränken.

- Einrichtung unterschiedlicher Benutzerkonten mit und ohne Erlaubnis zum Forcen
- Verbieten des globalen Forcens für eine Ressource
- Verbieten des lokalen Forcens, bzw. der Prozesswert-Eingabe
- Zusätzlich kann das Forcen per Schlüsselschalter unmittelbar abgeschaltet werden. Hierzu muss die Systemvariable *Force-Deaktivierung* mit einem digitalen Eingang verbunden sein, an den ein Schlüsselschalter angeschlossen ist.

Die Systemvariable *Force-Deaktivierung* verhindert, dass das Forcen für globale und lokale Variable gestartet wird und schaltet bereits gestartetes Forcen unmittelbar ab.

12 Inbetriebnahme

Die Inbetriebnahme der Sicherheitssteuerung besteht aus folgenden Phasen:

- Mechanische Installation. Beachten Sie dazu das Kapitel "Mechanische Installation" in der Betriebsanleitung "Dezentrale Sicherheitssteuerung MOVISAFE® HM31".
- Elektrische Installation. Beachten Sie dazu das Kapitel "Elektrische Installation" in der Betriebsanleitung "Dezentrale Sicherheitssteuerung MOVISAFE® HM31".
- Konfiguration
 - Erstellung des Anwenderprogramms
 - Festlegung von Sicherheits-, Kommunikations- und anderen Parametern

12.1 Checkliste zur Projektierung, Programmierung und Inbetriebnahme

Diese Checkliste ist eine Empfehlung für den Anwender

- zur Projektierung, Programmierung und Inbetriebnahme von sicherheitsgerichteten Ein- und Ausgängen
- zur Erstellung eines Anwenderprogramms mit dem Programmierwerkzeug SILworX®

Durch das Ausfüllen der Checkliste kann sichergestellt werden, dass die Anforderungen vollständig und übersichtlich erfasst sind. Die Checkliste ist auch eine Dokumentation über die Verbindung zwischen externer Verdrahtung und Anwenderprogramm.

Die Checkliste *PFF_HM31A_Checkliste_DE.pdf* kann als PDF-Dokument auf der SEW-Homepage (www.sew-eurodrive.de) unter der Rubrik "Dokumentationen" im Bereich "safetyDRIVE" heruntergeladen werden.

12.2 Konfiguration mit SILworX®

Der Hardware-Editor des Programmierwerkzeugs SILworX® zeigt MOVISAFE® HM31 ähnlich einem Basisträger, bestückt mit folgenden Modulen:

- Prozessormodul (CPU)
- Kommunikationsmodul (COM)
- Digitales Eingangsmodul (DI 26)
- Digitales Ausgangsmodul (DO 8)
- Zählermodul (HSC 2)

Durch Doppelklicken auf die Module öffnet sich die Detailansicht mit Registern. In den Registern können die im Anwenderprogramm konfigurierten globalen Variablen den Systemvariablen des jeweiligen Moduls zugeordnet werden.

12.2.1 Prozessormodul

Durch Doppelklicken auf die Module öffnet sich die Detailansicht mit Registern. In den Registern können die im Anwenderprogramm konfigurierten globalen Variablen den Systemvariablen des jeweiligen Moduls zugeordnet werden.

Register Modul

Das Register Modul enthält die folgenden Parameter.

Parameter	Beschreibung
Name	Name des Moduls.
Max. μ P-Budget für HH-Protokoll verwenden	<ul style="list-style-type: none"> Aktiviert: Limit der CPU-Last aus dem Feld <i>Max. μP-Budget für HH-Protokoll [%]</i> übernehmen. Deaktiviert: Kein Limit der CPU-Last für safeethernet verwenden. Standardeinstellung: Deaktiviert
Max. μ P-Budget für HH-Protokoll [%]	Maximale CPU-Last des Moduls, welche bei der Abarbeitung des safeethernet-Protokolls produziert werden darf. Hinweis: Die maximale Last muss unter allen verwendeten Protokollen aufgeteilt werden, die dieses Kommunikationsmodul benutzen.
IP Adresse	IP-Adresse der Ethernet-Schnittstelle. Standardwert: 192.168.0.99
Subnet Mask	32-Bit-Adressmaske zur Unterteilung einer IP-Adresse in Netzwerk- und Host-Adresse. Standardwert: 255.255.252.0
Standard-Schnittstelle	Aktiviert: Schnittstelle wird als Standardschnittstelle für ein System-Login verwendet. Standardeinstellung: Deaktiviert
Default-Gateway	IP-Adresse des Default Gateway. Standardwert: 0.0.0.0

Parameter	Beschreibung
ARP Aging Time [s]	<p>Ein CPU- oder COM-Modul speichert die MAC-Adressen seiner Kommunikationspartner in einer MAC-/IP-Adresse Zuordnungstabelle (ARP-Cache).</p> <p>Wenn während einer Zeitspanne von $1 \times \dots 2 \times \text{ARP Aging Time}$</p> <ul style="list-style-type: none"> Nachrichten vom Kommunikationspartner eintreffen, bleibt die MAC-Adresse im ARP-Cache erhalten. Keine Nachrichten vom Kommunikationspartner eintreffen, wird die MAC-Adresse aus dem ARP-Cache gelöscht. <p>Der typische Wert für die <i>ARP Aging Time</i> in einem lokalen Netzwerk ist 5 s – 300 s.</p> <p>Der Inhalt des ARP-Cache kann vom Anwender nicht ausgelesen werden.</p> <p>Bei der Verwendung von Routern oder Gateways <i>ARP Aging Time</i> an die zusätzlichen Verzögerungen für Hin- und Rückweg anpassen (erhöhen). Bei zu geringer <i>ARP Aging Time</i> löscht das CPU-/COM-Modul die MAC-Adresse des Kommunikationspartners aus dem ARP-Cache und die Kommunikation wird nur verzögert ausgeführt oder bricht ab. Für einen effizienten Einsatz muss die <i>ARP Aging Time</i> größer als die <i>ReceiveTimeouts</i> der verwendeten Protokolle sein.</p> <p>Wertebereich: 1 s – 3600 s</p> <p>Standardwert: 60 s</p>
MAC Learning	<p>Lernverhalten des ARP-Cache:</p> <ul style="list-style-type: none"> konservativ: MAC-Adressen gespeicherter ARP-Einträge werden durch empfangene Meldungen nicht überschrieben. tolerant: MAC-Adressen gespeicherter ARP-Einträge werden durch empfangene Meldungen überschrieben. <p>Standardeinstellung: konservativ</p>
IP Forwarding	<p>Ermöglicht einem Prozessormodul, als Router zu arbeiten und Datenpakete anderer Netzwerkknoten weiterzuleiten.</p> <p>Standardeinstellung: Deaktiviert</p>
ICMP Mode	<p>Meldungstypen des Internet Control Message Protocol (ICMP), die vom Prozessormodul unterstützt werden:</p> <ul style="list-style-type: none"> Keine ICMP-Antworten Echo Response Host unerreichbar Alle implementierten ICMP-Antworten <p>Standardeinstellung: Echo Response</p>
Max. Kom. Zeitscheibe ASYNC [ms]	<p>Höchstwert in ms der Zeitscheibe, die innerhalb des Zyklus der Ressource für Kommunikation verwendet wird.</p> <p>Einstellbereich: 2 – 5000 ms</p>
Max. Dauer Konfigurationsverbindungen [ms]	<p>Definiert, wie viel Zeit innerhalb eines CPU-Zyklus für die Prozessdaten-Kommunikation zur Verfügung steht.</p> <p>Einstellbereich: 6 – 5000 ms</p>

Parameter	Beschreibung
Sollzykluszeit [ms]	<p>Gewünschte oder maximale Zykluszeit, siehe <i>Sollzykluszeit-Modus</i>. Die <i>Sollzykluszeit</i> darf höchstens so groß sein wie die eingestellte Watchdog-Zeit (6 ms), andernfalls lehnt das PES sie ab.</p> <p>Einstellbereich: 0 – 7500 ms</p>
Sollzykluszeit-Modus	<p>Verwendung der <i>Sollzykluszeit</i> [ms].</p> <p>fest:</p> <p>Das PES hält die <i>Sollzykluszeit</i> ein und verlängert den Zyklus, falls nötig. Dies gilt nicht, falls die Abarbeitungszeit der Anwenderprogramme die <i>Sollzykluszeit</i> überschreitet.</p> <p>fest-tolerant:</p> <p>Wie bei fest, aber beim 1. Aktivierungszyklus der Reload-Funktion (Funktion ist lizenziert) findet die <i>Sollzykluszeit</i> keine Beachtung.</p> <p>dynamisch-tolerant:</p> <p>Das PES hält möglichst die <i>Sollzykluszeit</i> ein, führt aber den Zyklus in möglichst kurzer Zeit aus. Beim 1. Aktivierungszyklus der Reload-Funktion (Funktion ist lizenziert) findet die <i>Sollzykluszeit</i> keine Beachtung.</p>
Maximale Systembus-Latenzzeit [µs]	Für die Sicherheitssteuerung MOVISAFE® HM31 nicht anwendbar!
safeethernet-CRC	<p>Aktuelle Version:</p> <p>Die Bildung des CRC für safeethernet erfolgt mit dem aktuellen Algorithmus.</p>

Register Routings

Das Register Routings enthält die folgenden Parameter.

Parameter	Beschreibung
Name	Bezeichnung der Routing-Einstellung.
IP Adresse	<p>Ziel IP-Adresse des Kommunikationspartners (bei direktem Host-Routing) oder Netzwerkadresse (bei Subnet Routing).</p> <p>Wertebereich: 0.0.0.0 – 255.255.255.255</p> <p>Standardwert: 0.0.0.0</p>
Subnet Mask	<p>Definiert Ziel-Adressbereich für einen Routing-Eintrag. 255.255.255.255 (bei direktem Host-Routing) oder Subnetzmaske des adressierten Subnetzes.</p> <p>Wertebereich: 0.0.0.0 – 255.255.255.255</p> <p>Standardwert: 255.255.255.255</p>
Gateway	<p>IP-Adresse des Gateways zum adressierten Netzwerk.</p> <p>Wertebereich: 0.0.0.0 – 255.255.255.255</p> <p>Standardwert: 0.0.0.1</p>

Register Ethernet-Switch

Das Register Ethernet-Switch enthält die folgenden Parameter.

Parameter	Beschreibung
Name	Name des Ports (Eth1 – Eth4) wie Gehäuseaufdruck; pro Port darf nur eine Konfiguration vorhanden sein.
Speed [Mbit/s]	10 Mbit/s: Datenrate 10 Mbit/s 100 Mbit/s: Datenrate 100 Mbit/s 1000 Mbit/s: Datenrate 1000 Mbit/s (wird nicht unterstützt) Autoneg: Automatische Einstellung der Baudrate Standardwert: Autoneg
Flow-Control	Vollduplex: Kommunikation in beide Richtungen gleichzeitig Halbduplex: Kommunikation in eine Richtung Autoneg: Automatische Kommunikationssteuerung Standardwert: Autoneg
Autoneg auch bei festen Werten	Das <i>Advertising</i> (Übermitteln der Speed und Flow-Control-Eigenschaften) wird auch bei fest eingestellten Werten von <i>Speed</i> und <i>Flow-Control</i> durchgeführt. Hierdurch erkennen andere Geräte, deren Ports auf <i>Autoneg</i> eingestellt sind, die Einstellung der Ports der Sicherheitssteuerung.
Limit	Eingehende Multicast- und/oder Broadcast-Pakete limitieren. Aus: keine Limitierung Broadcast: Broadcast limitieren (128 kbit/s) Multicast und Broadcast: Multicast und Broadcast limitieren (1024 kbit/s) Standardwert: Broadcast

Register VLAN (Port-Based LAN)

Konfiguriert die Verwendung von Port-based VLAN.

HINWEIS



Soll VLAN unterstützt werden, muss Port-based VLAN abgeschaltet sein, so dass jeder Port mit jedem anderen Port des Switches kommunizieren kann.

Für jeden Port eines Switches kann eingestellt werden, zu welchem anderen Port des Switches empfangene Ethernet Frames gesendet werden dürfen. Die Tabelle im Register VLAN enthält Einträge, mit denen die Verbindung zwischen zwei Ports aktiv oder inaktiv geschaltet werden kann.

	Port				
Port (Ethernet-Schnittstelle an MOVISAFE® HM31)	Eth 1 (X4233_1)	Eth 2 (X4233_2)	Eth 3 (X4223)	Eth 4	COM
Eth 1 (X4233_1)					
Eth 2 (X4233_2)	aktiv				

Port (Ethernet-Schnittstelle an MOVISAFE® HM31)	Port				
	Eth 1 (X4233_1)	Eth 2 (X4233_2)	Eth 3 (X4223)	Eth 4	COM
Eth 3 (X4223)	aktiv	aktiv			
Eth 4	aktiv	aktiv	aktiv		
COM	aktiv	aktiv	aktiv	aktiv	
CPU	aktiv	aktiv	aktiv	aktiv	aktiv

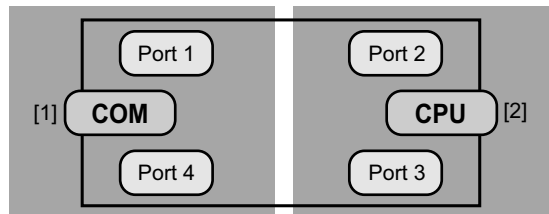
HINWEIS



Port Eth 4 ist ohne Funktion.

Switchports durch VLAN trennen

Es ist möglich, die verfügbaren Ethernet Ports entsprechend der benötigten Anwendung aufzuteilen. So lässt sich eine Verbindung mit zwei IP-Adressen aufbauen, oder eine sichere Kommunikation über die CPU von der nicht sicheren Kommunikation über die COM trennen.



12209301643

[1] Port 1 und Port 4 sind der COM zugeordnet

[2] Port 2 und Port 3 sind der CPU zugeordnet

Die Konfiguration des Switchports erfolgt in SILworX® über die Detailansicht des CPU- oder COM-Moduls. Einstellungen im Register VLAN gemäß obiger Grafik.

Register LLDP (Link Layer Discovery Protocol)

LLDP (Link Layer Discovery Protocol) sendet per Multicast in periodischen Abständen Informationen über das eigene Gerät (z. B. MAC-Adresse, Geräte name, Portnummer) und empfängt die gleichen Informationen von Nachbargeräten.

Das Prozessor- und das Kommunikationsmodul unterstützen LLDP auf den Ports Eth1, Eth2 und Eth3. Einstellungen für Port Eth4 sind ohne Funktion.

Die folgenden Parameter legen fest, wie der betreffende Port arbeitet.

- Aus
LLDP ist auf diesem Port deaktiviert.
- Send
LLDP sendet LLDP Ethernet-Frames, empfangene LLDP Ethernet-Frames werden gelöscht, ohne diese zu verarbeiten.

- Receive
LLDP sendet keine LLDP Ethernet-Frames, aber empfangene LLDP Ethernet-Frames werden verarbeitet.
- Send/Receive
LLDP sendet und verarbeitet empfangene LLDP Ethernet-Frames.

Register Mirroring

Konfiguriert, ob das Modul Ethernet-Pakete auf einen Port dupliziert, so dass sie von einem dort angeschlossenen Gerät mitgelesen werden können, z. B. zu Testzwecken. Die folgenden Parameter legen fest, wie der betreffende Port arbeitet.

- Aus
Dieser Port nimmt am Mirroring nicht teil.
- Egress
Ausgehende Daten dieses Ports werden dupliziert.
- Ingress/Egress
Ein- und ausgehende Daten dieses Ports werden dupliziert.
- Dest Port
Duplizierte Daten werden auf diesen Port geschickt.

12.2.2 Kommunikationsmodul

Das Kommunikationsmodul (COM) enthält die Register "Modul" und "Routings" mit denselben Parametern wie das Prozessormodul. Der Standardwert der IP-Adresse ist hier 192.168.0.100.

12.2.3 Konfiguration der Ressource

Es sind die Eigenschaften der Ressource zu konfigurieren und die Ausgangsvariablen der Hardware.

Eigenschaften der Ressource

Diese Parameter legen das Verhalten der Steuerung während des Betriebs fest und werden in SILworX® im Dialogfenster "Eigenschaften" der Ressource eingestellt.

Parameter/ Schalter	Beschreibung	Standardwert	Einstellung für sicheren Betrieb
Name	Name der Ressource.		Beliebig
System ID [SRS]	System-ID der Ressource. Die System-ID muss einen anderen Wert als den Standardwert erhalten, sonst ist das Projekt nicht ablauffähig. Einstellbereich: 1 – 65535	60000	Eindeutiger Wert innerhalb des Netzwerks der Steuerungen, die potenziell miteinander verbunden sind.
Sicherheitszeit [ms]	Sicherheitszeit in Millisekunden. Einstellbereich: 20 – 22500 ms	600 ms	Applikationsspezifisch

Parameter/ Schalter	Beschreibung	Standardwert	Einstellung für sicheren Betrieb
Watchdog-Zeit [ms]	Watchdog-Zeit in Millisekunden. Einstellbereich: 8 – 5000 ms	200 ms	Applikationsspezifisch
Hauptfreigabe	<p>Nur bei gestopptem PES ist es möglich, <i>Hauptfreigabe</i> auf "ON" zu setzen!</p> <p>ON: Folgende Schalter/Parameter sind im Betrieb (= RUN) mit dem PADT änderbar :</p> <ul style="list-style-type: none"> • <i>System-ID</i> • <i>Watchdog-Zeit der Ressource</i> • <i>Sicherheitszeit</i> • <i>Sollzykluszeit</i> • <i>Sollzykluszeit-Modus</i> • <i>Autostart</i> • <i>Globales Forcen erlaubt</i> • <i>Globale Force-Timeout-Reaktion</i> • <i>Reload-Funktion erlaubt (Funktion über Lizenzfreischaltung verfügbar)</i> • <i>Start erlaubt</i> <p>OFF: Die Parameter sind nicht im Betrieb änderbar.</p>	ON	OFF empfohlen
Autostart	<p>ON: Wird das Prozessorsystem an die Versorgungsspannung angeschlossen, startet das Anwenderprogramm automatisch.</p> <p>OFF: Kein automatischer Start nach Zuschalten der Versorgungsspannung.</p>	OFF	Applikationsspezifisch
Start erlaubt	<p>ON: Kaltstart oder Warmstart durch PADT im Zustand RUN oder STOPP erlaubt.</p> <p>OFF: Kein Start erlaubt.</p>	ON	Applikationsspezifisch

Parameter/ Schalter	Beschreibung	Stan- dardwert	Einstellung für sicheren Be- trieb
Laden erlaubt	ON: Download des Anwenderpro- gramms erlaubt. OFF: Download des Anwenderpro- gramms nicht erlaubt.	ON	Applikationsspe- zifisch
Reload	ON: Reload-Funktion (Funktion über Li- zenzfreischaltung verfügbar) des Anwenderprogramms erlaubt. OFF: Reload-Funktion (Funktion über Li- zenzfreischaltung verfügbar) des Anwenderprogramms nicht erlaubt. Ein laufendes Reload (Funktion über Lizenzfreischaltung verfügbar) wird beim Umschalten auf OFF nicht abgebrochen.	ON	OFF empfohlen
Globales Forcen erlaubt	ON: Globales Forcen für diese Ressour- ce erlaubt. OFF: Globales Forcen für diese Ressour- ce nicht erlaubt.	ON	Applikationsspe- zifisch
Globale Force- Timeout-Reakti- on	Legt fest, wie sich die Ressource beim Ablauf des globalen Force- Timeout verhält: <ul style="list-style-type: none"> • Forcen beenden • Ressource stoppen 	Forcen beenden	Applikationsspe- zifisch
Max. Kom.Zeit- scheibe ASYNC [ms]	Höchstwert in ms der Zeitscheibe, die innerhalb des Zyklus der Res- source für Kommunikation verwen- det wird. Einstellbereich: 2 – 5000 ms	60 ms	Applikationsspe- zifisch
Max. Dauer Kon- figurationsverbin- dungen [ms]	Definiert, wieviel Zeit innerhalb ei- nes CPU-Zyklus für die Prozessda- ten-Kommunikation zur Verfügung steht. Einstellbereich: 6 – 5000 ms	6 ms	-

Parameter/ Schalter	Beschreibung	Standardwert	Einstellung für sicheren Betrieb
Sollzykluszeit [ms]	Gewünschte oder maximale Zykluszeit, siehe <i>Sollzykluszeit-Modus</i> . Die <i>Sollzykluszeit</i> darf höchstens so groß sein wie die eingestellte <i>Watchdogzeit</i> (6 ms), andernfalls lehnt das PES sie ab. Einstellbereich: 0 – 7500 ms	0 ms	-
Sollzykluszeit-Modus	Verwendung der <i>Sollzykluszeit</i> [ms]. fest: Das PES hält die <i>Sollzykluszeit</i> ein und verlängert den Zyklus, falls nötig. Dies gilt nicht, falls die Abarbeitungszeit der Anwenderprogramme die <i>Sollzykluszeit</i> überschreitet. fest-tolerant: Wie bei fest, aber beim 1. Aktivierungszyklus der Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) findet die <i>Sollzykluszeit</i> keine Beachtung. dynamisch-tolerant: Wie bei dynamisch, aber beim 1. Aktivierungszyklus der Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) findet die <i>Sollzykluszeit</i> keine Beachtung. dynamisch: Die Sicherheitssteuerung hält möglichst die <i>Sollzykluszeit</i> ein, führt aber den Zyklus in möglichst kurzer Zeit aus.	fest	-
Minimale Konfigurationsversion	-	SILworX® V4	-
Maximale Systembus-Latenzzeit [µs]	Für die Sicherheitssteuerung nicht anwendbar.	0 ms	-
safeethernet-CRC	In der aktuellen Version erfolgt die Bildung des CRC für safeethernet mit dem aktuellen Algorithmus.	Aktuelle Version	Applikationsspezifisch

Systemvariablen der Hardware zum Erstellen von Parametern

Diese Variablen dienen dazu, das Verhalten der Steuerung im laufenden Betrieb bei bestimmten Zuständen zu verändern. Diese Variablen befinden sich im Hardware-Editor von SILworX®, in der Detailansicht der Hardware.

