

XN Safety Designer



Impressum

Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhälter.

Service

Für Service und Support kontaktieren Sie bitte Ihre lokale Vertriebsorganisation.

Kontaktdaten: Eaton.com/contact

Service-Seite: Eaton.com/aftersales

Originaldokumentation

ist die deutsche Ausführung dieses Dokuments.

Redaktionsdatum

06/25 MN050028, 1. Auflage

Copyright

© 2025 Eaton Industries GmbH, 53105 Bonn

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Handbuches darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Zustimmung der Firma Eaton, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten.

Inhaltsverzeichnis

	XN Safety Designer	1
	Impressum	2
	Inhaltsverzeichnis	3
0.1	Zu dieser Dokumentation	17
0.1.1	Änderungsprotokoll	17
0.1.2	Zielgruppe	18
0.1.3	Haftungsausschluss	19
0.1.4	Kurzbezeichnungen	20
0.1.5	Lesekonventionen	20
0.1.5.1	Warnhinweise	21
0.1.5.2	Weitere Nutzungsinformationen	22
1.	Bedienung Programmiersoftware XN Safety Designer	23
1.1	Begriffsklärung: Projekt	23
1.2	Übersichtsbeispiel Projekt, Hardware-Topologie, Programmnetzwerke	24
1.3	Schematische Darstellung einer Projekterstellung	24
1.4	Laden von Projekten mit einer neueren Version von XN Safety Designer	26
2.	Bedingungen und Anforderungen für den Anwender	27
3.	Erstellen der Hardware-Konfiguration	28
3.1	Ermittlung der Hardware-Konfiguration durch Online-Verbindung	28
3.2	Manuelles Editieren einer XS-102-C00-000 im Stand-Alone-Betrieb	29
3.3	Einfaches Ermitteln der Daten auf der Safety-CPU	31
4.	Grafik Editor	32
4.1	Oberfläche	32
4.2	Menüleiste	32
4.2.1	File	33
4.2.2	Edit	34

4.2.3	View	35
4.2.4	Build	36
4.2.5	Debug	37
4.2.6	Help	37
4.3	Werkzeug-Symbolleiste	38
4.4	Tree-Ansichten	40
4.4.1	Project Tree	41
4.4.2	Hardware Tree	43
4.4.3	Function Block Tree	46
4.4.4	Variable Tree	48
4.4.4.1	Bit-Variablen	49
4.4.5	Library Tree	51
4.4.6	Function Block Macro Tree	52
4.4.7	Unterordner in der Tree-Ansicht	55
4.5	Property Browser	57
4.6	Editor	58
4.7	Spalte für Eingänge	59
4.8	Spalte für Ausgänge	62
4.9	Network Editor	63
4.9.1	Netzwerkelemente	64
4.9.1.1	Bit-Variablen	65
4.10	Allgemeine Arbeitsweise des Editors	67
4.10.1	Füllen der Eingangsspalte	67
4.10.2	Füllen der Ausgangsspalte	67
4.11	Arbeiten im Netzwerkeeditor	68
4.11.1	Einfügen von Netzwerkelementen	68
4.11.2	Selektion von Netzwerkelementen	68
4.11.3	Verschieben von Netzwerkelementen	69
4.11.4	Löschen von Netzwerkelementen	69
4.11.5	Parametrierung von Netzwerkelementen	69
4.11.6	Ziehen von Verbindungen	70
4.11.7	Selektion von Verbindungen	71
4.11.8	Löschen von Verbindungen	71
4.11.9	Globale Verbindungen	72

4.11.10	Einfügen von Kommentaren	73
4.11.11	Selektion von Kommentaren	73
4.11.12	Verschieben von Kommentaren	73
4.11.13	Löschen von Kommentaren	73
4.11.14	Ändern der Kommentargröße	74
4.11.15	Einfügen von Kommentartexten	74
4.11.16	Kopieren, Ausschneiden und Einfügen von Elementen	74
4.11.17	Ersetzen von Platzhaltern	75
4.11.18	Ersetzen von bestehenden Elementen	75
4.12	Aufruf der Definition eines Elements	75
4.13	Aufruf der Startpunkte/Endpunkte einer Verbindung	76
4.14	Aufruf der temporären Variablen	77
4.15	Zoom - Pan	78
4.16	Ändern der Spaltenbreite	78
4.17	Fehlermeldungen beim Einfügen eines Elements ins Netzwerk ..	79
4.18	Einstellparameter im Projekt	81
4.18.1	Temporary Operation Time	81
4.18.2	Max. Cycle Time	81
4.18.3	Maximum Transmission Time	81
4.18.4	Maximum Transmission Time (Output as Input)	81
4.18.5	Filter Time	81
4.18.6	Cross Circuit Detection	81
4.18.7	Hardware Revision	82
4.18.8	Parameter Revision	82
4.18.9	Vereinfachte Darstellung ausgewählter Funktionsblöcke	83
4.18.10	Aktivieren des Schreibens der CRC	84
4.18.11	Aktivieren der Easy Instance Einstelllung für Instanzen	84
4.18.12	Aktivieren der Einstellung für einen Download mit Zusatz- informationen	84
4.18.13	Deaktivieren des Blinkens einer LED (lokale Module)	84
4.19	Enable Signal von sicheren Ausgängen	85
4.20	Optionale Module	87
4.21	Validierung und Generierung der Downloaddaten	89
4.22	Berechnung der Monitoring-Settings	97