Variable	Funktion	Standard-einstellung	Einstellung für sicheren Betrieb
Force-Deaktivierung	Dient zum Verhindern und unmittelbaren Abschalten des Forcens.	FALSE	Applikationsspezifisch
Leer 2 – Leer 16	Keine Funktion.	-	-
Notaus 1 – Notaus 4	NOT-AUS-Schalter zum Abschalten der Steuerung in vom Anwenderprogramm erkannten Störfällen.	FALSE	Applikationsspezifisch
Ready-only in RUN	Nach dem Starten der Steuerung ist keine Bedienaktion (Stopp, Start, Download) über SILworX® mehr möglich. Ausnahmen: Forcen und Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar).	FALSE	Applikationsspezifisch
Relaiskontakt 1 – 4	Keine Funktion.	-	-
Reload-Deaktivierung	Verhindert ein Laden der Steuerung mittels Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar).	FALSE	Applikationsspezifisch
User-LED 1 – 2	Steuert die entsprechende LED an, sofern vorhanden.	FALSE	Applikationsspezifisch

Diesen Systemvariablen lassen sich globale Variablen zuweisen, deren Wert durch einen physikalischen Eingang oder die Logik des Anwenderprogramms verändert wird.

Systemvariablen der Hardware zum Auslesen von Parametern

Diese Systemvariablen sind im Hardware-Editor von SILworX® zugänglich. Dazu den grauen Hintergrund außerhalb der (gelben) Baugruppenträger-Darstellung selektieren und die Detailansicht der Hardware durch Doppelklick oder über das Kontextmenü öffnen.

Variable	Beschreibung	Datentyp
Anzahl I/O-Fehler	Anzahl aktueller E/A-Fehler.	UDINT
Anzahl I/O-Fehler historisch	Aufsummierte Anzahl E/A-Fehler (Zähler rücksetzbar).	UDINT
Anzahl E/A-Warnungen	Anzahl aktueller E/A-Warnungen.	UDINT
Anzahl E/A-Warnungen historisch	Aufsummierte Anzahl E/A-Warnungen (Zähler rücksetzbar).	UDINT
Anzahl Kommunikationsfehler	Anzahl aktueller Kommunikationsfehler.	UDINT

Variable	Beschreibung	Datentyp
Anzahl Kommunikationsfehler historisch	Aufsummierte Anzahl Kommunikationsfehler (Zähler rücksetzbar).	UDINT
Anzahl Kommunikationswarnungen	Anzahl aktueller Kommunikationswarnungen.	UDINT
Anzahl Kommunikationswarnungen historisch	Aufsummierte Anzahl Kommunikationswarnungen (Zähler rücksetzbar).	UDINT
Anzahl Systemfehler	Anzahl aktueller Systemfehler.	UDINT
Anzahl Systemfehler historisch	Aufsummierte Anzahl Systemfehler (Zähler rücksetzbar).	UDINT
Anzahl Systemwarnungen	Anzahl aktueller Systemwarnungen.	UDINT
Anzahl Systemwarnungen historisch	Aufsummierte Anzahl Systemwarnungen (Zähler rücksetzbar).	UDINT
Autostart CPU Release	ON: Das Prozessorsystem startet beim Anlegen der Versorgungsspannung automatisch das Anwenderprogramm. OFF: Das Prozessorsystem geht beim Anlegen der Versorgungsspannung in den Zustand STOPP.	BOOL
BS Major	Ausgabe des Betriebssystems im Prozessorsystem.	UINT
BS Minor		UINT
CRC	Prüfsumme der Projektkonfiguration.	UDINT
Datum/Uhrzeit [ms-Anteil]	Systemdatum und -uhrzeit in s und ms seit 01.01.1970.	UDINT
Datum/Uhrzeit [Sek.-Anteil]		UDINT
Force-Deaktivierung	ON: Forcen ist deaktiviert. OFF: Forcen ist möglich.	BOOL
Forcen aktiv	ON: Globales oder lokales Forcen ist aktiv. OFF: Globales und lokales Forcen sind nicht aktiv.	BOOL
Force-Schalterzustand	Zustand der Force-Schalter. 0xFFFFFFFFE: Kein Force-Schalter gesetzt 0xFFFFFFFFF: Mindestens ein Force-Schalter gesetzt	UDINT

Variable	Beschreibung	Datentyp
Globales Forcen gestartet	ON: Globales Forcen ist aktiv. OFF: Globales Forcen ist nicht aktiv.	BOOL
Leer 0 – 16	Reserviert.	USINT
Leer ein17		BOOL
Letzte I/O-Warnung [ms]	Datum und Uhrzeit der letzten I/O-Warnung in s und ms seit 01.01.1970.	UDINT
Letzte I/O-Warnung [s]		UDINT
Letzte Kommunikationswarnung [ms]	Datum und Uhrzeit der letzten Kommunikationswarnung in s und ms seit 01.01.1970.	UDINT
Letzte Kommunikationswarnung [s]		UDINT
Letzte Systemwarnung [ms]	Datum und Uhrzeit der letzten Systemwarnung in s und ms seit 01.01.1970.	UDINT
Letzte Systemwarnung [s]		UDINT
Letzter I/O-Fehler [ms]	Datum und Uhrzeit des letzten I/O-Fehlers in s und ms seit 01.01.1970.	UDINT
Letzter I/O-Fehler [s]		UDINT
Letzter Kommunikationsfehler [ms]	Datum und Uhrzeit des letzten Kommunikationsfehlers in s und ms seit 01.01.1970.	UDINT
Letzter Kommunikationsfehler [s]		UDINT
Letzter Systemfehler [ms]	Datum und Uhrzeit des letzten Systemfehlers in s und ms seit 01.01.1970.	UDINT
Letzter Systemfehler [s]		UDINT
Lüfterzustand	0xFF: Nicht vorhanden	BYTE
Major CPU Release	Haupt-Freigabeschalter des Prozessorsystems: ON: Die untergeordneten Freigabeschalter können verändert werden. OFF: Die untergeordneten Freigabeschalter können nicht verändert werden.	BOOL
Read-only in RUN	ON: Die Bedienaktionen Stopp, Start, Download sind gesperrt. OFF: Die Bedienaktionen Stopp, Start, Download sind nicht gesperrt.	BOOL

Variable	Beschreibung	Datentyp
Reload Release	ON: Steuerung kann mittels Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) geladen werden. OFF: Die Steuerung kann nicht mittels Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) geladen werden.	BOOL
Reload-Deaktivierung	ON: Laden mittels Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) ist gesperrt. OFF: Laden mittels Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) ist möglich.	BOOL
Reload-Zyklus	TRUE im ersten Zyklus nach einer Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar), sonst FALSE.	BOOL
Sicherheitszeit CPU [ms]	Für die Steuerung eingestellte Sicherheitszeit in ms.	UDINT
Start CPU Release	ON: Start des Prozessorsystems durch das PADT erlaubt. OFF: Start des Prozessorsystems durch das PADT nicht erlaubt.	BOOL
Start Cycle	ON während erstem Zyklus nach dem Start, sonst OFF.	BOOL
Stromversorgungszustand	Bitcodierter Zustand der Spannungsversorgung.	
	Wert	Zustand
	0x00	Normal
	0x01	Unterspannung bei Versorgungsspannung 24 V.
	0x02	(Unterspannung bei Batterie) unbe- nutzt.
	0x04	Unterspannung bei intern erzeugter Spannung 5 V.
	0x08	Unterspannung bei intern erzeugter Spannung 3.3 V.
	0x10	Überspannung bei intern erzeugter Spannung 3.3 V.

Variable	Beschreibung	Datentyp
System ID	System ID der Steuerung. Einstellbereich: 1 – 65535	UINT
Systemtick HIGH	Umlaufender Millisekundenzähler (64 Bit).	UDINT
Systemtick LOW		UDINT
Temperaturzustand	Bitcodierter Temperaturzustand des Prozessorsystems.	BYTE
	Wert Zustand	
	0x00 Normale Temperatur	
	0x01 Temperaturschwelle 1 überschritten	
	0x03 Temperaturschwelle 2 überschritten	
	0xFF Nicht vorhanden	
Verbleibende globale Force-Dauer [ms]	Zeit in ms bis zum Ablauf der globalen Force-Zeitbegrenzung.	DINT
Watchdog-Zeit CPU [ms]	Höchste zulässige Dauer eines RUN-Zyklus in ms.	UDINT
Zykluszeit, letzte [ms]	Aktuelle Zykluszeit in ms.	UDINT
Zykluszeit, max [ms]	Maximale Zykluszeit in ms.	UDINT
Zykluszeit, min [ms]	Minimale Zykluszeit in ms.	UDINT
Zykluszeit, mittlere [ms]	Mittlere Zykluszeit in ms.	UDINT

Konfiguration des Anwenderprogramms

Die folgenden Schalter und Parameter eines Anwenderprogramms lassen sich im Dialogfenster "Eigenschaften" des Anwenderprogramms einstellen.

Parameter/Schalter	Beschreibung	Standardwert	Einstellung für sicheren Betrieb
Name	Name des Anwenderprogramms.		Beliebig
Sicherheitsintegritätslevel	Sicherheitslevel: SIL 0, SIL 3	SIL 3	Applikationsspezifisch
Start erlaubt	ON: Start des Anwenderprogramms durch das PADT erlaubt. OFF: Start des Anwenderprogramms durch das PADT nicht erlaubt.	ON	Applikationsspezifisch
Programm-Hauptfreigabe	Freigabe der Änderung an anderen Anwenderprogramm-Schaltern: Es ist nur der Freigabe-Schalter der Ressource relevant!	ON	Applikationsspezifisch
Autostart	Freigegebene Art des Autostarts: Kaltstart, Warmstart, Aus.	Warmstart	Applikationsspezifisch

Parameter/Schalter	Beschreibung	Standardwert	Einstellung für sicheren Betrieb
Testbetrieb erlaubt	ON: Für das Anwenderprogramm ist der Testbetrieb erlaubt. OFF: Für das Anwenderprogramm ist der Testbetrieb nicht erlaubt.	OFF	Applikationsspezifisch
Lokales Forcen erlaubt	ON: Forcen auf Programmebene erlaubt. OFF: Forcen auf Programmebene nicht erlaubt.	OFF	OFF empfohlen
Lokale Force-Timeout-Reaktion	Verhalten des Anwenderprogramms nach Ablauf der Force-Zeit: <ul style="list-style-type: none"> Nur Forcen beenden Programm stoppen 	Nur Forcen beenden	Applikationsspezifisch
Reload Erlaubt	ON: Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) des Anwenderprogramms erlaubt. OFF: Reload-Funktion (Funktion über Lizenzfreischaltung verfügbar) des Anwenderprogramms nicht erlaubt.	ON	Applikationsspezifisch
Maximale CPU-Zyklen Programm	Maximale Anzahl an CPU-Zyklen, die ein Zyklus des Anwenderprogramms dauern darf. Ein Wert > 1 ist zulässig.	1	Applikationsspezifisch
Max. Dauer pro Zyklus [µs]	Maximale Ausführungsdauer pro Zyklus des Prozessormoduls für ein Anwenderprogramm. Einstellbereich: 1 – 7 500 000 µs 0: Keine Begrenzung	0 µs	0 µs
Programm ID	ID für die Identifizierung des Programms bei der Anzeige in SILworX®. Einstellbereich: 1 – 32	1	Applikationsspezifisch
Watchdog-Zeit [ms] (berechnet)	Nicht änderbare Überwachungszeit des Anwenderprogramms, errechnet aus <i>Maximale CPU-Zyklen Programm</i> und <i>Watchdog-Zeit der Resource</i> . Hinweis: Werden Zählereingänge verwendet, ist darauf zu achten, dass die Watchdog-Zeit des Anwenderprogramms ≤ 5 000 ms ist.		-

12.2.4 Konfiguration der Ein- und Ausgänge

Im Hardware-Editor erfolgt die Konfiguration der Eingänge und Ausgänge dadurch, dass den Systemvariablen für die Eingangs- oder Ausgangskanäle globale Variablen zugewiesen werden.

So gelangen Sie zu den Systemvariablen der Kanäle:

1. Im Hardware-Editor die gewünschte Ressource anzeigen.
2. Durch Doppelklick auf das gewünschte Eingangs- oder Ausgangsmodul die Detailansicht öffnen.
3. In der Detailansicht das Register mit den gewünschten Kanälen öffnen. Die Systemvariablen der Kanäle sind sichtbar.

Verwendung digitaler Eingänge

Folgende Schritte sind notwendig, um den Wert eines digitalen Eingangs im Anwenderprogramm zu verwenden:

1. Eine globale Variable vom Typ BOOL definieren.
2. Bei der Definition einen geeigneten Initialwert angeben.
3. Die globale Variable dem Kanalwert des Eingangs zuweisen.
4. Im Anwenderprogramm eine sicherheitsgerichtete Fehlerreaktion unter Verwendung des Fehlercodes *-> Fehlercode [Byte]* programmieren.

Die globale Variable liefert Werte ins Anwenderprogramm.

Durch Zuweisen globaler Variablen auf *DI.Fehlercode* und *ModulFehlercode* bestehen zusätzliche Möglichkeiten, Fehlerreaktionen im Anwenderprogramm zu programmieren. Einzelheiten zu den Fehlercodes finden Sie im Kapitel "Parameter und Fehlercodes der Ein- und Ausgänge".

Verwendung sicherheitsgerichteter Zählereingänge

Es ist möglich, den Zählerstand oder die Drehzahl/Frequenz als ganzzahligen Wert oder als skalierten Gleitkommawert zu verwenden. Im Folgenden bezeichnet "xx" die jeweilige Kanalnummer.

Folgende Schritte sind notwendig, um den ganzzahligen Wert verwenden:

1. Eine globale Variable vom Typ UDINT definieren.
2. Bei der Definition einen geeigneten Initialwert angeben.
3. Die globale Variable dem ganzzahligen Wert *Zähler[xx].Wert* des Eingangs zuweisen.
4. Im Anwenderprogramm eine sicherheitsgerichtete Fehlerreaktion unter Verwendung des Fehlercodes *Zähler[xx].Fehlercode [Byte]* programmieren.

Die globale Variable liefert Werte ins Anwenderprogramm.

Durch Zuweisen globaler Variablen auf *Zähler.Fehlercode* und *ModulFehlercode* bestehen zusätzliche Möglichkeiten, Fehlerreaktionen im Anwenderprogramm zu programmieren. Einzelheiten zu den Fehlercodes finden Sie im Kapitel "Parameter und Fehlercodes der Ein- und Ausgänge".

Verwendung digitaler Ausgänge

Folgende Schritte sind notwendig, um einen Wert im Anwenderprogramm auf einen digitalen Ausgang zu schreiben:

1. Eine globale Variable vom Typ BOOL definieren, die den auszugebenden Wert erhält.
2. Bei der Definition einen geeigneten Initialwert angeben.
3. Die globale Variable dem Kanalwert Wert [BOOL] -> des Ausganges zuweisen.
4. Im Anwenderprogramm eine sicherheitsgerichtete Fehlerreaktion unter Verwendung des Fehlercodes -> *Fehlercode [Byte]* programmieren.

Die globale Variable liefert Werte an den digitalen Ausgang.

Durch Zuweisen globaler Variable auf *DO.Fehlercode* und *ModulFehlercode* bestehen zusätzliche Möglichkeiten, Fehlerreaktionen im Anwenderprogramm zu programmieren.

12.2.5 Generierung der Ressourcenkonfiguration

Gehen Sie so vor:

1. Im Strukturbaum die Ressource auswählen.
2. In der Aktionsleiste auf die Schaltfläche [Codegenerierung] klicken oder im Kontextmenü den Eintrag [Codegenerierung] auswählen. Das Dialogfenster "Codegenerierung starten" öffnet sich.
3. Im Dialogfenster "Codegenerierung starten" auf [OK] klicken. Ein weiteres Dialogfenster "Codegenerierung starten" öffnet sich, zeigt den Ablauf der Codegenerierung an und schließt sich wieder. Im Logbuch erscheint eine Zeile, die das Ergebnis der Codegenerierung anzeigt.
4. Bei weiterhin ausgewählter Ressource aus dem Menü [Extras] den Eintrag [Versionsvergleich] auswählen. Das Dialogfenster "Versionsübersicht" öffnet sich. Es enthält den CRC des generierten Codes.
5. Auf [Export] klicken. Es erscheint ein Dialogfenster "Archivieren", das Eingabemöglichkeiten für einen Kommentar zum Projektstand und für den Namen der Archivdatei enthält.
6. Ein weiteres Mal Code generieren, wie in den Schritten 2 und 3 beschrieben.
7. Bei weiterhin ausgewählter Ressource aus dem Menü [Extras] den Eintrag [Versionsvergleich] auswählen. Das Dialogfenster "Versionsübersicht" öffnet sich.
8. Auf [Import] klicken und im Dialogfenster "Wiederherstellen" die in Schritt 5 exportierte Archivdatei importieren. Das Fenster "Versionsübersicht" enthält nun die Informationen zum letzten generierten und zum importierten Projektstand.
9. Auf [OK] klicken. Im Arbeitsbereich erscheint das Ergebnis des Versionsvergleichs. Erscheint "ok" in der Spalte "Vergleich der CRCs", sind die generierten Codes beider Projektstände gleich und geeignet für den sicherheitsgerichteten Betrieb. Abweichungen sind durch Hinterlegung mit roter Farbe gekennzeichnet.

Damit ist der Code der Ressourcenkonfiguration generiert.

ACHTUNG



Fehler bei der Codegenerierung durch nicht sicheren PC möglich!

Für sicherheitsgerichtete Anwendungen muss der Codegenerator zweimal Code generieren und die Prüfsummen (CRCs) beider Generierungsdurchläufe müssen miteinander übereinstimmen. Nur dann ist ein fehlerfreier Code sichergestellt.

12.2.6 System-ID und Verbindungsparameter konfigurieren

Gehen Sie folgendermaßen vor:

1. Im Strukturbaum die Ressource auswählen
2. In der Aktionsleiste auf die Schaltfläche [Online] klicken oder im Kontextmenü den Eintrag "Online" auswählen.
Das Dialogfenster System-Login öffnet sich.
3. Auf [Suchen] klicken
Das Dialogfenster "Suchen per MAC" öffnet sich.
4. Die für die Steuerung gültige MAC-Adresse (siehe Aufkleber auf dem Gehäuse) eingeben und auf [Suchen] klicken.
Das Dialogfenster zeigt die in der Steuerung eingestellten Werte für IP-Adresse, Subnet Mask und SRS an.
5. Zum Übernehmen der Werte die Schaltfläche [übernehmen] klicken
6. Mit dem Benutzer "Administrator" einloggen
7. Menü [Online] / [Inbetriebnahme] / [System-ID einstellen] wählen und gewünschte System-ID vergeben
Die Änderung wird sofort wirksam, sodass die Verbindung abbricht.
8. Falls noch nicht geschehen: IP-Adresse COM&CPU über Hardware vergeben und Projekt übersetzen.
9. Zur Eingabe weiterer IP-Adressen/System-IDs die Schritte 1 bis 6 wiederholen
10. Programm in die Steuerung laden
Die Änderung wird sofort wirksam, sodass die Verbindung abbricht.

Jetzt können Sie sich mit den im Projekt eingestellten IP-Adressen und System-ID einloggen.

Für einen Systemverbund mehrerer Sicherheitssteuerungen empfehlen wir Ihnen, die einzelnen Steuerungen nacheinander zu konfigurieren und danach diese in den Netzverbund aufzunehmen. In der Gerätegrundkonfiguration der Sicherheitssteuerung ist für die IP-Adresse ein Standardwert eingetragen. Hierdurch ist die Zuordnung der Steuerung nur durch die MAC-Adresse der einzelnen Steuerung möglich.

▲ WARNUNG



Vertauschungsgefahr der angesprochenen Steuerung. Dadurch kann ein falsches Anwenderprogramm in die Sicherheitssteuerung geladen werden.

Tod oder schwere Körperverletzungen.

12.2.7 Ressourcenkonfiguration vom Programmiergerät laden

Bevor ein Anwenderprogramm zusammen mit den Verbindungsparametern der Steuerung (IP-Adresse, Subnetzmaske und System-ID) in die Steuerung geladen werden kann, muss der Code für die Ressource generiert worden sein und das Programmiergerät und die Ressource müssen gültige Verbindungsparameter haben (siehe Kapitel "System-ID und Verbindungsparameter konfigurieren").

Gehen Sie folgendermaßen vor, um die Ressourcenkonfiguration vom Programmiergerät zu laden:

1. Ressource im Strukturbaum wählen.

2. In der Aktionsleiste [Online] klicken oder aus dem Kontextmenü den Eintrag [Online] wählen.
3. In Fenster "System-Login" eine Benutzergruppe mit Administrator-Rechten oder Schreibzugang angeben. Das Controlpanel öffnet sich im Arbeitsbereich und zeigt den Zustand der Steuerung an.
4. Im Menü [Online] den Eintrag [Ressource Download] wählen. Das Dialogfenster "Ressource Download" öffnet sich.
5. Im Dialogfenster den Download mit "OK" bestätigen. SILworX lädt die Konfiguration in die Steuerung.
6. Nach dem Laden das Anwenderprogramm mit dem Eintrag [Ressource Kaltstart] des Menüs [Online] starten. Nach dem Kaltstart gehen "Systemzustand" und "Programm-Status" in den Modus RUN.

Die Ressourcenkonfiguration ist vom Programmiergerät geladen. Die Funktionen "Starten", "Stoppen" und "Laden" sind auch als Symbole in der Symbolleiste verfügbar.

12.2.8 Ressourcenkonfiguration aus dem Flash-Speicher des Kommunikationssystems laden

Bei Datenfehler im NVRAM und damit verbundener Überschreitung der Watchdog-Zeit kann es sinnvoll sein, die Ressourcenkonfiguration aus dem Flash-Speicher des Kommunikationssystems, anstatt vom Programmiergerät zu laden.

Besteht kein Zugang mehr zum Control Panel (CP), müssen die Verbindungsparameter vom Anwenderprogramm in die Steuerung neu gesetzt werden, siehe Kapitel "System-ID und Verbindungsparameter konfigurieren").

Geht die Steuerung nach dem Neustart in den Zustand STOPP/GÜLTIGE KONFIGURATION, kann das Anwenderprogramm wieder gestartet werden.

Geht die Steuerung nach dem Neustart in den Zustand STOPP/UNGÜLTIGE KONFIGURATION, ist das Anwenderprogramm wieder ins NVRAM zu laden.

Mit dem Befehl [Konfiguration aus Flash] laden kann eine Sicherheitskopie der letzten, lauffähigen Konfiguration aus dem Flash-Speicher des Kommunikationssystems ausgelesen und in das NVRAM des Prozessors übertragen werden. Nun lässt sich das Anwenderprogramm mit [Online] / [Ressource Kaltstart] wieder starten, ohne dass ein Download des Projektes erforderlich wurde.

Gehen Sie folgendermaßen vor, um die Ressourcenkonfiguration aus dem Flash-Speicher des Kommunikationssystems zu laden:

1. Bei der gewünschten Ressource anmelden.
2. Im Menü [Online] das Untermenü [Wartung/Service] und dort den Eintrag [Konfiguration aus Flash laden] wählen.
3. Das Laden der Konfiguration im Dialogfenster bestätigen.

Die Steuerung lädt die Ressourcenkonfiguration aus dem Flash-Speicher des Kommunikationssystems ins NVRAM.

12.2.9 Ressourcenkonfiguration im Flash-Speicher des Kommunikationssystems bereinigen

Nach temporären Fehlern der Hardware ist es möglich, dass der Flash-Speicher des Kommunikationssystems Reste ungültiger Konfigurationen enthält.

Zur Beseitigung dieser Reste gibt es den Befehl [Konfiguration bereinigen].

Ressourcenkonfiguration bereinigen:

1. Ressource im Strukturbaum wählen.

2. In der Aktionsleiste [Online] klicken oder aus dem Kontextmenü den Eintrag [Online] wählen.
3. In Fenster "System-Login" eine Benutzergruppe mit Administratorrechten oder Schreibzugang angeben. Das Controlpanel öffnet sich im Arbeitsbereich und zeigt den Zustand der Steuerung an.
4. Aus dem Menü [Online] und dem Untermenü [Wartung/Service] den Eintrag [Konfiguration bereinigen] wählen.
5. Aktion im Dialogfenster "Konfiguration bereinigen" mit [OK] bestätigen.

Die Konfiguration im Flash-Speicher des Kommunikationssystems wurde bereinigt. Das Bereinigen der Konfiguration ist nur in seltenen Fällen notwendig. Eine gültige Konfiguration bleibt beim Bereinigen unangetastet.

12.2.10 Datum und Uhrzeit setzen

Gehen Sie so vor:

1. Ressource im Strukturbaum wählen.
2. In der Aktionsleiste [Online] klicken oder aus dem Kontextmenü den Eintrag [Online] wählen.
3. In Fenster "System-Login" eine Benutzergruppe mit Administrator-Rechten oder Schreibzugang angeben. Das Controlpanel öffnet sich im Arbeitsbereich und zeigt den Zustand der Steuerung an.
4. Aus dem Menü [Online] und dem Untermenü [Inbetriebnahme] den Eintrag [Datum/Uhrzeit einstellen] wählen. Das Dialogfenster "Datum/Uhrzeit einstellen" öffnet sich.

Eine der Optionen auswählen:

- Datum und Uhrzeit des Programmiergeräts verwenden.

Dadurch wird die angezeigte Uhrzeit mit Datum des Programmiergeräts in die Steuerung übertragen.

- Benutzerdefiniert.

Datum und Uhrzeit aus den beiden Eingabefeldern werden in die Steuerung übertragen. Beim Eingeben von Datum / Uhrzeit das angegebene Format beachten.

6. Klicken auf [OK] überträgt Datum und Uhrzeit auf die Steuerung. Datum und Uhrzeit auf der Steuerung sind gesetzt.

12.3 Benutzerverwaltung mit SILworX®

SILworX® kann eigene Benutzerverwaltungen für jedes Projekt und für jede Steuerung einrichten und pflegen.

12.3.1 Benutzerverwaltung für ein SILworX®-Projekt

In jedes SILworX®-Projekt lässt sich eine PADT-Benutzerverwaltung einfügen, die den Zugang zum Projekt in SILworX® regelt.

Ohne PADT-Benutzerverwaltung kann jeder Benutzer ein Projekt öffnen und alle Bestandteile ändern. Hat ein Projekt eine Benutzerverwaltung, lässt es sich nur durch einen Benutzer öffnen, der sich authentifiziert hat. Der Benutzer kann nur dann Änderungen durchführen, wenn er dazu berechtigt ist. Es gibt folgende Stufen der Berechtigung.

Stufe	Bedeutung
Sicherheitsadministrator (Sec Adm)	Kann die Benutzerverwaltung ändern: Einrichten, Löschen, Ändern von Benutzerkonten und Benutzergruppen und der PADT-Benutzerverwaltung, Festlegen des Standard-Benutzerkontos. Außerdem sind alle sonstigen Funktionen von SILworX® zulässig.
Lesen/Schreiben (R/W)	Alle Funktionen von SILworX®, mit Ausnahme der Benutzerverwaltung.
Nur Lesen (RO)	Nur lesende Zugriffe, keine Änderungen, kein Archivieren.

Die Benutzerverwaltung vergibt die Berechtigung an Benutzergruppen. Die Benutzerkonten erhalten ihre Berechtigung von der Benutzergruppe, der sie zugeordnet sind.

Eigenschaften von Benutzergruppen:

- Der Name muss im Projekt eindeutig sein und 1 – 31 Zeichen enthalten.
- Einer Benutzergruppe ist eine Berechtigungsstufe zugeordnet.
- Einer Benutzergruppe können beliebig viele Benutzerkonten zugeordnet sein.
- Ein Projekt kann bis zu 100 Benutzergruppen enthalten.

Eigenschaften von Benutzerkonten:

- Der Name muss im Projekt eindeutig sein und 1 – 31 Zeichen enthalten.
- Ein Benutzerkonto ist einer Benutzergruppe zugeordnet.
- Ein Projekt kann bis zu 1000 Benutzerkonten enthalten.
- Ein Benutzerkonto kann Standardbenutzer des Projekts sein.

12.3.2 Benutzerverwaltung für die Steuerung

Die Benutzerverwaltung für eine Steuerung (PES-Benutzerverwaltung) dient dazu, eine Sicherheitssteuerung vor unberechtigten Eingriffen zu schützen. Die Benutzer und ihre Zugriffsrechte sind ein Teil des Projekts und werden mit SILworX® definiert und auf das Prozessormodul geladen.

Die Benutzerverwaltung kann Zugriffsrechte für maximal zehn Anwender einer Steuerung verwalten. Die Zugriffsrechte sind in der Steuerung abgelegt und bleiben auch nach dem Ausschalten der Betriebsspannung erhalten.

Jedes Benutzerkonto besteht aus Name, Passwort und Zugriffsrecht. Sobald das Projekt per Download auf die Steuerung übertragen wurde, stehen diese Informationen für Logins zur Verfügung. Die Benutzer identifizieren sich beim Login auf die Steuerung mit ihrem Namen und Passwort.

Es ist nicht erforderlich, Benutzerkonten anzulegen, dieses trägt jedoch zum sicheren Betrieb bei. Ist für eine Ressource eine Benutzerverwaltung definiert, muss diese mindestens einen Benutzer mit Administratorrechten enthalten.

Standardbenutzer

Sind für eine Ressource keine anwenderspezifischen Benutzerkonten eingerichtet, gelten die werkseitigen Einstellungen.

Werkseinstellungen:

- Anzahl der Benutzer: 1
- Benutzerkennung: Administrator
- Passwort: ohne
- Zugriffsrecht: Administrator

HINWEIS



Beachten Sie, dass es beim Definieren eigener Benutzerkonten nicht möglich ist, die Standardeinstellung beizubehalten.

Parameter für Benutzerkonten

Beim Einrichten neuer Benutzerkonten sind die folgenden Parameter zu definieren.

Parameter	Beschreibung
Benutzername	Name oder Kennzeichen des Benutzers, unter dem er sich in der Steuerung einloggt. Der Benutzername darf nicht mehr als 32 Zeichen enthalten (empfohlen: max. 16 Zeichen) und darf nur aus Buchstaben (A bis Z, a bis z), Zahlen (0 bis 9) und den Sonderzeichen Unterstrich "_" und Bindestrich "-" bestehen. Groß-/Kleinschreibung beachten.
Passwort	Zum Benutzername gehörendes Kennwort, das zum Einloggen erforderlich ist. Das Passwort darf nicht mehr als 32 Zeichen enthalten und darf nur aus Buchstaben (A bis Z, a bis z), Zahlen (0 bis 9) und den Sonderzeichen Unterstrich "_" und Bindestrich "-" bestehen. Groß-/Kleinschreibung beachten.
Passwort bestätigen	Wiederholung des Kennwortes zur Bestätigung der Eingabe.

Parameter	Beschreibung
Zugriffsart	<p>Die Zugriffsarten definieren die Privilegien, die ein Benutzer haben kann. Folgende Zugriffsarten sind möglich:</p> <ul style="list-style-type: none"> • Lesen: Der Benutzer darf nur Informationen von der Steuerung lesen, aber keine Änderungen durchführen. • Lesen und Bediener: Wie Lesen, zusätzlich darf der Benutzer: <ul style="list-style-type: none"> – Anwenderprogramme per Download laden und starten – Prozessormodule in Redundanz setzen – Zykluszeit- und Fehlerstatistiken zurücksetzen – Systemzeit stellen – Forcen – Module neu starten und zurücksetzen – bei Prozessormodulen den Systembetrieb starten • Lesen und Schreiben: Wie Lesen und Bediener, zusätzlich darf der Benutzer Programme erstellen, übersetzen, in die Steuerung laden und testen. • Administrator: Wie Lesen und Schreiben, zusätzlich darf der Benutzer: <ul style="list-style-type: none"> – Betriebssysteme laden – Hauptfreigabeschalter ändern – SRS ändern – IP-Einstellungen ändern <p>Wenigstens einer der Benutzer muss über Administratorrechte verfügen, andernfalls akzeptiert die Steuerung die Einstellungen nicht. Der Administrator kann einem Benutzer nachträglich den Zugriff auf eine Steuerung entziehen, indem er den Benutzer gänzlich aus der Liste entfernt.</p>

Einrichten von Benutzerkonten

Ein Benutzer mit Administratorrechten hat Zugriff auf alle Benutzerkonten. Beim Einrichten von Benutzerkonten ist Folgendes zu beachten:

- Es ist sicherzustellen, dass wenigstens ein Benutzerkonto mit Administratorrechten eingerichtet ist. Für ein Benutzerkonto mit Administratorrechten ein Passwort definieren.
- Wenn der Administrator in der Benutzerverwaltung ein Benutzerkonto erstellt hat und dieses Benutzerkonto erneut bearbeiten möchte, muss er zur Legitimierung das Passwort des Benutzerkontos eingeben.
- Die Verifikation von SILworX® verwenden, um die eingerichteten Benutzerkonten zu überprüfen.
- Nach der Codegenerierung und einem Download des Projekts auf die Steuerung werden die neuen Benutzerkonten gültig. Alle zuvor gespeicherten Benutzerkonten, z. B. die Standardeinstellung, werden ungültig!

12.4 Konfiguration der Kommunikation mit SILworX®

Dieses Kapitel beschreibt die Konfiguration der Kommunikation bei Einsatz des Programmierwerkzeugs SILworX®.

Zu konfigurieren sind je nach Anwendung

- Ethernet/safeethernet
- Standardprotokolle

Für die Konfiguration der Standardprotokolle siehe Kapitel "Modbus TCP/UDP".

12.4.1 Konfiguration der Ethernet-Schnittstellen

Die Konfiguration erfolgt in der Detailansicht des Kommunikationsmoduls (COM).

HINWEIS



SILworX® stellt das Prozessorsystem und das Kommunikationssystem innerhalb eines Geräts oder einer Baugruppe als Prozessormodul und Kommunikationsmodul dar.

Für die Sicherheitssteuerung in den Ethernet-Switch-Einstellungen die Parameter *Speed [Mbit/s]* und *Flow-Control* auf "Autoneg" einstellen. Die Parameter *ARP Aging Time*, *MAC Learning*, *IP Forwarding*, *Speed [Mbit/s]* und *Flow-Control* sind ausführlich in der Online-Hilfe von SILworX® erklärt.

HINWEIS



Austausch einer Steuerung mit gleicher IP-Adresse:

Beim Austausch einer Steuerung, für die *ARP Aging Time* = 5 Minuten und *MAC Learning* = *Konservativ* eingestellt ist, übernimmt der Kommunikationspartner erst nach mindestens 5 Minuten bis höchstens 10 Minuten die neue MAC-Adresse. In dieser Zeit ist keine Kommunikation mit der ausgetauschten Steuerung möglich.

Die Port-Einstellungen des integrierten Ethernet-Switches der Sicherheitssteuerung lassen sich individuell parametrieren. Im Register "Ethernet-Switch" kann für jeden Switch-Port ein Tabelleneintrag angelegt werden.

Parameter der Port-Konfiguration	Erklärung
Port	Nummer des Ports wie Gehäuseaufdruck. Pro Port darf nur eine Konfiguration vorhanden sein. Wertebereich: 1 – n, je nach Ressource.
Speed [Mbit/s]	10 Mbit/s: Datenrate 10 Mbit/s 100 Mbit/s: Datenrate 100 Mbit/s Autoneg (10/100): Automatische Einstellung der Baudrate Standard: Autoneg
Flow-Control	Vollduplex: Kommunikation in beide Richtungen gleichzeitig Halbduplex: Kommunikation in eine Richtung zu einer Zeit Autoneg: Automatische Kommunikationssteuerung Standard: Autoneg

Parameter der Port-Konfiguration	Erklärung
Autoneg auch bei festen Werten	Das Übermitteln der <i>Speed</i> - und <i>Flow-Control</i> -Eigenschaften (Advertising) wird auch bei fest eingestellten Werten von <i>Speed</i> und <i>Flow-Control</i> durchgeführt. Hierdurch können andere Geräte, deren Ports auf <i>Autoneg</i> eingestellt sind, erkennen, wie die Ports der Sicherheitssteuerung eingestellt sind.
Limit	Eingehende Multicast- und/oder Broadcast-Pakete limitieren. Aus: Keine Limitierung Broadcast: Broadcast limitieren (128 kbit/s) Multicast und Broadcast: Multicast und Broadcast limitieren (1024 kbit/s) Standard: Broadcast

Die Parameter lassen sich durch Doppelklicken auf jede Zelle der Tabelle ändern und in die Konfiguration des Kommunikationssystems eintragen. Die Einträge sind mit dem Anwenderprogramm neu zu kompilieren und in die Steuerung zu übertragen, bevor sie für die Kommunikation der Sicherheitssteuerung wirksam werden.

Die Eigenschaften des Kommunikationssystems und des Ethernet-Switches sind auch online über das Control Panel änderbar. Diese Einstellungen werden sofort wirksam, aber nicht in das Anwenderprogramm übernommen.

Einzelheiten zur Konfiguration der safeethernet-Kommunikation finden Sie im Kapitel "safeethernet".

12.5 Konfigurieren von Alarmen und Ereignissen

Definition von Ereignissen:

1. Für jedes Ereignis eine globale Variable definieren. In der Regel globale Variablen verwenden, die bereits für das Programm definiert sind.
2. Unter der Ressource einen neuen Unterzweig "Alarm & Events" erzeugen, falls dieser noch nicht existiert.

Im Alarm & Event-Editor Ereignisse definieren.

- Globale Variable ins Ereignisfenster für boolesche oder skalare Ereignisse ziehen.
- Die Einzelheiten der Ereignisse festlegen, siehe die beiden nachfolgenden Tabellen.

Ereignisse sind definiert. Weitere Informationen finden Sie in der Online-Hilfe zu SILworX®.

Die **Parameter der Booleschen Ereignisse** sind in eine Tabelle einzugeben, die folgende Spalten enthält.

Spalte	Beschreibung	Wertebereich
Name	Name der Ereignisdefinition, muss in der Ressource eindeutig sein.	Text, max. 32 Zeichen
Globale Variable	Name der zugewiesenen globalen Variable (eingefügt z. B. durch Drag & Drop).	
Datentyp	Datentyp der globalen Variable, nicht änderbar.	BOOL

Spalte	Beschreibung	Wertebereich
Event-Quelle	CPU Event: Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch. Auto Event: Wie CPU Event Standardwert: Auto Event	CPU, Auto
Alarm bei FALSE	Aktiviert: Die Wertänderung TRUE → FALSE der globalen Variablen löst ein Ereignis aus. Deaktiviert: Die Wertänderung FALSE → TRUE der globalen Variablen löst ein Ereignis aus. Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
Alarm-Text	Text, der den Alarmzustand benennt.	Text
Alarm-Priorität	Priorität des Alarmzustands. Standardwert: 500	0 – 1000
Alarmbestätigung erforderlich	Aktiviert: Bestätigung des Alarmzustandes durch den Bediener erforderlich (Quittierung). Deaktiviert: Bestätigung des Alarmzustandes durch den Bediener nicht erforderlich. Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
Return to Normal Text	Text, der den Alarmzustand benennt.	Text
Return to Normal Severity	Priorität des Normalzustands. Standardwert: 500	0 – 1000
Return to Normal Ack Required	Bestätigung des Normalzustandes durch den Bediener erforderlich (Quittierung). Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert

Die **Parameter der skalaren Ereignisse** sind in eine Tabelle einzugeben, die folgende Spalten enthält.

Spalte	Beschreibung	Wertebereich
Name	Name der Ereignisdefinition, muss in der Ressource eindeutig sein.	Text, max. 32 Zeichen
Globale Variable	Name der zugewiesenen globalen Variable (eingefügt z. B. durch Drag-and-Drop).	
Datentyp	Datentyp der globalen Variable, nicht änderbar.	Abhängig vom Typ der globalen Variablen.
Event-Quelle	CPU Event: Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch. Auto Event: Wie CPU Event Standardwert: Auto Event	CPU, Auto
HH-Alarmtext	Text, der den Alarmzustand des obersten Grenzwerts benennt.	Text

Spalte	Beschreibung	Wertebereich
HH-Alarmwert	Oberster Grenzwert, der ein Ereignis auslöst. Bedingung: $(HH \text{ Alarm Value} - \text{Hysterese}) > H \text{ Alarm Value}$ oder $HH \text{ Alarm Value} = H \text{ Alarm Value}$	Abhängig vom Typ der globalen Variablen
HH-Alarmpriorität	Priorität des obersten Grenzwerts. Standardwert: 500	0 – 1000
HH-Alarmbestätigung erforderlich	Aktiviert: Bediener muss Überschreitung des obersten Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Überschreitung des obersten Grenzwerts nicht bestätigen. Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
H-Alarmtext	Text, der den Alarmzustand des oberen Grenzwerts benennt.	Text
H-Alarmwert	Oberer Grenzwert, der ein Ereignis auslöst. Bedingung: $(H \text{ Alarm Value} - \text{Hysterese}) > (L \text{ Alarm Value} + \text{Hysterese})$ oder $H \text{ Alarm Value} = L \text{ Alarm Value}$	Abhängig vom Typ der globalen Variablen
H-Alarmpriorität	Priorität des oberen Grenzwerts. Standardwert: 500	0 – 1000
H-Alarmbestätigung erforderlich	Aktiviert: Bediener muss Überschreitung des obersten Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Überschreitung des obersten Grenzwerts nicht bestätigen. Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
Return to Normal Text	Text, der den Alarmzustand benennt.	Text
Return to Normal Severity	Priorität des Normalzustands. Standardwert: 500	0 – 1000
Return to Normal Ack Required	Bestätigung des Normalzustandes durch den Bediener erforderlich (Quittierung) Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
L-Alarmtext	Text, der den Alarmzustand des unteren Grenzwerts benennt.	Text

Spalte	Beschreibung	Wertebereich
L-Alarmwert	Unterer Grenzwert, der ein Ereignis auslöst. Bedingung: $(L \text{ Alarm Value} + \text{Hysterese}) < (H \text{ Alarm Value} - \text{Hysterese})$ oder $L \text{ Alarm Value} = H \text{ Alarm Value}$	Abhängig vom Typ der globalen Variablen
L-Alarmpriorität	Priorität des unteren Grenzwerts. Standardwert: 500	0 – 1000
L-Alarmbestätigung erforderlich	Aktiviert: Bediener muss Unterschreitung des unteren Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Unterschreitung des unteren Grenzwerts nicht bestätigen. Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
LL-Alarmtext	Text, der den Alarmzustand des untersten Grenzwerts benennt.	Text
LL-Alarmwert	Unterster Grenzwert, der ein Ereignis auslöst. Bedingung: $(LL \text{ Alarm Value} + \text{Hysterese}) < (L \text{ Alarm Value})$ oder $LL \text{ Alarm Value} = L \text{ Alarm Value}$	Abhängig vom Typ der globalen Variablen
LL-Alarmpriorität	Priorität des untersten Grenzwerts. Standardwert: 500	0 – 1000
LL-Alarmbestätigung erforderlich	Aktiviert: Bediener muss Unterschreitung des untersten Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Unterschreitung des untersten Grenzwerts nicht bestätigen. Standardwert: Deaktiviert	Kontrollkästchen aktiviert, deaktiviert
Alarm-Hysterese	Die Hysterese verhindert ein ständiges Erzeugen von vielen Ereignissen, wenn der Prozesswert häufig um einen Grenzwert schwankt.	Abhängig vom Typ der globalen Variablen

ACHTUNG



Fehlerhafte Ereignisbildung durch Parametrierungsfehler möglich!

Setzen der Parameter *L-Alarmwert* und *H-Alarmwert* auf denselben Wert kann zu unerwünschtem Verhalten der Ereignisbildung führen, da in diesem Fall kein Normalbereich existiert.

12.6 Umgang mit dem Anwenderprogramm

Der Anwender hat über das Programmiergerät folgende Möglichkeiten, die Funktion seines Programms in der Steuerung zu beeinflussen.

12.6.1 Setzen der Parameter und Schalter

Während der Projektierung eines Anwenderprogramms werden die Parameter und Schalter offline gesetzt und mit dem codegenerierten Programm in die Steuerung geladen. Das Setzen der Parameter und Schalter kann aber auch in den Zuständen STOPP und RUN erfolgen, wenn der Schalter *Hauptfreigabe* gesetzt ist. Nur die Elemente im NVRAM können geändert werden, alle anderen werden beim Laden gesetzt.

12.6.2 Starten des Programms von STOPP/GÜLTIGE KONFIGURATION

Das Starten des Programms entspricht dem Überführen der Steuerung von der Betriebsart STOPP/GÜLTIGE KONFIGURATION in die Betriebsart RUN. Auch das Programm geht in den RUN-Modus. Das Programm geht in den Testmodus, wenn während des Startens der Testmodus aktiviert ist. Nach IEC 61131 ist zusätzlich zum Start im Testmodus auch der Kalt- oder Warmstart möglich.