4.23	Ressourcenverbrauch und Reaktionszeit	98
4.23.1	Ressourcenverbrauch	99
4.23.2	Abschätzung der Zykluszeit des Anwenderprogramms	100
4.23.3	Gesamtreaktionszeit	100
4.23.4	Länge der PDO-Frames	100
4.24	Debuggen	101
4.25	Forcen	102
4.26	Watch-Fenster	104
4.27	Anzeige der Online-Werte von Modulen im Hardware Tree	106
4.28	Log-Dateien von Safety-Modulen auslesen und anzeigen	109
4.29	Drucken	114
4.29.1	Einstellungen beim Netzwerk drucken	116
4.29.2	Einstellungen beim Project Tree drucken	117
4.30	CRC-Vergleichsdialog	118
4.31	Globale Suche	120
5.	Konstanten	121
6.	Versionskontrolle und Archivierung	122
6.1	Versionierung	122
6.2	Archivierung	122
6.3	Speichern als XML-Datei	122
7.	Protection Level Settings Passwort-Mechanismus	123
8.	Verwendung unsicherer Variablen über den CAN-Bus	125
8.1	Voraussetzung zur Verwendung	125
8.2	Aufbau der unsicheren Variablen	125
8.3	Erstellen eines Projekts mit der Verwendung der unsicheren Variablen über den CAN-Bus	126
8.4	Anzeigen der Variablen im Online-Betrieb	128
8.5	Austausch von Statusinformationen über CAN-Bus zur Safety- CPU	129
9.	Verwendung unsicherer Variablen im Stand-alone-Betrieb ..	130
9.1	SDO-Kommunikation	130
9.2	Aufbau SDO-Frame	131

9.3	Schreiben eines SDO-Frames	131
9.4	Lesen eines SDO-Frames	132
9.5	Übersicht über SDO-Kommandos	132
9.6	Detailbeschreibung SDO-Kommando READ	133
9.7	Detailbeschreibung SDO-Kommando WRITE	133
9.8	Detailbeschreibung SDO-Kommando GET_STATE	134
9.9	Detailbeschreibung SDO-Kommando GET_DIAG_INFO	135
9.10	Detailbeschreibung SDO-Kommando GET_SAFENBR	136
9.11	Detailbeschreibung SDO-Kommando GET_FIRMWARE_VER	136
9.12	Detailbeschreibung SDO-Kommando GET_NOP	137
9.13	Detailbeschreibung SDO-Kommando READ_VALUES	138
9.14	Detailbeschreibung SDO-Kommando WRITE_VALUES	138
9.15	Wichtige virtuelle Adressen der Safety-CPU	139
9.16	Blocktypen der Konfiguration	139
9.17	Auslesen der Konfiguration	140
9.18	Aufbau der einzelnen Blocktypen der Listeneinträge	141
9.18.1	CFG_MODULE_V3	141
9.18.2	CFG_REV	144
9.19	Status der I/Os der Erweiterungsmodule	144
9.20	Erkannte Erweiterungsmodule	145
9.21	Schnelle nicht sichere Ein- und Ausgangsvariablen	145
9.22	Unsichere Freigabesignale für lokale Ausgänge	145
10.	Ermittlung der Reihenfolge für das Kompilat einer Safety-CPU	146
10.1	Reihenfolge der Netzwerke	146
10.2	Reihenfolge der Instanzen innerhalb der Netzwerke	147
10.3	Anzeige der Abarbeitungsreihenfolge in Netzwerken	147
11.	Online-Aktionen und Download	149
11.1	Soll-/ Istvergleich der Hardware	151
11.2	Online Aktionen	152
11.3	Download Wizard	154
11.3.1	Schritt 1: Hardware Tree vergleichen und Kontrolle der Safety-Nummer des Moduls.	154