HINWEIS



Das Starten des Programms ist nur möglich, wenn der Schalter *Start/Neustart erlaubt* gesetzt ist.

12.6.3 Neustart des Programms nach Fehler

Geht das Programm in STOPP/UNGÜLTIGE KONFIGURATION, z. B. durch unerlaubte Zugriffe auf Bereiche des Betriebssystems, startet es neu. Geht es innerhalb von ca. einer Minute nach dem Neustart erneut in den Zustand STOPP/UNGÜLTIGE KONFIGURATION, bleibt es in diesem Zustand. Dann kann es über die Start-Schaltfläche des Control Panel wieder gestartet werden. Nach dem Neustart prüft das Betriebssystem das gesamte Programm.

12.6.4 Stoppen des Programms

Wird das Anwenderprogramm gestoppt, geht die Steuerung von der Betriebsart RUN nach STOPP/GÜLTIGE KONFIGURATION.

12.6.5 Testmodus des Programms

Der Testmodus wird über das Control Panel im Menü [Testmodus] / [Testmodus mit Heissstart] (oder Kaltstart/Warmstart) gestartet. Mit dem Befehl *Zyklussschritt* wird jedes Mal ein Einzelschritt (einmaliger Logikdurchgang) aktiviert.

Verhalten von Variablenwerten im Testmodus:

Die Wahl Kaltstart, Warmstart oder Heißstart legt fest, welche Variablenwerte für den ersten Durchgang im Testmodus verwendet werden.

- Kaltstart: Alle Variablen erhalten ihren Initialwert.
- Warmstart: Retain-Variablen behalten ihren Wert, andere werden auf ihren Initialwert gesetzt
- Heißstart: Alle Variablen behalten ihren aktuellen Wert.

Anschließend kann mit dem Befehl *Zyklusschritt* das Anwenderprogramm im Einzelschrittmodus gestartet werden. Alle aktuellen Werte bleiben für den nächsten Zyklus erhalten (eingefrorener Zustand).



⚠ WARNUNG

Aktoren im nicht sicheren Zustand.

Tod oder schwere Körperverletzungen.

Funktion Testmodus nicht im sicherheitsgerichteten Betrieb verwenden!

13 Betrieb

Dieses Kapitel beschreibt die Bedienung und Diagnose während des Betriebs der Steuerung.

13.1 Bedienung

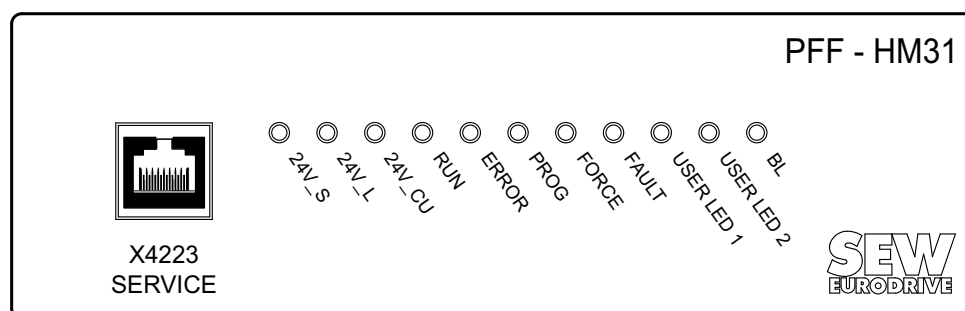
Eine Bedienung der Steuerung ist im normalen Betrieb nicht erforderlich. Nur beim Auftreten von Problemen kann ein Eingreifen mit dem Programmiergerät erforderlich sein.

13.2 Diagnose

Eine erste, grobe Diagnose kann mit Hilfe der Leuchtdiodenanzeigen erfolgen. Eine detailliertere Analyse des Betriebs- oder Fehlerzustands ist mit Hilfe der Diagnosehistorie möglich. Diese ist mit dem Programmiergerät anzeigbar.

13.2.1 LED-Anzeige

Die System-LEDs befinden sich auf der Service-Einheit des Geräts und zeigen die Feldbus- und Gerätestatus an. Zusätzlich existieren 2 vom Anwender frei konfigurierbare User-LEDs:



4867138571

Die folgende Tabelle zeigt den Status und die Bedeutung der LED an:

Bezeichnung	Status LED	Bedeutung
BL	Blinkt rot	<ul style="list-style-type: none"> • BL (Boot-Loader) defekt oder Hardware-Fehler. • Fehler der externen Prozessdaten-Kommunikation • Es wurde eine doppelte IP-Adresse entdeckt¹⁾. • PROFINET hat einen Identify-Request erhalten¹⁾
	Aus	Keines der beschriebenen Ereignisse ist eingetreten.
USER LED 2	Leuchtet rot	Codierung: 1 (siehe Abschnitt „User-LEDs“)
USER LED 1	Blinkt rot	Codierung: 2 (siehe Abschnitt „User-LEDs“)
	Aus	Codierung: 0 oder 3 – 255

Bezeichnung	Status LED	Bedeutung
FAULT	Leuchtet gelb / Blinkt gelb ²⁾	<ul style="list-style-type: none"> Das neue Betriebssystem ist verfälscht (nach dem Herunterladen) Fehler beim Laden eines neuen Betriebssystems Die geladene Konfiguration ist fehlerhaft. Ein oder mehrere E/A-Fehler haben sich ereignet. Es wurde eine doppelte IP-Adresse entdeckt.¹⁾ PROFINET hat einen Identify-Request erhalten¹⁾
	Aus	Keines der beschriebenen Ereignisse ist eingetreten.
FORCE	Leuchtet gelb	Forcen vorbereitet: <ul style="list-style-type: none"> Force-Schalter einer Variablen ist gesetzt, der Force-Hauptschalter ist noch deaktiviert. Das Gerät ist im Zustand RUN oder STOPP.
	Blinkt gelb	Forcen aktiv: Mindestens eine lokale oder globale Variable hat ihren Force-Wert angenommen. <ul style="list-style-type: none"> Es wurde eine doppelte IP-Adresse entdeckt.¹⁾ PROFINET hat einen Identify-Request erhalten¹⁾
	Aus	Keines der beschriebenen Ereignisse ist eingetreten.
PROG	Leuchtet gelb	<ul style="list-style-type: none"> Das Gerät wird mit einer neuen Konfiguration geladen. Ein neues Betriebssystem wird geladen. Änderung der WDZ oder Sicherheitszeit. Prüfung auf doppelte IP-Adresse. Änderung der SRS.
	Blinkt gelb	<ul style="list-style-type: none"> Reload-Funktion wird durchgeführt (Funktion muss über Lizenz freigeschaltet werden) Es wurde eine doppelte IP-Adresse entdeckt.¹⁾ PROFINET hat einen Identify-Request erhalten¹⁾
	Aus	Keines der beschriebenen Ereignisse ist eingetreten.
ERROR	Leuchtet rot / Blinkt rot ²⁾	<ul style="list-style-type: none"> Das Gerät ist im Zustand FEHLERSTOPP: Durch Selbsttest festgestellter interner Fehler, z. B. Hardware-Fehler, Software-Fehler oder Fehler der Spannungsversorgung. Abhilfe: Das Prozessorsystem kann nur durch einen Befehl vom PADT wieder gestartet werden (Reboot). Es werden nicht aktivierte Protokolle/Funktionen verwendet (Warnung). Fehler beim Laden des Betriebssystems
	Aus	Keines der beschriebenen Ereignisse ist eingetreten.

Bezeichnung	Status LED	Bedeutung
RUN	Leuchtet grün	<ul style="list-style-type: none"> Gerät im Zustand RUN, Normalbetrieb Ein geladenes Anwenderprogramm wird ausgeführt
	Blinkt grün	<ul style="list-style-type: none"> Gerät im Zustand STOPP Ein neues Betriebssystem wird geladen
	Aus	Gerät ist nicht im Zustand RUN oder STOPP
24V_CU	Leuchtet grün	Zwischen X1541.1 und X1541.2 liegt 24 V an.
24V_L	Leuchtet grün	Zwischen X1541.3 und X1541.4 liegt 24 V an.
24V_S	Leuchtet grün	Zwischen X2312.1 und X2312.3 liegt 24 V an.

1) Bei gemeinsamem Blinken der LEDs: PROG, FORCE, FAULT und BL

2) Der Status "Leuchtet" signalisiert eine Warnung und "Blinken" signalisiert einen Alarm.

Beim Zuschalten der Versorgungsspannung erfolgt immer ein Test der Leuchtdioden, bei dem für kurze Zeit alle Leuchtdioden leuchten.

User-LEDs

Die beiden frei konfigurierbaren User-LEDs (USER LED 1/2) werden über Systemvariablen angesteuert. Dazu müssen den zugehörigen Systemvariablen globale Variablen vom Datentyp USINT zugewiesen werden.

13.2.2 Diagnosehistorie

Die Diagnosehistorie erfasst die verschiedenen Zustände des Prozessor- und des Kommunikationssystems und legt sie in einem nicht-flüchtigen Speicher ab. Dabei wird für beide zwischen Langzeit- und Kurzzeitdiagnose gemäß folgender Tabelle unterschieden.

	CPU	COM
Einträge in der Langzeitdiagnose	700	300
Einträge in der Kurzzeitdiagnose	700	700

Die Langzeitdiagnose des Prozessorsystems umfasst folgende Ereignisse:

- Reboot
- Wechsel der Betriebsart
(INIT, RUN, STOPP/GÜLTIGE KONFIGURATION, STOPP/UNGÜLTIGE KONFIGURATION)
- Wechsel der Programm-Betriebsart
(START, RUN, FEHLER, TESTMODUS)
- Laden / Löschen einer Konfiguration
- Setzen und Rücksetzen von Schaltern
- Fehler im Prozessorsystem
- Laden eines Betriebssystems
- Forcen (Setzen und Rücksetzen des Schalters Forcen erlaubt)
- Diagnose der Spannungsversorgung und Temperatur

Die Langzeitdiagnose des Kommunikationssystems umfasst folgende Ereignisse:

- Reboot des Kommunikationssystems
- Wechsel der Betriebsart (INIT, RUN, STOPP/GÜLTIGE KONFIGURATION, STOPP/UNGÜLTIGE KONFIGURATION)

- Anmelden von Benutzern
- Laden eines Betriebssystems

Ist der Speicher der Langzeitdiagnose voll, werden alle Daten, die älter als drei Tage sind, gelöscht, und es können neue Einträge aufgenommen werden. Sind alle Daten weniger als drei Tage alt, können keine neuen Daten gespeichert werden und sind verloren. Ein Eintrag in der Langzeitdiagnose zeigt an, dass Daten nicht gespeichert werden konnten.

Die Kurzzeitdiagnose des Prozessorsystems umfasst folgende Ereignisse:

- Diagnose des Prozessorsystems (Setzen der Force-Schalter und Force-Werte)
- Diagnose des Anwenderprogramms (Zyklusbetrieb)
- Diagnose der Kommunikation
- Diagnose der Spannungsversorgung und der Temperatur

Die Kurzzeitdiagnose des Kommunikationssystems umfasst folgende Ereignisse:

- safeethernet-bezogene Ereignisse
- Start / Stopp beim Schreiben des Flash-Speichers
- Fehler, die beim Laden einer Konfiguration aus dem Flash-Speicher auftreten können
- Auseinandergelaufene Zeitsynchronisation zwischen Kommunikationssystem und Prozessorsystem

Parametrierungsfehler der Eingänge und Ausgänge werden bei der Codegenerierung u. U. nicht erkannt. Im Rückmeldefenster der Diagnose erscheint bei einem Parametrierungsfehler die Meldung INVALID CONFIG mit Angabe der Fehlerquelle und eines Fehlercodes. Diese Meldung hilft bei der Analyse von Fehlern bei der Parametrierung der Ein- und Ausgänge.

Ist der Speicher der Kurzzeitdiagnose voll, werden die jeweils ältesten Einträge entfernt, um Platz für neue Einträge zu schaffen. Es erfolgt keine Anzeige, wenn alte Einträge gelöscht werden.

Die Aufzeichnung der Diagnosedaten ist nicht sicherheitsgerichtet. Die in chronologischer Reihenfolge aufgezeichneten Daten können über das Programmierwerkzeug für eine Analyse ausgelesen werden. Das Auslesen löscht nicht die Daten in der Steuerung. Das Programmierwerkzeug kann den Inhalt des Diagnosefensters abspeichern.

13.2.3 Diagnose in SILworX®

Der Zugang zur Diagnose erfolgt über die Online-Ansicht des Hardware-Editors in SILworX®.

Gehen Sie folgendermaßen vor, um die Diagnose zu öffnen:

1. Unter der gewünschten Ressource den Zweig "Hardware" markieren.
2. Im Kontextmenü oder in der Aktionsleiste [Online] anklicken. Das Systemlogin-Fenster öffnet sich.

Ins Systemlogin-Fenster die folgenden Informationen auswählen oder eingeben:

- IP-Adresse der Steuerung
- Benutzer und Passwort

Die Online-Ansicht des Hardware-Editors öffnet sich.

4. In der Online-Ansicht das gewünschte Modul auswählen, normalerweise das Prozessor- oder das Kommunikationsmodul.
5. Aus dem Kontextmenü oder dem Menü [Online] den Punkt [Diagnose] auswählen.

Die Diagnose für das betreffende Modul öffnet sich.

Bei laufender Steuerung erscheinen Meldungen über Zustände des Prozessorsystems, des Kommunikationssystems und der E/A-Baugruppen über bestimmte, einstellbare Zeiträume.

13.3 Parameter und Fehlercodes der Ein- und Ausgänge

In den folgenden Übersichten sind die lesbaren und einstellbaren Systemparameter der Ein-/Ausgänge einschließlich der Fehlercodes aufgeführt. Die Fehlercodes können innerhalb des Anwenderprogramms über die entsprechenden, in der Logik zugewiesenen Variablen ausgelesen werden. Die Anzeige der Fehlercodes kann auch in SILworX® erfolgen.

13.3.1 Digitale Eingänge MOVISAFE® HM31

Die nachfolgenden Tabellen enthalten die Status und Parameter des Eingangsmoduls (DI 26) in derselben Reihenfolge wie im Hardware-Editor.

Register-Modul

Das Register-Modul enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung	
DI AnzahlTakt-speisekanäle	USINT	W	Anzahl der Taktausgänge (Speiseausgänge).	
			Codierung	Beschreibung
			0	Kein Taktausgang für LS/LB-Erkennung vorgesehen. ¹⁾
			1	Taktausgang 1 für LS/LB-Erkennung vorgesehen. ¹⁾
			2	Taktausgänge 1 und 2 für LS/LB-Erkennung vorgesehen. ¹⁾
		
			6	Taktausgänge 1 bis 6 für LS/LB-Erkennung vorgesehen. ¹⁾
			Taktausgänge dürfen nicht als sicherheitsgerichtete Ausgänge verwendet werden!	
DI Speisung [01]	BOOL	W	Ansteuerung des Versorgungsausgangs. TRUE: Speisung ist eingeschaltet FALSE: Speisung ist nicht eingeschaltet	
DI Steckpl. Takt-speise-Bg	UDINT	W	Steckplatz der Taktspeisebaugruppe (LS/LB-Erkennung), Wert auf "2" einstellen. ¹⁾	
DI Taktverzögerung [µs]	UINT	W	Wartezeit für Line Control (Schluss- / Querschluss-erkennung).	

Systemparameter	Datentyp	R/W	Beschreibung	
DI.Fehlercode	WORD	R	Fehlercodes aller digitalen Eingänge.	
			Codierung	Beschreibung
			0x0001	Fehler der Baugruppe.
			0x0002	FTZ-Test des Testmusters fehlerhaft.
			0x2000	Parametrierung der LS/LB-Erkennung fehlerhaft. ¹⁾
DI.Fehlercode Speisung	WORD	R	Baugruppen-Fehlercodes des Versorgungsausgangs.	
			Codierung	Beschreibung
			0x0001	Fehler der Baugruppe.
DI[01].Fehlercode Speisung	BYTE	R	Kanal-Fehlercodes des Versorgungsausgangs.	
			Codierung	Beschreibung
			0x01	Fehler DI Speiseeinheit.
			0x02	Speisung wegen Überstrom abgeschaltet.
			0x04	Fehler beim Rücklesen der Speisung.
DO.Fehlercode	WORD	R	Fehlercodes aller Taktausgänge.	
			Codierung	Beschreibung
			0x0001	Fehler der Baugruppe.
ModulFehlercode	WORD	R	Fehlercodes des Moduls.	
			Codierung	Beschreibung
			0x0000	E/A-Verarbeitung, ggf. mit Fehlern, siehe weitere Fehlercodes.
			0x0001	Keine E/A-Verarbeitung (CPU nicht in RUN).
			0x0002	Keine E/A-Verarbeitung während des Hochfahrttests.
			0x0004	Hersteller-Interface in Betrieb.
			0x0010	Keine E/A-Verarbeitung: falsche Parametrierung.
			0x0020	Keine E/A-Verarbeitung: Fehlerrate überschritten.
			0x0040 / 0x0080	Keine E/A-Verarbeitung: konfiguriertes Modul nicht gesteckt.
ModulSRS	UDINT	R	Steckplatz-Nummer (System-Rack-Slot).	

Systemparameter	Datentyp	R/W	Beschreibung
ModulTyp	UINT	R	Typ des Moduls, Sollwert: 0x001A [26 _{dez}].

1) LS = Leitungsschluss / LB = Leitungsbruch (Line Control)

Register DI26: DI-Kanäle

Das Register DI 26: DI-Kanäle enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung	
Kanal-Nr.	-	R	Kanalnummer, fest vorgegeben.	
→ Fehlercode [BYTE]	BYTE	R	Fehlercodes der digitalen Eingangskanäle.	
			Codierung	Beschreibung
			0x01	Fehler im digitalen Eingangsmodul.
			0x10	Leitungsschluss des Kanals.
			0x80	Unterbrechung zwischen Taktausgang (DO-Kanal) und digitalem Eingang DI, z. B. <ul style="list-style-type: none">Leitungsbruchgeöffneter SchalterL+ Unterspannung (+24 V_CU)
→ Wert [BOOL]	BOOL	R	Eingangswert der digitalen Eingänge. 0 = Eingang nicht angesteuert 1 = Eingang angesteuert	
Taktspeisekanal [USINT] →	USINT	W	Quellkanal der Taktspeisung.	
			Codierung	Beschreibung
			0	Eingangskanal.
			1	Takt vom 1. DO-Kanal.
			2	Takt vom 2. DO-Kanal.
		
6	Takt vom 6. DO-Kanal.			

Register DI26: DO-Kanäle

Das Register DI 26: DO-Kanäle enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung
Kanal-Nr.	-	R	Kanalnummer, fest vorgegeben.

Systemparameter	Datentyp	R/W	Beschreibung	
→ Fehlercode [BYTE]	BYTE	R	Fehlercodes der digitalen Ausgänge.	
			Codierung	Beschreibung
			0x01	Fehler im digitalen Ausgangsmodul oder der Baugruppe.
→ Wert [BOOL]	BOOL	W	Ausgabewert der DO-Kanäle.	
			Codierung	Beschreibung
			0	Ausgang stromlos.
			1	Ausgang angesteuert.

13.3.2 Digitale Ausgänge MOVISAFE® HM31

Die nachfolgenden Tabellen enthalten die Status und Parameter des Ausgangsmoduls (DO 8) in derselben Reihenfolge wie im Hardware-Editor.

Register Modul

Das Register Modul enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung	
DO. Fehlercode	WORD	R	Fehlercodes aller digitalen Ausgänge.	
			Codierung	Beschreibung
			0x0001	Fehler der Baugruppe.
			0x0002	MEZ-Test der Sicherheitsabschaltung liefert einen Fehler.
			0x0004	MEZ-Test Hilfsspannung liefert einen Fehler.
			0x0008	FTZ-Test des Testmusters fehlerhaft.
			0x0010	MEZ-Test des Testmusters der Ausgangsschalter fehlerhaft.
			0x0020	MEZ-Test des Testmusters der Ausgangsschalter (Abschalttest der Ausgänge) fehlerhaft.
			0x0040	MEZ-Test Aktive Abschaltung über WD fehlerhaft.
			0x0400	FTZ-Test: 1. Temperaturschwelle überschritten.
			0x0800	FTZ-Test: 2. Temperaturschwelle überschritten.
			0x4000	Parametrierung der 2-poligen Überwachung fehlerhaft.

Systemparameter	Datentyp	R/W	Beschreibung
Einschaltverzögerung	UINT	W	Einschaltverzögerung für 2-polige Tests, wegen Leitungskapazitäten, induktiver und kapazitiver Last. Wertebereich: 0 – 30 ms
ModulFehlercode	WORD	R	Fehlercodes des Moduls.
			Codierung
			Beschreibung
			0x0000 E/A-Verarbeitung, ggf.. mit Fehlern, siehe weitere Fehlercodes.
			0x0001 Keine E/A-Verarbeitung (CPU nicht in RUN).
			0x0002 Keine E/A-Verarbeitung während des Hochfahrtests.
			0x0004 Hersteller-Interface in Betrieb.
			0x0010 Keine E/A-Verarbeitung: falsche Parametrierung.
			0x0020 Keine E/A-Verarbeitung: Fehlerrate überschritten.
			0x0040/0x0080 Keine E/A-Verarbeitung: konfiguriertes Modul nicht gesteckt.
ModulSRS	UDINT	R	Steckplatz-Nummer (System-Rack-Slot).
ModulTyp	UINT	R	Typ des Moduls, Sollwert: 0x0029 [41 _{dez}].

Register DO 8: Kanäle

Das Register DO 8: Kanäle enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung
Kanal-Nr.	-	R	Kanalnummer, fest vorgegeben.

Systemparameter	Datentyp	R/W	Beschreibung	
→ + Fehlercode [BYTE]	WORD	R	Fehlercodes der digitalen Ausgänge.	
			Codierung	Beschreibung
			0x0001	Fehler im digitalen Ausgangsmodul.
			0x0002	Ausgang abgeschaltet wegen Überstrom.
			0x0004	Fehler beim Rücklesen der Ansteuerung der digitalen Ausgänge.
			0x0008	Fehler beim Rücklesen des Status der digitalen Ausgänge.
			0x0020	Externer Leitungsschluss oder Schluss des EMV-Schutzes liefert einen Fehler (L + Schluss (24V_L)).
			0x0040	Externer Leitungsschluss oder Schluss des EMV-Schutzes liefert einen Fehler (L – Schluss (0V24)).
			0x0080	Kanal ist wegen Fehler des zugeordneten DO-Kanals abgeschaltet.
			0x0100	Test Schluss des Ausgangs gegen L+ (24V_L) aufgrund von Sollwertänderungen oder Unterspannung nicht durchgeführt.
			0x0200	Test Schluss des Ausgangs gegen L– (0V24) aufgrund von Sollwertänderungen oder Unterspannung nicht durchgeführt.
			0x0400	Alle Kanäle abgeschaltet, Gesamtstrom überschritten.
			0x0800	FTZ-Test: Überwachung der Hilfsspannung 1: Unterspannung.
→ – Fehlercode [BYTE]	WORD	R	Siehe -> + Fehlercode [BYTE]	
→ Wert [BOOL]	BOOL	W	Ausgangswert der DO-Kanäle. 0 = Ausgang stromlos 1 = Ausgang angesteuert	
2-polig abgeschaltet [BOOL] →	BOOL	W	Parametrierung, ob der Kanal 2-polig verwendet wird. 0 = Kanal wird 2-polig verwendet 1 = Kanal wird 1-polig verwendet	

13.3.3 Zähler MOVISAFE® HM31

Die nachfolgenden Tabellen enthalten die Status und Parameter des Zählermoduls (HSC 2) in derselben Reihenfolge wie im Hardware-Editor.

Register Modul

Das Register Modul enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung	
ModulFehlercode	WORD	R	Fehlercodes des Moduls.	
			Codierung	Beschreibung
			0x0000	E/A-Verarbeitung, ggf. mit Fehlern, siehe weitere Fehlercodes.
			0x0001	Keine E/A-Verarbeitung (CPU nicht in RUN).
			0x0002	Keine E/A-Verarbeitung während des Hochfahrttests.
			0x0004	Hersteller-Interface in Betrieb.
			0x0010	Keine E/A-Verarbeitung: falsche Parametrierung.
			0x0020	Keine E/A-Verarbeitung: Fehlerrate überschritten.
			0x0040/ 0x0080	Keine E/A-Verarbeitung: konfiguriertes Modul nicht gesteckt.
ModulSRS	UDINT	R	Steckplatz-Nummer (System-Rack-Slot).	
ModulTyp	UINT	R	Typ des Moduls, Sollwert: 0x0003 [3 _{dez}].	
Zähler.Fehlercode	WORD	R	Fehlercodes des Zählermoduls.	
			Codierung	Beschreibung
			0x0001	Fehler der Baugruppe.
			0x0002	Fehler beim Vergleich der Zeitbasis.
			0x0004	Adressfehler beim Lesen der Zeitbasis.
			0x0008	Parameter für die Zeitbasis fehlerhaft.
			0x0010	Adressfehler beim Lesen des Zählerstands.
			0x0020	Parametrierung des Zählers wurde verfälscht.
			0x0040	Adressfehler beim Lesen des Gray-Codes.
			0x0080	FTZ-Test des Testmusters fehlerhaft.
			0x0100	FTZ-Test Fehler bei Überprüfung der Koeffizienten
			0x0200	Fehler bei der initialen Parametrierung der Baugruppe.

Register HSC 2: Kanäle

Das Register HSC 2: Kanäle enthält die folgenden Systemparameter.

Systemparameter	Datentyp	R/W	Beschreibung	
Zähler[0x].5/24V Modus	BOOL	R/W	Zählereingang 5 V oder 24 V. TRUE: 24 V FALSE: 5 V	
Zähler [0x]Autom. Drehrichtungserkennung	BOOL	R/W	Automatische Drehrichtungserkennung. TRUE: Automatische Drehrichtungserkennung EIN FALSE: Manuelles Setzen der Drehrichtung	
Zähler[0x].Fehlercode	BYTE	R	Fehlercodes der Zählerkanäle.	
			Codierung	Beschreibung
			0x01	Fehler im Zählermodul.
			0x02	Fehler beim Vergleich der Zählerstände.
			0x08	Fehler beim Einstellen der Parametrierung (Reset).
Zähler[0x].Gray-Code	BOOL	R/W	Decoder/Impulsbetrieb. TRUE: Gray-Code Decoder FALSE: Impulsbetrieb Decoder-Betrieb nicht zulässig!	
Zähler[0x].Leer1	BOOL	R/W	Keine Funktion.	
Zähler[0x].Leer2	BOOL	R/W		
Zähler[0x].Leer3	BOOL	R/W		
Zähler[0x].Reset	BOOL	R/W	Reset des Zählkanals (nur wenn <i>Zähler[0x].Autom. Drehrichtungserkennung = FALSE</i>) TRUE: kein Reset FALSE: Reset	
Zähler[0x].Richtung	BOOL	R/W	Zählrichtung des Zählers (nur wenn <i>Zähler[0x].Autom. Drehrichtungserkennung = FALSE</i>). TRUE: Abwärts (Dekrementieren) FALSE: Aufwärts (Inkrementieren)	
Zähler[0x].Wert	UDINT	R	Zählerstand der Zähler: 24 Bit für Impulszähler.	
Zähler[0x].Wert-Überlauf	BOOL	R	Zähler-Überlaufanzeige. TRUE: Überlauf seit letztem Zyklus (nur wenn <i>Zähler[0x].Autom. Drehrichtungserkennung = FALSE</i>) FALSE: Kein Überlauf seit letztem Zyklus	
Zähler[0x].Zeitstempel	UDINT	R	Zeitstempel für <i>Zähler[0x].Wert</i> , 24 Bit, Zeitauflösung 1 µs.	

Systemparameter	Datentyp	R/W	Beschreibung
Zähler[0x].Zeit-Überlauf	BOOL	R	Überlaufanzeige für den Zeitstempel der Zähler. TRUE: 24-Bit-Überlauf seit letztem Zyklus FALSE: Kein Überlauf seit letztem Zyklus

14 Service

14.1 Inspektion / Wartung

Die Sicherheitssteuerung ist wartungsfrei. SEW-EURODRIVE legt keine regelmäßigen Inspektionsarbeiten fest, empfiehlt aber eine regelmäßige Überprüfung der folgenden Komponenten:

- Anschlusskabel:

Falls Beschädigungen oder Ermüdungserscheinungen auftreten, müssen die beschädigten Kabel umgehend ausgetauscht werden.

HINWEIS



Reparaturen führt nur SEW-EURODRIVE aus!

14.2 Gerätetausch

14.2.1 Voraussetzungen

Zum Tausch einer Sicherheitssteuerung MOVISAFE® HM31 müssen die folgenden Voraussetzungen erfüllt sein:

- Alle Verbindungen zu Ein- und Ausgängen müssen getrennt sein.
- Sämtliche Kommunikationsverbindungen müssen getrennt sein.
- Die Sicherheitssteuerung darf nicht extern beschaltet sein.
- Die Dokumentation / Beschreibung der elektrischen Anschlüsse muss vorhanden sein.
- Es muss ein Programmiergerät (PC/Notebook) mit installierter Software SILworX® (Version 4.116, SILworX®-Vollversion) und ein USB-Dongle (Hardlock inkl. SEW-Option) vorhanden sein.
- Es muss das dazugehörige SILworX®-Projekt vorhanden sein.
- In SILworX® muss das fehlerlos kompilierte Projekt (zweimalige Codegenerierung) geöffnet sein.
- Es muss ein Ethernet-Kabel vorhanden sein.
- Mitarbeiter von SEW-EURODRIVE verwenden zu Dokumentationszwecken während des Gerätetauschs das entsprechende Formular.

HINWEIS



Das Hochladen eines kompilierten Anwenderprogramms von der bisherigen Steuerung auf ein Programming and Debugging Tool (PADT) und das anschließende Herunterladen auf die neue Steuerung ist nicht möglich.

14.2.2 Verbindung zur Sicherheitssteuerung

Verbinden Sie das Programmiergerät mit der Sicherheitssteuerung (X4223: Ethernet-Service-Schnittstelle).

Login

Ein Login erfordert die Eingabe der passenden IP-Zieladresse. Für den Systembetrieb hingegen ist diese nicht wichtig. Die IP-Adresse eines Moduls wird in einem nicht-flüchtigen Speicher im Modul abgelegt.

Die IP-Adresse wird nach folgenden Prioritäten gewählt:

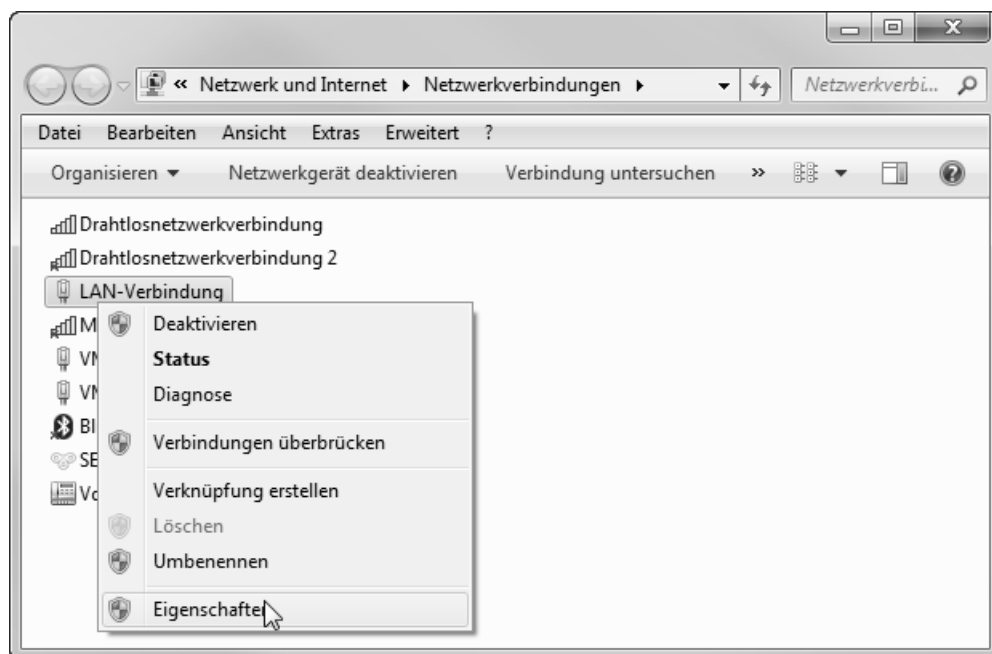
- Wenn eine gültige SILworX®-Konfiguration geladen ist, werden die IP-Adressen aus der Konfiguration übernommen.
- Wenn keine gültige Konfiguration vorhanden ist, wird die letzte gültige IP-Adresse des Moduls verwendet. Dies ist zu beachten, wenn Sicherheitssteuerungen verwendet werden, die bereits an anderer Stelle im Einsatz waren.
- Werkseinstellung MOVISAFE® HM31:
 - Standard IP-Adresse des CPU-Moduls: 192.168.0.99
 - Standard IP-Adresse des COM-Moduls: 192.168.0.100
 - Subnetzmaske: 255.255.252.0
 - Standard System-ID: 60000
- Zur eindeutigen Bestimmung der aktuellen IP-Adresse eines Moduls wird empfohlen, die IP-Adresse mit Hilfe des Dialogs "Suchen per MAC" auszulesen und für den ersten Login zu verwenden.
- Die IP-Adresse des Programmiergeräts muss passend zur Subnetzmaske im gleichen Netz liegen wie die IP-Adresse des zu verbindenden Moduls. Eventuell muss die IP-Adresse des PCs korrigiert werden.

Einstellen der IP-Adresse des Programmiergeräts

Im folgenden Beispiel wird die Einstellung der IP-Adresse des Programmiergeräts beschrieben. Sollte das Programmiergerät über mehrere Netzwerkkarten verfügen, achten Sie darauf, die richtige Netzwerkkarte für die Anwendung auszuwählen.

Gehen Sie so vor:

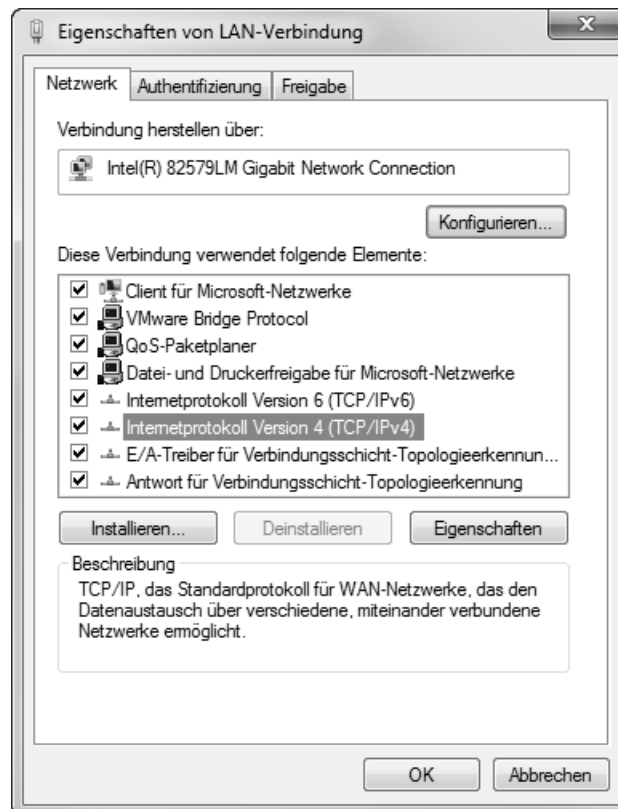
1. Wählen Sie im Kontextmenü der Netzwerkkarte den Menüpunkt [Eigenschaften].



12085296395

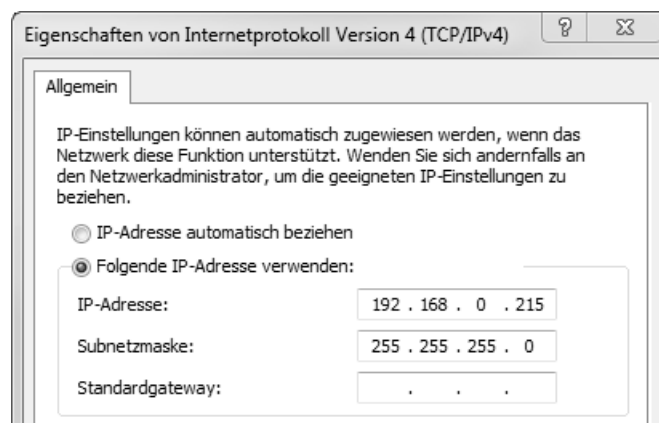
21233780 / DE – 07/2014

2. Markieren Sie im Register „Netzwerk“ das Element „Internetprotokoll Version 4“ und klicken Sie dann auf die Schaltfläche [Eigenschaften].



12085301387

3. Das Fenster „Eigenschaften von Internetprotokoll Version 4“ wird aufgerufen. Markieren Sie die Option „Folgende IP-Adresse verwenden“ und geben Sie die in Ihrem Projekt benötigte IP-Adresse und die Subnetzmaske ein.



12085359243

4. Bestätigen Sie Ihre Eingaben mit [OK]. Damit ist die IP-Adresse des Programmiergeräts eingestellt.

14.2.3 Verifizierung der Systemdaten

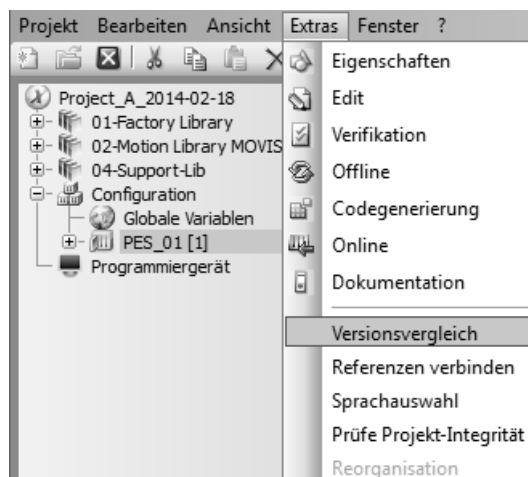
Sollte die Sicherheitssteuerung noch funktionsfähig sein, muss zur Dokumentation und zur Nachweisführung für die Betriebsgenehmigung (Abnahme) der aktuelle auf der Sicherheitssteuerung laufende Konfigurations-CRC sowie der Parameter "Konfigurationspfad im FS" verglichen und dokumentiert werden.

Dazu muss zusätzlich für die entsprechende Ressource ein Versionsvergleich der zuletzt geladenen Version durchgeführt werden. Dies stellt sicher, dass die Dokumentation die aktuellen CRCs der Codegenerierung enthält.

Versionsvergleich

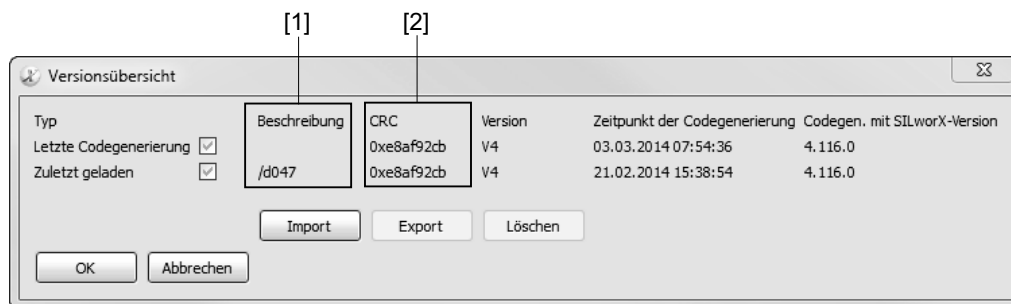
Um einen Versionsvergleich für eine Ressource durchzuführen, gehen Sie in dieser Reihenfolge vor.

1. Markieren Sie im Strukturbaum die entsprechende Ressource (z. B. „PES_01“). Wählen Sie im Menü [Extras] den Menüpunkt [Versionsvergleich].



12085364235

2. Das Fenster „Versionsübersicht“ wird aufgerufen. Die beiden CRCs [2] für „Letzte Codegenerierung“ und „Zuletzt geladen“ müssen identisch sein. Diese CRCs müssen auch mit dem CRC aus der Projektdokumentation übereinstimmen. Zusätzlich muss unter „Zuletzt geladen“ der Parameter „Beschreibung“ [1] verglichen werden.



12085370123

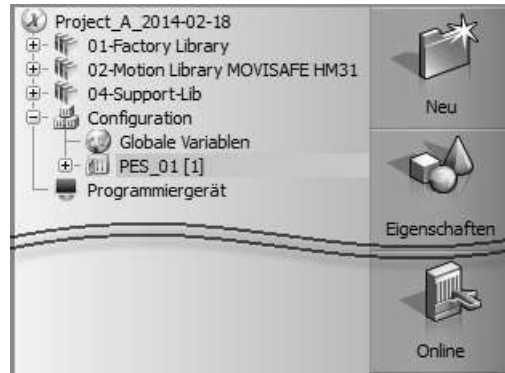
3. Damit ist der Versionsvergleich der Ressource beendet.

Control Panel (Online)

Bei einer noch lauffähigen Sicherheitssteuerung müssen die Parameter „CRC“ und „Beschreibung“ aus dem Versionsvergleich mit den dazugehörigen Parametern in den Systemdaten (Online) übereinstimmen.

Um einen System-Login durchzuführen gehen Sie so vor:

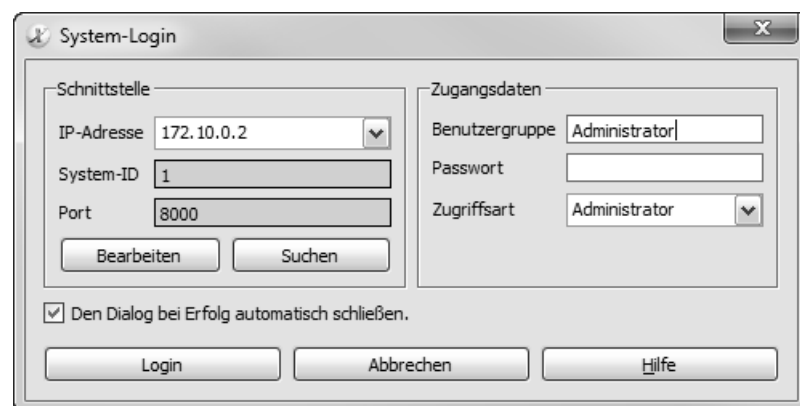
1. Markieren Sie im Strukturbaum die entsprechende Ressource (z. B. „PES_01“) und klicken danach in der Aktionsleiste auf die Schaltfläche [Online]. Das Fenster „System-Login“ wird aufgerufen.



12085478283

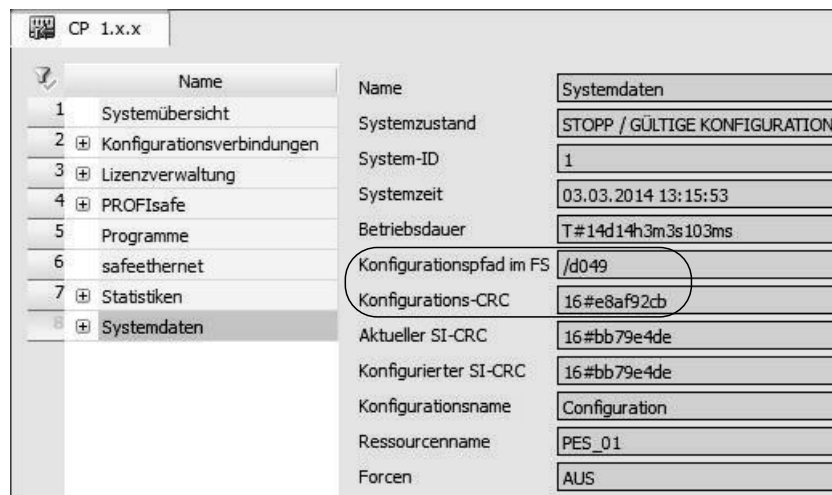
Prüfen Sie in der Gruppe „Schnittstelle“ ob in der Auswahlliste „IP-Adresse“ die richtige IP-Adresse angezeigt wird. Geben Sie in der Gruppe „Zugangsdaten“ zur Autorisierung die Daten der Standardbenutzergruppe folgendermaßen ein:

- Klicken Sie in das Feld „Benutzergruppe“ und drücken Sie die Tastenkombination <Strg + A>.
- Die Einträge in den Feldern „Benutzergruppe“ und „Zugriffsart“ werden automatisch ausgefüllt.



12085484171

3. Klicken Sie auf die Schaltfläche [Login]. Das Control Panel der Ressource wird aufgerufen.
4. Unter „Systemdaten“ finden Sie die Parameter „Konfigurationspfad im FS“ und „Konfigurations-CRC“.



12085629579

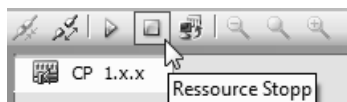
5. Diese Parameterwerte müssen mit den Werten aus dem Versionsvergleich (siehe Kapitel „Versionsvergleich“) und der Projektdokumentation übereinstimmen. Der Parameter „Konfigurationspfad im FS“ entspricht dabei dem Parameter „Beschreibung“ im Versionsvergleich.

Ressource stoppen

Bevor die Sicherheitssteuerung getauscht werden kann, muss sie in den Systemzustand „STOPP“ gebracht werden. Hintergrund dabei ist, dass z. B. bei der Fehlersuche bei einem defektem Ausgang die Diagnoseeinträge der Ressource überschrieben werden.

Gehen Sie so vor, um die Ressource zu stoppen:

1. Klicken Sie in Symbolleiste auf das Symbol „Ressource Stopp“.



12085635339

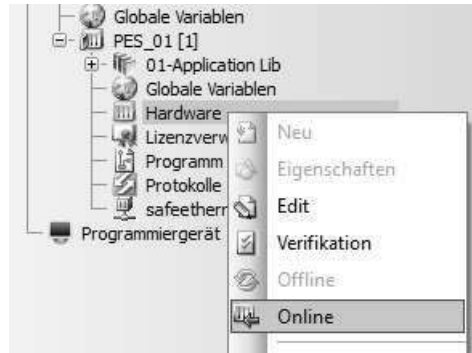
2. Schalten Sie die Spannungsversorgung der Sicherheitssteuerung ab und tauschen Sie die Steuerung aus.

14.2.4 Diagnosedaten sichern (CPU und COM)

Sollte die Sicherheitssteuerung noch funktionsfähig sein, müssen für nachfolgende Analysen die Diagnoseeinträge des COM-Moduls und CPU-Moduls gesichert werden.

Gehen Sie so vor:

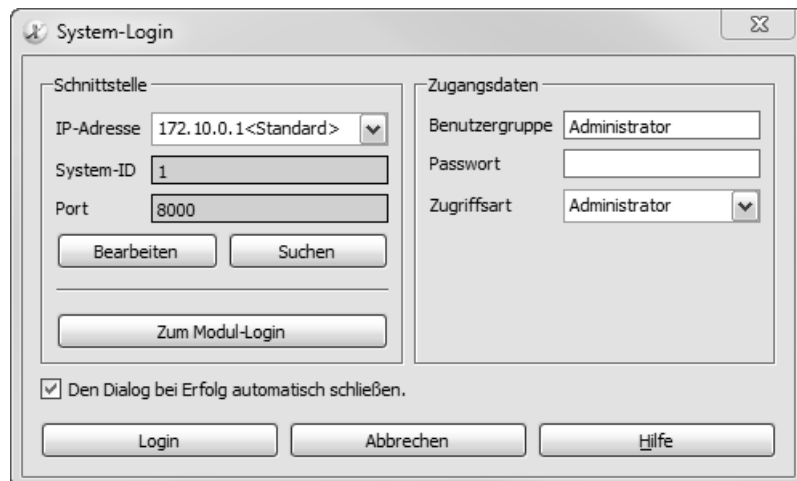
1. Markieren Sie im Strukturbaum die Hardware. Klicken Sie danach in der Aktionsleiste auf die Schaltfläche [Online] oder rufen Sie im Kontextmenü den Menüpunkt [Online] auf. Das Fenster „System-Login“ wird aufgerufen.



12085999627

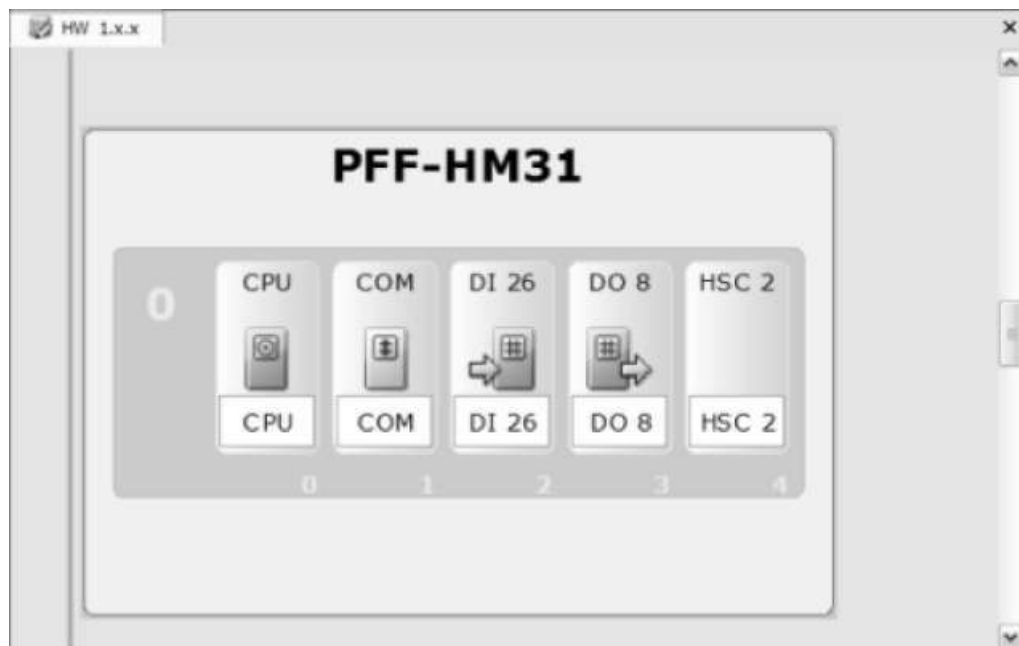
Prüfen Sie in der Gruppe „Schnittstelle“ ob in der Auswahlliste „IP-Adresse“ die richtige IP-Adresse angezeigt wird. Geben Sie in der Gruppe „Zugangsdaten“ zur Autorisierung die Daten der Standardbenutzergruppe folgendermaßen ein:

- Klicken Sie in das Feld „Benutzergruppe“ und drücken Sie die Tastenkombination <Strg + A>.
- Die Einträge in den Feldern „Benutzergruppe“ und „Zugriffsart“ werden automatisch ausgefüllt.



12086004235

3. Klicken Sie auf die Schaltfläche [Login]. Die Hardware-Ansicht der Ressource wird aufgerufen.



12086008843

4. Klicken Sie in der Online-Ansicht des Hardware-Editors mit der rechten Maustaste auf ein Modulsymbol. Wählen Sie den Menüpunkt [Diagnose] aus dem Kontextmenü. Die Diagnoseanzeige wird geöffnet.

Beachten Sie folgende Hinweise:

- **Module mit Warnungen** werden in **gelber Farbe** angezeigt
 - **Module mit Fehlern oder Störungen** werden in **roter Farbe** angezeigt
5. Wählen Sie [Alle Einträge], wenn Sie den kompletten Inhalt des Diagnosespeichers anzeigen möchten.

Wählen Sie [Einträge seit], und ändern Sie Datum und Uhrzeit, um nur Einträge ab einem bestimmten Datum anzuzeigen.

Wenn Sie den Diagnosespeicher zur weiteren Auswertung in einer Datei speichern wollen, gehen Sie folgendermaßen vor:

- Klicken Sie mit der rechten Maustaste in die Liste und rufen Sie im Kontextmenü den Menüpunkt [Speichern] auf.

Die Daten werden als lesbare XML-Datei zusammen mit einigen Basisdaten des Moduls gespeichert.

- Speichern Sie die Diagnosedatei unter einem eindeutigen Dateinamen und senden Sie die Datei bei Bedarf an SEW-EURODRIVE.

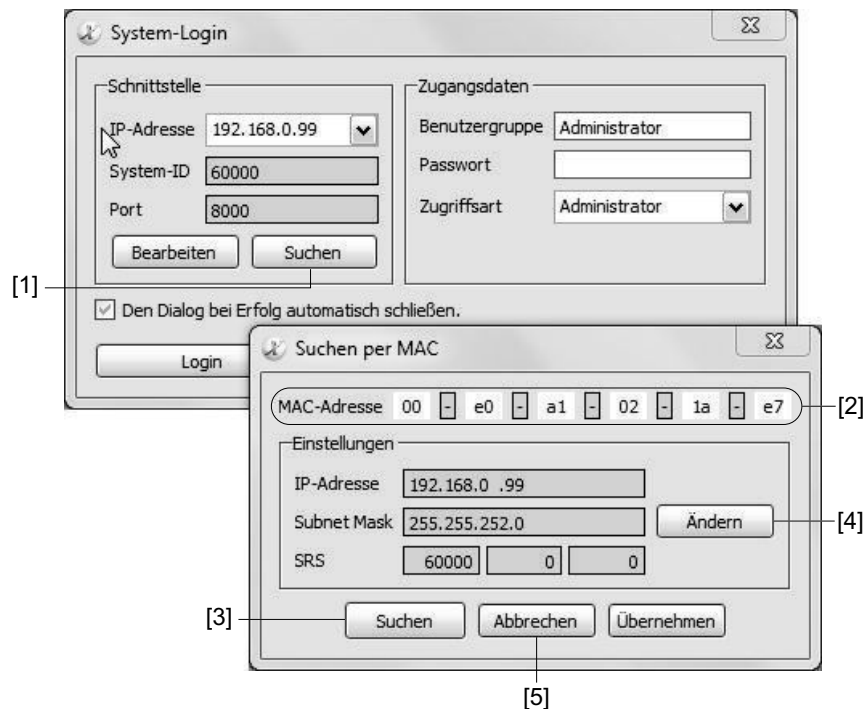
14.2.5 Inbetriebnahme MOVISAFE® HM31 mit Werkseinstellungen

Zur Inbetriebnahme einer Sicherheitssteuerung MOVISAFE® HM31 mit Werkseinstellungen gehen Sie folgendermaßen vor:

1. Trennen Sie alle Kommunikationsverbindungen sowie alle Verbindungen zu Ein- und Ausgängen. Die Sicherheitssteuerung darf nicht extern beschaltet sein.
2. Schalten Sie die Spannungsversorgung ein und warten Sie, bis die Initialisierung beendet ist (LED RUN blinkt). Verbinden Sie das Programmiergerät über ein Ethernet-Kabel mit der Sicherheitssteuerung.
3. Starten Sie SILworX® und öffnen Sie Ihr Projekt.

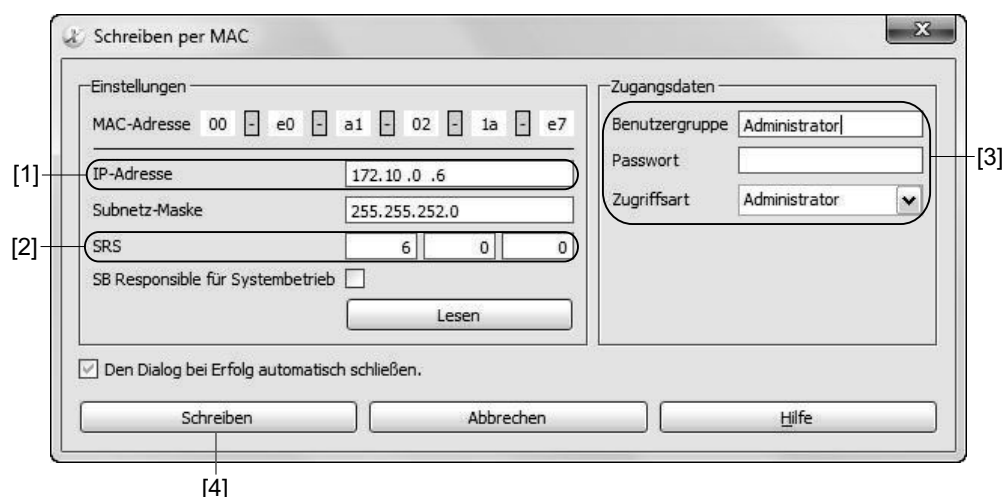
21233780 / DE – 07/2014

4. Markieren Sie im Strukturbaum den Ressourcennamen und klicken Sie in der Aktionsleiste auf „Online“. Das Dialogfenster „System-Login“ öffnet sich.
5. Klicken Sie in der Gruppe „Schnittstelle“ auf die Schaltfläche [Suchen] [1]. Das Fenster „Suchen per MAC“ wird aufgerufen.
6. Geben Sie im Feld „MAC-Adresse“ [2] die MAC-Adresse der CPU ein. Sie finden die MAC-Adresse auf einem Aufkleber auf der Sicherheitssteuerung.
7. Klicken Sie auf die Schaltfläche [Suchen] [3]. Die Daten für „IP-Adresse“, „Subnet Mask“ und „SRS“ werden ausgelesen und in der Gruppe „Einstellungen“ angezeigt.



12086117899

8. Klicken Sie auf die Schaltfläche [Ändern] [4]. Das Fenster „Schreiben per MAC“ öffnet sich (siehe folgendes Bild).



12086382091

9. Verwenden Sie hier die System-ID und die IP-Adresse aus dem System-Login-Dia-log. Geben Sie die System-ID (hier: 6) im Feld „SRS“ [2] und die IP-Adresse [1] (hier: 172.10.0.6) ein.

10. Geben Sie in der Gruppe "Zugangsdaten" [3] zur Autorisierung die Daten der Standardbenutzergruppe ein. Klicken Sie dazu in das Feld "Benutzergruppe" und drücken Sie die Tastenkombination [Strg+A]. Die Einträge in den Feldern „Benutzergruppe“ und „Zugriffsart“ werden automatisch ausgefüllt.
11. Klicken Sie auf die Schaltfläche [Schreiben] [4].
12. Anschließend schließen Sie das Fenster "Suchen per MAC" mit [Abbrechen] [5]. Sie können nun ein Programm in die Steuerung laden und starten.

14.2.6 Inbetriebnahme MOVISAFE® HM31 ohne Werkseinstellungen

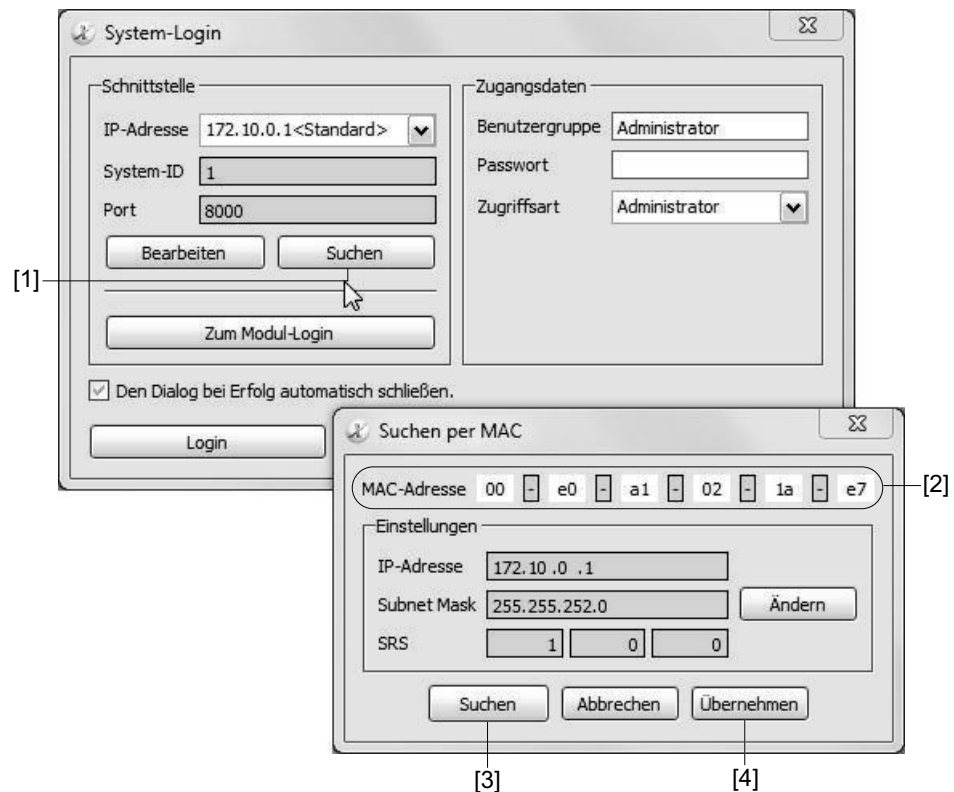
Zur Inbetriebnahme einer Sicherheitssteuerung MOVISAFE® HM31 ohne Werkseinstellungen gehen Sie folgendermaßen vor:

1. Trennen Sie alle Kommunikationsverbindungen sowie alle Verbindungen zu Ein- und Ausgängen. Die Sicherheitssteuerung darf nicht extern beschaltet sein.
2. Schalten Sie die Spannungsversorgung ein und warten Sie, bis die Initialisierung beendet ist (LED RUN blinkt). Verbinden Sie das Programmiergerät über ein Ethernet-Kabel mit der Sicherheitssteuerung.
3. Starten Sie SILworX® und öffnen Sie Ihr Projekt.
4. Markieren Sie im Strukturbaum den Ressourcenamen und klicken Sie in der Aktionsleiste auf „Online“. Das Dialogfenster "System-Login" öffnet sich und zeigt die Ethernet-Parameter gemäß den Projekteinstellungen an.

Ethernet-Parameter der Steuerung sind nicht bekannt

Wenn Sie die aktuellen Ethernet-Parameter der Steuerung nicht kennen, gehen Sie so vor:

1. Klicken Sie im Fenster „System-Login“ auf die Schaltfläche [Suchen] [1]. Das Fenster "Suchen per MAC" öffnet sich.



12086505227

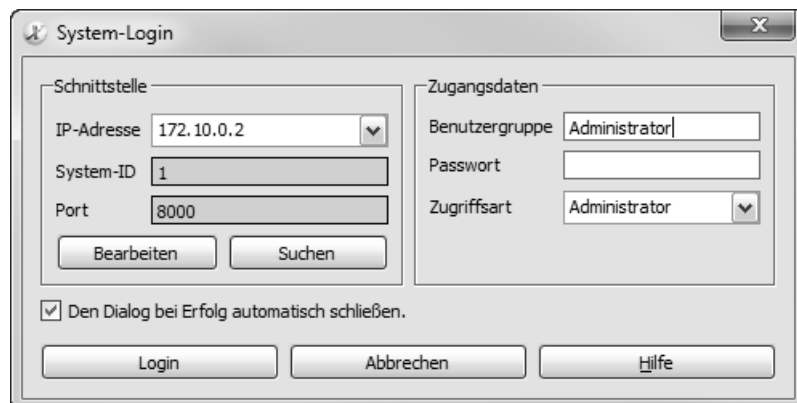
2. Geben Sie im Feld "MAC-Adresse" [2] die MAC-Adresse der CPU ein. Sie finden die MAC-Adresse auf einem Aufkleber auf der Sicherheitssteuerung.
3. Klicken Sie auf die Schaltfläche [Suchen] [3]. Die Daten für „IP-Adresse“, „Subnet Mask“ und „SRS“ werden ausgelesen und in der Gruppe „Einstellungen“ angezeigt.
4. Klicken Sie auf die Schaltfläche [Übernehmen] [4]. Die ausgelesenen Daten werden in das Dialogfenster System-Login übernommen.

System-Login durchführen

Gehen Sie so vor:

Geben Sie in der Gruppe „Zugangsdaten“ zur Autorisierung die Daten der Standardbenutzergruppe folgendermaßen ein:

- Klicken Sie in das Feld „Benutzergruppe“ und drücken Sie die Tastenkombination <Strg + A>.
- Die Einträge in den Feldern „Benutzergruppe“ und „Zugriffsart“ werden automatisch ausgefüllt.



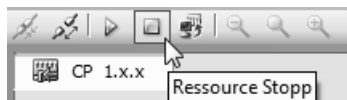
12085484171

2. Klicken Sie auf die Schaltfläche „Login“. Das Control Panel der Ressource wird aufgerufen. Wenn die Daten der Standardbenutzergruppe nicht akzeptiert werden, wurde auf der Sicherheitssteuerung eine Benutzerverwaltung konfiguriert. Sie müssen zum Login die Daten eines Administrators aus dieser Benutzerverwaltung verwenden. Falls diese Daten nicht bekannt sind, müssen Sie die Sicherheitssteuerung auf die Werkseinstellungen zurücksetzen.

System-ID einstellen

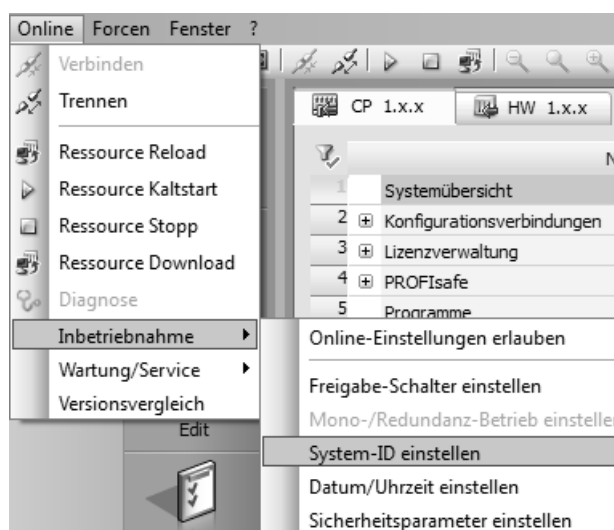
Gehen Sie so vor:

1. Stellen Sie sicher, dass sich das System im Zustand „STOPP“ befindet. Andernfalls kann die System-ID nicht geändert werden.
2. Klicken Sie in der Symbolleiste auf das Symbol „Ressource Stopp“.



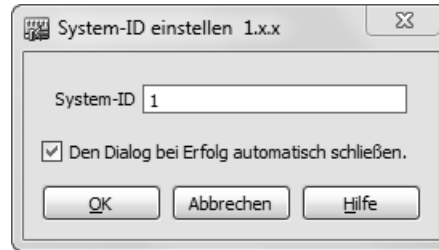
12085635339

3. Wählen Sie im Menü [Online] den Menüpunkt [Inbetriebnahme] / [System-ID einstellen].



12086834187

4. Das Fenster „System-ID einstellen“ wird aufgerufen. In der Kopfzeile wird die aktuelle System-ID angezeigt.



12088041483

5. Geben Sie die gewünschte System-ID ein und bestätigen Sie mit [OK].
6. Schließen Sie das Control Panel und fahren Sie fort mit Kapitel „Ressource (MOVISAFE® HM31) laden und starten“.

HINWEIS



Durch das Ändern der System-ID wird die Kommunikation zwischen PADT und Sicherheitssteuerung unterbrochen, da der System-Login mit einer anderen System-ID erfolgte.

14.2.7 Ressource (MOVISAFE® HM31) laden und starten

Voraussetzungen

Um eine Ressource laden und starten zu können, muss die Steuerung wie in den Kapiteln „Inbetriebnahme MOVISAFE® HM31 mit/ohne Werkseinstellungen“ beschrieben in Betrieb genommen werden. Folgende Voraussetzungen müssen erfüllt sein:

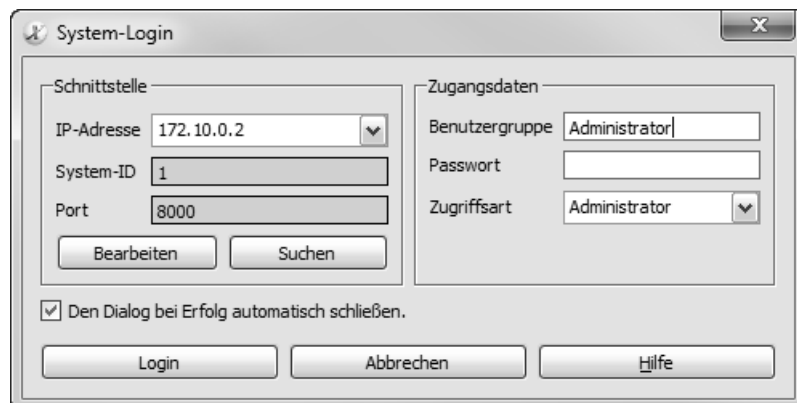
1. Die im Projekt verwendete System-ID muss in der Sicherheitssteuerung MOVISAFE® HM31 eingestellt sein.
2. In SILworX® muss ein fehlerfrei kompiliertes Projekt geöffnet sein.
3. Der Benutzer hat die Berechtigung, einen System-Login mit Schreibrechten durchzuführen.

System-Login durchführen

Gehen Sie so vor:

Geben Sie in der Gruppe „Zugangsdaten“ zur Autorisierung die Daten der Standardbenutzergruppe folgendermaßen ein:

- Klicken Sie in das Feld „Benutzergruppe“ und drücken Sie die Tastenkombination <Strg + A>.
- Die Einträge in den Feldern „Benutzergruppe“ und „Zugriffsart“ werden automatisch ausgefüllt.



12085484171

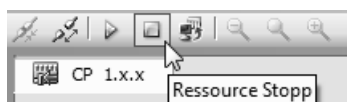
2. Klicken Sie auf die Schaltfläche [Login]. Das Control Panel der Ressource wird aufgerufen.

Download durchführen

Für einen Download muss sich das System im Zustand „STOPP“ befinden. Der Systemzustand wird im Control Panel in der Gruppe „Systeminformation“ angezeigt.

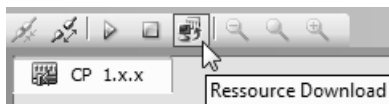
Gehen Sie folgendermaßen vor:

1. Klicken Sie in der Symbolleiste auf das Symbol „Ressource Stopp“



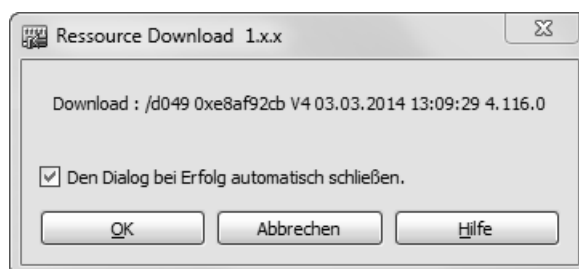
12085635339

2. Klicken Sie in der Symbolleiste auf das Symbol „Ressource Download“.



12088151691

3. Das Fenster „Ressource Download“ wird aufgerufen. Starten Sie den Download mit [OK].



12088159883



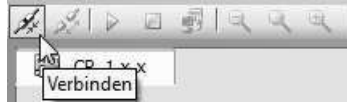
HINWEIS

Nach dem erfolgreichen Download werden die im Projekt konfigurierten IP-Adressen aktiv. Wenn sich die neue IP-Adresse der Ressource von der beim Login verwendeten IP-Adresse unterscheidet, wird die Verbindung zwischen dem Programmiergerät und der Ressource unterbrochen.

Ressource Kaltstart

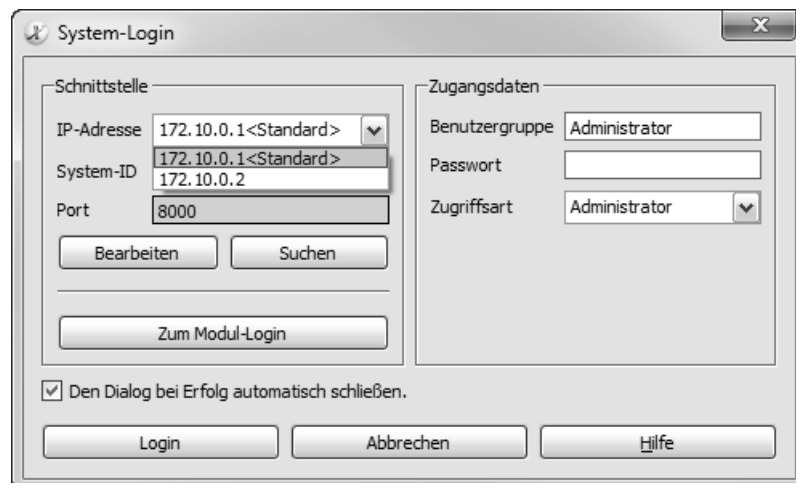
Gehen Sie so vor:

1. Loggen Sie sich nach dem Verbindungsverlust nach Download erneut ein. Klicken Sie dazu in der Symbolleiste auf das Symbol „Verbinden“.



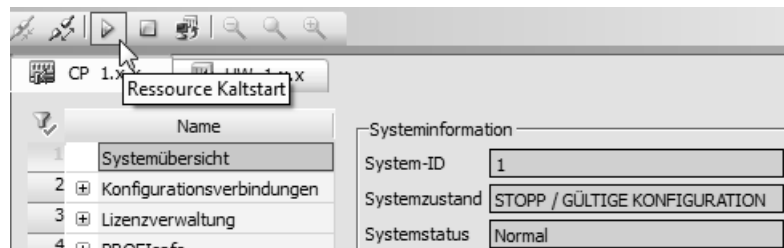
12088190475

2. Das Fenster „System-Login“ wird aufgerufen. Wählen Sie in der Gruppe „Schnittstelle“ in der Auswahlliste „IP-Adresse“ die passende IP-Adresse.



12088307083

3. Klicken Sie auf die Schaltfläche [Login]. Klicken Sie im Control Panel in der Symbolleiste auf das Symbol „Ressource Kaltstart“.



12088313355

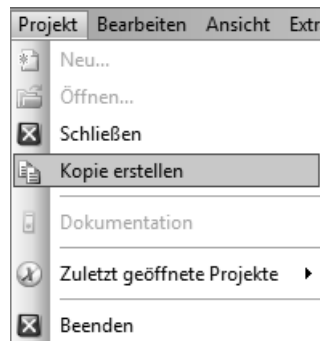
4. Die CPU geht in den Zustand „RUN“. Beachten Sie auch die Systeminformation (System-ID, Systemzustand, Systemstatus) im Control Panel.

Sicherungskopie erstellen

Erstellen Sie grundsätzlich nach jedem Laden eine Sicherungskopie Ihres Projekts in einem gesonderten Verzeichnis. Versehen Sie die Sicherungskopie mit einem Schreibschutz. Dies stellt sicher, dass eine Kopie (ein Projekt) nicht unbeabsichtigt verändert wird.

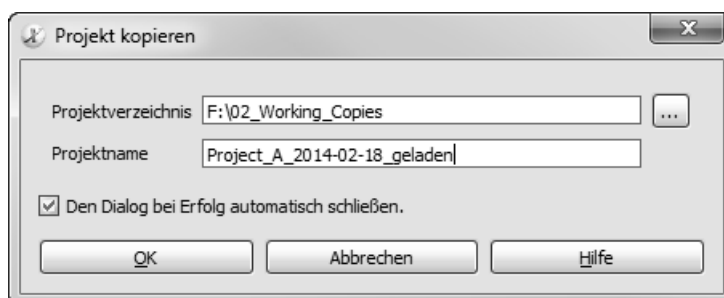
So erstellen Sie eine Sicherungskopie:

1. Wählen Sie in SILworX® im Menü [Projekt] den Menüpunkt [Kopie erstellen].



12088433547

2. Das Fenster "Projekt kopieren" öffnet sich. Wählen Sie ein Projektverzeichnis, in dem die Kopie des Projekts erstellt werden soll. Geben Sie außerdem einen Projektname mit entsprechenden Vermerken an.



12088438155

3. Klicken Sie auf [OK]. Eine Sicherungskopie Ihres Projekts wird erstellt.

14.2.8 Elektrische Installation

Nachdem die Sicherheitssteuerung ohne Störungen in den Systemzustand „RUN“ gegangen ist, können Sie die Spannungsversorgung der Steuerung ausschalten und alle Kommunikationsverbindungen und alle Verbindungen zu Ein- und Ausgängen wiederherstellen. Beachten Sie dabei die vorhandene Dokumentation / Beschreibung der elektrischen Anschlüsse.

14.2.9 Verifikation

Nach dem Gerätetausch und dem Anschluss aller Verbindungen zu Ein- und Ausgängen und zu Kommunikationsschnittstellen müssen alle relevanten Parameter der Sicherheitssteuerung verifiziert werden. Dazu gehören z. B. die Systemdaten (Konfigurations-CRC, Konfigurationspfad im FS) sowie Gebersignale und Kommunikationsschnittstelle. SEW-EURODRIVE verwendet dazu das entsprechende Formular.

14.3 Störungsinformation

Störungen im Prozessorsystem (CPU) haben meist das Abschalten der gesamten Steuerung zur Folge und werden durch die Status-LED "ERROR" angezeigt.

Die Anzeige kann durch Ausführen des Befehls "Ressource Rebooten" im Menü [Extra] des Control Panels von SiLworX® gelöscht werden. Die Steuerung wird gebootet und erneut gestartet. Störungen in Eingangs- und Ausgangskanälen erkennt das System während des Betriebs automatisch und zeigt sie auf der Oberseite des Geräts durch die Status-LED "FAULT" an.

Das PADT (SILworX®) bietet auch bei einem Stopp der Steuerung die Möglichkeit, festgestellte Fehler über die Diagnose auszulesen, so weit die Kommunikation nicht ebenfalls gestört ist.

- Prüfen Sie vor dem Wechsel einer Steuerung, ob eine externe Leitungsstörung vorliegt und der entsprechende Sensor/Aktor in Ordnung ist.

14.4 Laden von Betriebssystemen

Prozessorsystem und Kommunikationssystem haben unterschiedliche Betriebssysteme, die in wieder beschreibbaren Flash-Speichern gespeichert sind und bei Bedarf ersetzt werden können.



⚠ WARNUNG

Unterbrechung des sicherheitsgerichteten Betriebs durch Laden neuer Betriebssysteme vom Programmierwerkzeug.

Tod oder schwere Körperverletzungen.

- Zum Laden neuer Betriebssysteme vom Programmierwerkzeug muss die Steuerung im Zustand „STOPP“ sein.
- Der Betreiber muss sicherstellen, dass während dieser Zeit die Sicherheit der Anlage gewährleistet bleibt, z. B. durch organisatorische Maßnahmen.



HINWEIS

- Das Programmierwerkzeug verhindert das Laden von Betriebssystemen im Zustand RUN und meldet dies.
- Eine Unterbrechung oder inkorrekte Beendigung des Ladens führt dazu, dass die Steuerung nicht mehr funktionsfähig ist. Es ist jedoch möglich, erneut ein Betriebssystem zu laden.

Das Betriebssystem für das Prozessorsystem (CPU-Betriebssystem) ist vor dem für das Kommunikationssystem (COM-Betriebssystem) zu laden. Voraussetzung zum Laden von Betriebssystemen ist, dass das neue Betriebssystem in einem Verzeichnis abgelegt ist, das mit dem Programmierwerkzeug zu erreichen ist.

14.4.1 Laden von Betriebssystemen mit SILworX®

Gehen Sie so vor, um ein neues Betriebssystem zu laden:

1. Steuerung in den Zustand „STOPP“ bringen, falls nicht bereits geschehen.
2. Online-Ansicht der Hardware öffnen, dabei auf der Steuerung mit Administratorrechten anmelden.
3. Zu ladendes Modul (Prozessormodul oder Kommunikationsmodul) mit rechter Maustaste klicken.
4. Im geöffneten Kontextmenü [Wartung/Service] / [Modul Betriebssystem laden] klicken.
5. Im Dialogfenster "Modul Betriebssystem laden" die Art des zu ladenden Betriebssystems auswählen.
6. Im geöffneten Dateiauswahlfenster die Datei mit dem zu ladenden Betriebssystem auswählen und [Öffnen] klicken.

SILworX® lädt das neue Betriebssystem in die Steuerung.

14.5 Außerbetriebnahme

Um die Sicherheitssteuerung außer Betrieb zu nehmen, schalten Sie das Gerät mit geeigneten Maßnahmen spannungsfrei.

15 Anhang

15.1 Glossar

Begriff	Beschreibung
DC-24V	Die Sicherheitssteuerung verfügt über folgende DC-24-V-Eingangsspannungspotenziale: 24V_CU: DC-24V-Eingang – Steuerung 24V_L: DC-24V-Eingang – Last 24V_S: DC-24V-Eingang – Sensorversorgung Bezugspotenzial: 0V24
A3	SILworX® Archive File
ARP	Address Resolution Protocol (Netzwerkprotokoll zur Zuordnung von Netzwerkadressen zu Hardware-Adressen)
BS	Betriebssystem
BL	Boot-Loader
BWS	Berührungslos Wirkende Schutzeinrichtung
COM	Kommunikationsmodul
COE	CANopen-Softwaremodul
CRC	Cyclic Redundancy Check (Prüfsumme)
CUT	Com-User Task
DCS	Distributed Control System (Prozessleitsystem)
DI	Digital Input (Binäreingang)
DO	Digital Output (Binärausgang)
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm
ESD	Electrostatic Discharge (elektrostatische Entladung)
FB	Feldbus-Schnittstelle der Steuerung
FBS	Funktionsbausteinsprache
FIFO	First In First Out (Datenspeicher)
FTA	Field Termination Assembly
FTZ	Fehlertoleranzzeit
ICMP	Internet Control Message Protocol (Netzwerkprotokoll für Status- und Fehlermeldungen)
IEC	Internationale Normen für die Elektrotechnik
IF	InterFace
MAC-Adresse	Hardware-Adresse eines Netzwerkanschlusses (Media Access Control)
PADT	Programming and Debugging Tool (gemäß IEC 61131-3), PC mit SILworX®
NVRAM	Non Volatile Random Access Memory, nicht-flüchtiger Speicher

Begriff	Beschreibung
PE	Protective Earth (Schutzerde)
PELV	Protective Extra Low Voltage (Funktionskleinspannung mit sicherer Trennung)
PES	Programmierbares elektronisches System
POE	Programm-Organisationseinheiten (gemäß IEC 61131-1)
PFD	Probability of Failure on Demand (Wahrscheinlichkeit eines Fehlers bei Anforderung einer Sicherheitsfunktion)
PFF-HM31A	Sicherheitssteuerung
PFH	Probability of Failure per Hour (Wahrscheinlichkeit eines gefährbringenden Ausfalls pro Stunde)
R	Read (Systemvariable liefert Wert, z. B. an Anwenderprogramm)
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 (Spaltenüberschrift für Art von Systemvariable)
SB	Systembus (-modul)
SELV	Safety Extra Low Voltage (Schutzkleinspannung)
SFF	Safe Failure Fraction (Anteil der sicher beherrschbaren Fehler)
SIL	Safety Integrity Level (gemäß IEC 61508)
SILworX®	Programmierungswerkzeug für Sicherheitssteuerung PFF-HM31A
SNTP	Simple Network Time Protocol (RFC 1769)
S.R.S	System.Rack.Slot (Adressierung eines Moduls)
SW	Software
S&R	Send und Receive; im Zusammenhang mit TCP-Protokoll
TMO	Timeout
W	Write (Systemvariable wird mit Wert versorgt, z. B. vom Anwenderprogramm)
Watchdog (WD)	Zeitüberwachung für Module oder Programme. Bei Überschreiten der Watchdog-Zeit geht das Modul oder Programm in den Fehlerstopp.
WDZ	Watchdog-Zeit

Stichwortverzeichnis

A

Abschnittsbezogene Sicherheitshinweise	10
ACTIVE, Hilfsfunktionsbaustein	104
ALARM, Hilfsfunktionsbaustein	105
Alarmer und Ereignisse, Aufzeichnung	14
Allgemeine Hinweise	9
Anhang	273
Anwenderprogramm	
Betriebsarten	198
Erstellen und laden	197
Forcen	208
Multitasking	198
Reload	206
Aufbau und Gebrauch der Dokumentation	9
Aufzeichnung von Alarmen und Ereignissen	14
Alarmer und Ereignisse, Definition	14
Aufzeichnung von Ereignissen	15
Bildung von Ereignissen	15
Weitegrabe von Ereignissen	15

B

Bedienelemente	242
Behebung von Störungen	270
Benutzerverwaltung mit SILworX®	231
Für die Steuerung	232
Für ein SILworX®-Projekt	231
Betrieb	242
Bedienung	242
Diagnose	242
Betriebsspannung, Überwachung	13
Betriebssystem	194
Fehlerarten	194
Funktionen	194
Prozessorsystem	195

C

Checkliste zur Projektierung, Programmierung und Inbetriebnahme	211
Com-User Task	
Eigenschaften	193
Einführung	193
Voraussetzungen	193
Control-Panel (safeethernet)	
Anzeigefeld	39

Control-Panel, Send & Receive TCP	185
---	-----

D

Darstellungskonventionen	10
DEID, Hilfsfunktionsbaustein	107
Diagnose	242, 270
Diagnosehistorie	244
In SILworX®	245
Dokumentation	
Weiterführende (mitgeltende) Unterlagen	11

E

Eingebettete Sicherheitshinweise	11
Ethernet-Schnittstellen	
Ethernet-Switch	17
Service-Schnittstelle X4223	17
X4233_1 und X4233_2	17

F

Fehlerarten und -behandlung	
Interne Fehler	195
Permanente Fehler bei Ein- und Ausgängen	195
Vorübergehende Fehler bei Ein- und Ausgängen	195
Fehlercodes	
Modbus TCP/IP-Verbindung	149
PROFINET IO Funktionsbausteine	110
Forcen	208
Einschränkungen	210
Force-Editor	210
Zeitbegrenzung	210
Fremdsysteme mit Pad-Bytes	167
Funktionen des Prozessor-Betriebssystems	194
Funktionsbausteine Send & Receive TCP	
TCP_Receive	174
TCP_ReceiveLine	177
TCP_ReceiveVar	181
TCP_Reset	168
TCP_Send	171
Funktionscodes	
Modbus, Modbus-Slave	138
Modbus-Master	121
Modbus-Master, Standard	121
SEW-spezifische, Modbus-Master	122

SEW-spezifische, Modbus-Slave	140	Ressourcenkonfiguration vom Programmierge- rät laden	229
G		System-ID und Verbindungsparameter konfigu- rieren	229
Gerät außer Betrieb nehmen	272	Konfiguration von Alarmen und Ereignissen	236
Glossar	273	Kurzschlussverhalten der Ausgangskanäle	14
H		L	
Haftungsausschluss	11	Laden von Betriebssystemen	271
Handbuch		Mit SILworX®	271
Weiterführende (mitgeltende) Unterlagen	11	LED-Anzeige	242
Hinweise		M	
Kennzeichnung in der Dokumentation	10	Mängelhaftungsansprüche	11
I		Marken	12
ID, Hilfsfunktionsbaustein	107	Menüfunktionen des Modbus-Master	
Inbetriebnahme	211	Edit	118
Konfiguration mit SILworX®	211	Eigenschaften	119
Instandhaltung		Menüfunktionen des PROFINET-IO-Controllers	
Laden von Betriebssystemen	271	Menü PROFINET-IO Controller	54
K		PROFINET-IO Device (im Controller)	56
Kommunikation		Menüfunktionen im Send & Receive TCP-Protokoll	
Lastbegrenzung	19	Edit	159
Mit dem Programmierwerkzeug	17	Eigenschaften	159
safeethernet	20	Menüfunktionen TCP-Verbindung	
Verfügbare Protokolle	16	Edit	161
Konfiguration der Kommunikation mit SILworX®	235	Modbus TCP/UDP	112
Ethernet-Schnittstellen konfigurieren	235	Beispiel	113
Konfiguration der PROFINET-IO-Funktionsbausteine		Beispiel zur alternativen Register/Bit-Adressie- rung	117
Funktionsbausteine im Anwenderprogramm konfigurieren	102	Modbus-Master	112
Funktionsbausteine im Strukturbaum von SIL- worX® konfigurieren	103	Modbus-Slave	132
Hinzufügen der Funktionsbausteinbibliotheken	102	Modbus-Beispiel	113
Konfiguration mit SILworX®	211	Konfiguration des Modbus-TCP-Master	115
Datum und Uhrzeit setzen	231	Konfiguration des Modbus-TCP-Slave	113
Ein- und Ausgänge	227	Modbus-Master	
Generierung der Ressourcenkonfiguration ...	228	Anforderungstelegramm zum Lesen und Schrei- ben	125
Kommunikationsmodul	217	Anforderungstelegramm zum Schreiben	127
Prozessormodul	211	Anforderungstelegramme zum Lesen	124
Ressourcenkonfiguration aus dem Flash-Spei- cher laden	230	Control-Panel	131
Ressourcenkonfiguration im Flash-Speicher be- reinigen	230	Eigenschaften	112
		Ethernet-Slaves (TCP/UDP-Slaves)	128
		Format der Request und Response Header ..	123
		Funktioncodes	121
		Menüfunktionen	118
		SEW-spezifische Funktioncodes	122

Standard Funktioncodes	121	PROFINET IO / PROFIsafe-Beispiel (Device)	
Modbus-Slave		Konfiguration des PROFINET-IO-Device in SIL-	
Control-Panel	148	worX®	76
Eigenschaften	132	PROFINET-IO	
Fehlercodes der Modbus TCP-IP-Verbindung	149	Funktionsbausteine	85
Konfiguration	133	PROFINET-IO / PROFIsafe-Beispiel (Device)	
Menü Eigenschaften Modbus-Slave-Set	133	Menüfunktionen des PROFINET-IO-Device ...	78
Menüfunktionen	133	PROFINET-IO Device	75
Modbus-Adressierung durch Bit und Register	142	Systemanforderung	75
Modbus-Funktioncodes	138	PROFINET-IO Funktionsbausteine	85
Offsets für alternative Modbus-Adressierung	145	Funktionsbaustein MSTAT	86
Sende-/Empfangsvariablen zuweisen	136	Funktionsbaustein RALRM	89
SEW-spezifische Functioncodes	140	Funktionsbaustein RDREC	93
MSTAT, Funktionsbaustein	86	Funktionsbaustein SLACT	96
Multitasking	198	Funktionsbaustein WRREC	99
Mode 1, 2, 3	203	PROFINET-IO Hilfsfunktionsbausteine	104
N		Hilfsfunktionsbaustein ACTIVE	104
NSLOT, Hilfsfunktionsbaustein	108	Hilfsfunktionsbaustein ALARM	105
P		Hilfsfunktionsbaustein DEID	107
PADT (Programmierwerkzeug)	17	Hilfsfunktionsbaustein ID	107
Parameter und Fehlercodes der Ein- und Ausgänge	246	Hilfsfunktionsbaustein NSLOT	108
Digitale Ausgänge MOVISAFE® HM31	249	Hilfsfunktionsbaustein SLOT	108
Digitale Eingänge MOVISAFE® HM31	246	Hilfsfunktionsbaustein STDDIAG	109
Zähler MOVISAFE® HM31	251	PROFIsafe	42
Produktnamen	12	Auflagen für den sicheren Betrieb	46
PROFINET IO		Control-Byte und Status-Byte	43
Beispiel für einen Strukturbaum des Controllers	54	PROFINET-IO / PROFIsafe-Beispiel (Device)	75
Controller und PROFIsafe-Host	48	PROFINET-IO-/PROFIsafe Beispiel (Controller)	
Einführung	41	49
Fehlercodes der Funktionsbausteine	110	PROFINET-IO-Controller und PROFIsafe-Host	
Hilfsfunktionsbausteine	104	48
Konfiguration der Funktionsbausteine	102	Safety Function Response Time (SFRT)	45
Menüfunktionen des Controllers	54	Watchdog-Zeit	43
PROFINET-IO / PROFIsafe-Beispiel (Device)	75	Programmierwerkzeug (PADT)	17
PROFINET-IO-/PROFIsafe Beispiel (Controller)		Projektübergreifende Kommunikation	
.....	49	Varianten	37
PROFIsafe	42	Protokolle	
Steuerung des Consumer/Provider Status (IOxS)	41	Registrierung und Aktivierung	18
		Simple Network Time Protokoll (SNTP)	189
		Verfügbare	16
		Prozessorssystem	
		Programmierung	196
		Prozessorssystem	195
		Betriebszustände	196

R

RALRM, Funktionsbaustein	89
RDREC, Funktionsbaustein	93
Reload	206
Einsatzbedingungen	207
Reparaturen	255

S

safeethernet	
Anschlüsse	40
Control Panel	38
Editor	23
Eigenschaften	20
Grundlagen	21
Maximale Kommunikationszeitscheibe	40
Maximale Reaktionszeit	31
Parameter	28
Profile	32
Projektübergreifende Kommunikation	36
Schnittstellen	40
Systemstruktur	21
safeethernet-Editor	23
Detailansicht	25
Register Sysemvariablen	25
safetethernet-Profil	
Profil I (Fast & Cleanroom)	33
Profil II (Fast & Noisy)	34
Profil III (Medium & Cleanroom)	34
Profil IV (Medium & Noisy)	35
Profil V (Slow & Cleanroom)	35
Profil VI (Slow & Noisy)	36
Send & Receive TCP	151
Anlegen eines Send & Receive TCP-Protokolls	151
Beispiel Konfiguration	152
Beispiel Konfiguration der Steuerung SIMATIC 300	154
Beispiel Konfiguration von MOVISAFE® HM31	157
Control-Panel	185
Datenaustausch	164
Fremdsysteme mit Pad-Bytes	167
Funktionsbausteine, Einführung und Übersicht	167
Menüfunktionen	159
Menüfunktionen TCP-Verbindung	161

Systemanforderungen	151
TCP-Verbindungen	165
Service	242
Sicherheitsgerichtetes Protokoll (safeethernet)	
Berechnung der maximalen Reaktionszeit	32
Maximale Zykluszeit der Sicherheitssteuerung	28
Receive Timeout	28
Response Time	29
Sicherheitshinweise	
Aufbau der abschnittsbezogenen	10
Aufbau der eingebetteten	11
Kennzeichnung in der Dokumentation	10
Signalworte in Sicherheitshinweisen	10
Simple Network Time Protokoll (SNTP)	
SNTP-Client	189
SNTP-Client (Server Info)	190
SNTP-Server	191
SLACT, Funktionsbaustein	96
SLOT, Hilfsfunktionsbaustein	108
SNTP	189
Client anlegen	189
Server anlegen	191
Server Info anlegen	190
Status-LED	242
STDDIAG, Hilfsfunktionsbaustein	109
Steuerung des Consumer/Provider Status (IOxS)	41
Steuervariablen im Controller	41
Steuervariablen im DI-Device	42
Steuervariablen im DO-Device	42
Systemaufbau	13

T

TCP-Verbindung	
Azyklischer Datenaustausch	166
Eigenschaften	162
Flusskontrolle	166
Gleichzeitiger zyklischer und azyklischer Datenaustausch	166
Systemvariablen	161
Zyklischer Datenaustausch	165
Temperaturzustand, Überwachung	13
Temperaturschwelle für Meldungen einstellen	14

U

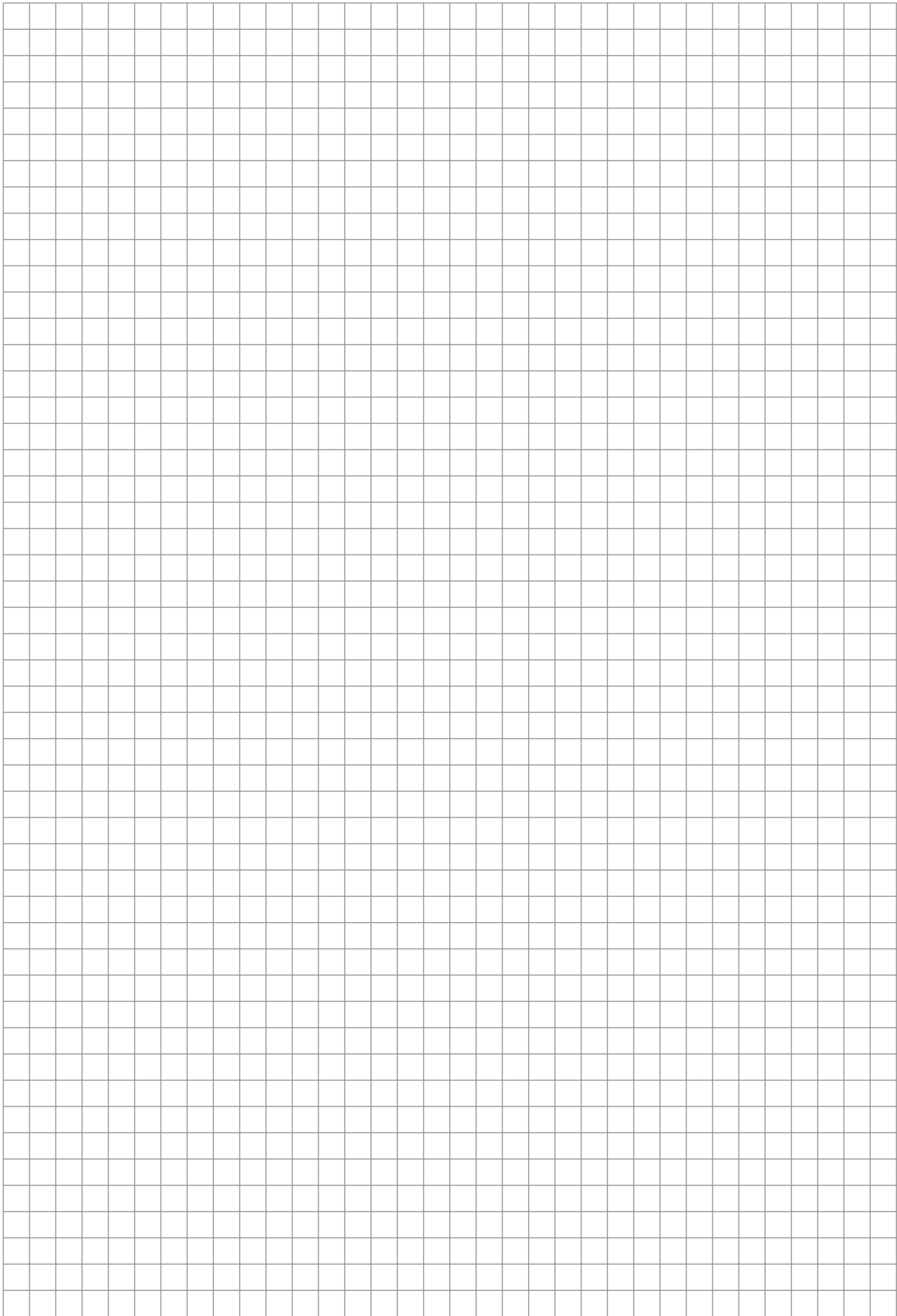
Umgang mit dem Anwenderprogramm	240
Neustart nach Fehler	240
Programmstart nach STOPP/GÜLTIGE KONFIGURATION	240
Setzen der Parameter und Schalter	240
Stoppen	240
Testmodus	240
Urheberrechtsvermerk	12

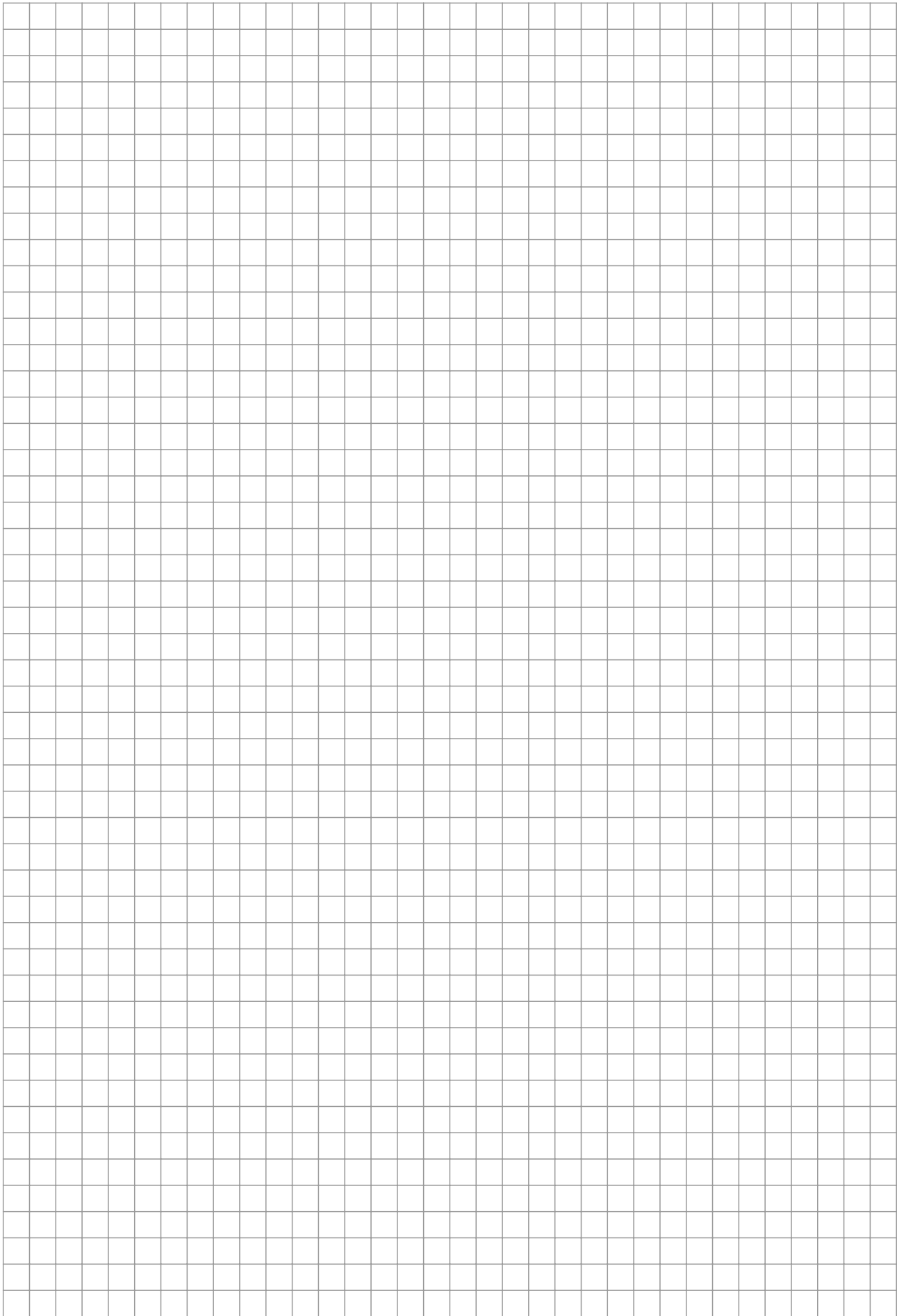
W

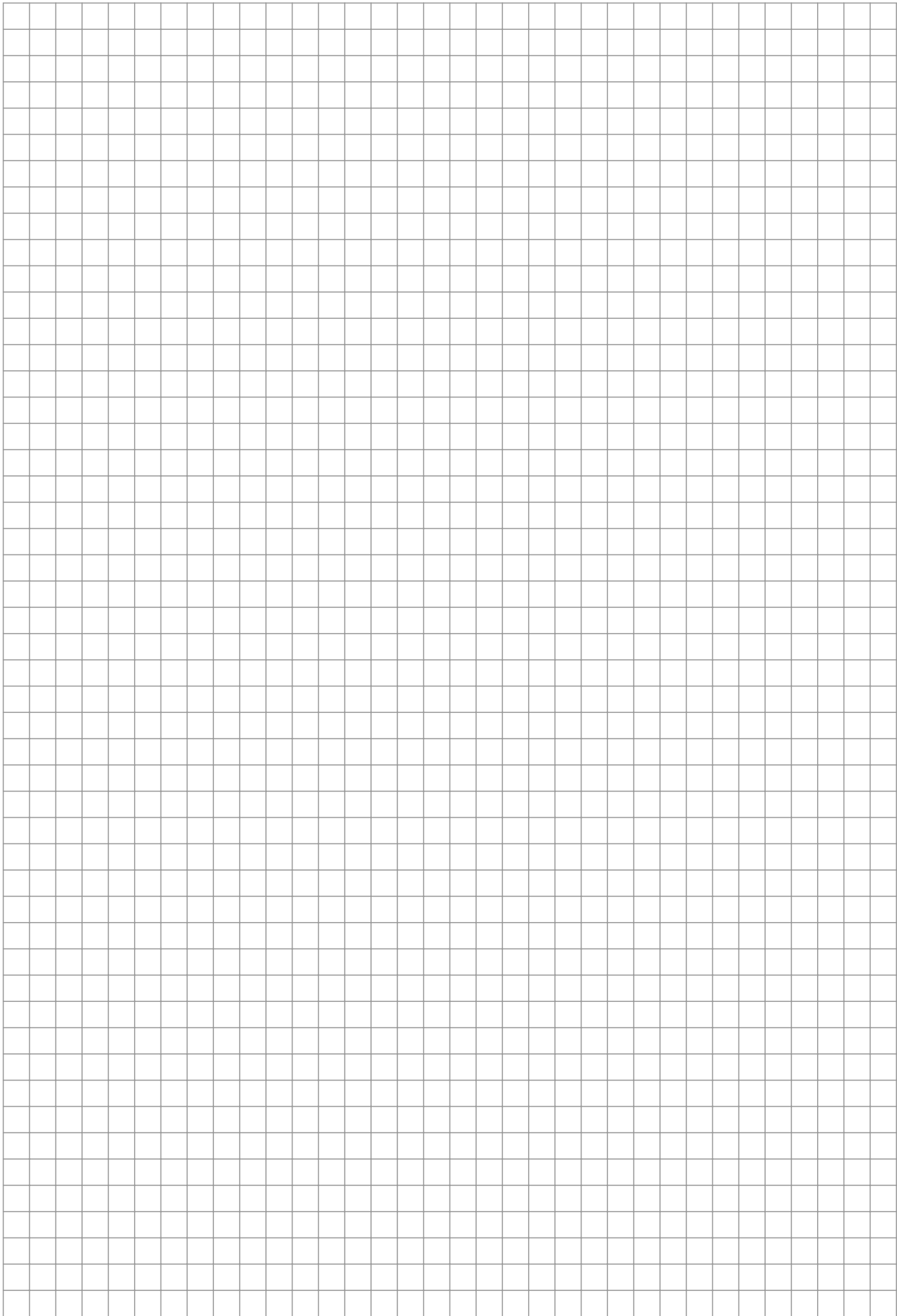
WRREC, Funktionsbaustein	99
--------------------------------	----

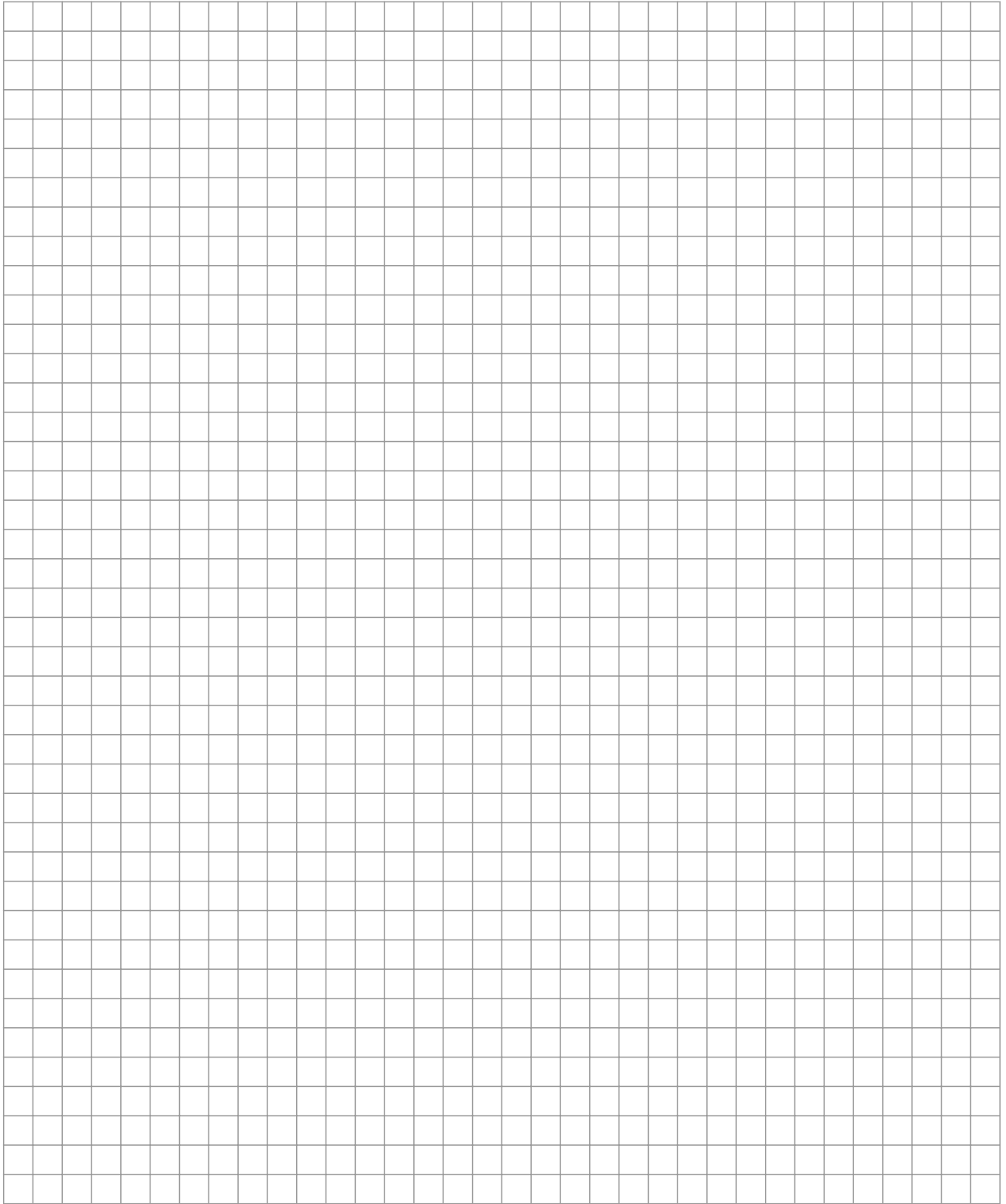
Z

Zielgruppe der Dokumentation	10
------------------------------------	----











SEW-EURODRIVE
Driving the world

SEW
EURODRIVE

SEW-EURODRIVE GmbH & Co KG
P.O. Box 3023
76642 BRUCHSAL
GERMANY
Phone +49 7251 75-0
Fax +49 7251-1970
sew@sew-eurodrive.com
→ www.sew-eurodrive.com