11.3.2	Schritt 2: Programm auf Safety-CPU löschen, neues Projekt downloaden, Code von der Safety-CPU hochladen und mit dem Projekt vergleichen.	155
11.4	Passwörter	157
11.4.1	Übersicht über Aktionen abhängig vom Password level	158
11.5	Upload Log-Data	158
11.6	Get Module State	159
12.	Export der Download-Daten für eine Safety-CPU	160
12.1	Erstellen der Binärdatei	160
12.1.1	Fehler bei der Prüfung im externen Tool:	161
12.1.2	Fehler beim Reload and Compare:	162
12.2	Binärdatei auf eine Safety-CPU spielen	163
13.	Funktionsblöcke	164
13.1	Datentypen	164
13.2	Besonderheiten bei der Implementierung	165
13.3	Zeitüberwachung bei Funktionsblöcken	166
13.4	Standard-Funktionsblöcke	167
13.4.1	Allgemeine Informationen	167
13.4.2	Funktionsblöcke zur Typkonvertierung	169
13.4.2.1	BOOL_TO_SAFEBOOL	169
13.4.2.2	DINT_TO_SAFEDINT	170
13.4.2.3	SAFEBOOL_TO_BOOL	170
13.4.2.4	SAFEDINT_TO_DINT	170
13.4.2.5	SAFEDINTERROR_TO_SAFEDINT	171
13.4.2.6	ERROR_TO_SFALSE	171
13.4.3	Numerische Standard-Funktionsblöcke	172
13.4.3.1	ABS (Absolutwert)	173
13.4.3.2	ABS_NS (Absolutwert, not safe)	173
13.4.3.3	ADD (Addition)	173
13.4.3.4	ADD_NS (Addition, not safe)	174
13.4.3.5	SUB (Subtraktion)	174
13.4.3.6	SUB_NS (Subtraktion, not safe)	175
13.4.3.7	MULDIV_NS (Multiplikation und Division, not safe)	175

13.4.3.8	MULDIV (Multiplikation und Division)	176
13.4.3.9	SF_SampleHold	176
13.4.3.10	SHL (Shift-Left)	177
13.4.3.11	SHL_NS (Shift-Left, not safe)	177
13.4.3.12	SHR (Shift-Right)	178
13.4.3.13	SHR_NS (Shift-Right, not safe)	178
13.4.3.14	AND_BW (UND, bitweise)	179
13.4.3.15	OR_BW (ODER, bitweise)	179
13.4.3.16	XOR_BW (XOR, bitweise)	180
13.4.3.17	NOT_BW (NOT, bitweise)	180
13.4.3.18	Restart_Interlock	181
13.4.3.19	FW_Check	182
13.4.3.20	Path Speed	183
13.4.3.21	Speed	185
13.4.3.22	Direction	187
13.4.4	Vergleichende Funktionsblöcke	189
13.4.4.1	GT (größer als)	189
13.4.4.2	GT_NS (größer als, not safe)	190
13.4.4.3	GE (größer gleich)	190
13.4.4.4	GE_NS (größer gleich, not safe)	190
13.4.4.5	LT (kleiner als)	191
13.4.4.6	LT_NS (kleiner als, not safe)	191
13.4.4.7	LE (kleiner gleich)	191
13.4.4.8	LE_NS (kleiner gleich, not safe)	192
13.4.4.9	EQ (gleich)	192
13.4.4.10	EQ_NS (gleich, not safe)	192
13.4.4.11	MIN (Minimum)	193
13.4.4.12	MIN_NS (Minimum, not safe)	193
13.4.4.13	MAX (Maximum)	194
13.4.4.14	MAX_NS (Maximum, not safe)	194
13.4.4.15	LIMIT (innerhalb Grenzen)	195
13.4.4.16	LIMIT_NS (innerhalb Grenzen, not safe)	195
13.4.4.17	SDINT_Compare	196
13.4.4.18	DINT_Compare	197

13.4.5	Logical-Funktionsblöcke	198
13.4.5.1	AND	198
13.4.5.2	AND_5I	199
13.4.5.3	AND_NS - AND NOT SAFE	199
13.4.5.4	NOT	200
13.4.5.5	OR	200
13.4.5.6	OR_5I	201
13.4.5.7	XOR	201
13.4.5.8	AND_XI	202
13.4.5.9	OR_XI	203
13.4.6	Timer-Funktionsblöcke	204
13.4.6.1	TOFF	204
13.4.6.2	TON	205
13.4.6.3	TP	205
13.4.7	Trigger-Funktionsblöcke	205
13.4.7.1	F_TRIG	206
13.4.7.2	R_TRIG	206
13.4.8	Counter-Funktionsblöcke	206
13.4.8.1	CTD - Abwärtszähler	207
13.4.8.2	CTU - Aufwärtszähler	207
13.4.8.3	CTUD - Auf-/ Abwärtszähler	208
13.4.9	FlipFlop-Funktionsblöcke	208
13.4.9.1	RS_Flipflop	209
13.4.9.2	SR_Flipflop	209
13.5	Komplexe Funktionsblöcke	210
13.5.1	Collision_Detection-Funktion	211
13.5.1.1	Funktionalitäten	212
13.5.1.2	Arbeitsraum	212
13.5.1.3	Tool Center Point und Hüllkugel	212
13.5.1.4	Erstellung eines komplexen Arbeitsraums	213
13.5.1.5	Schnittstellen	214
13.5.1.6	Beispiel zur Beschaltung des Funktionsblock	214
13.5.2	Safe_Area-Funktion	215
13.5.2.1	Funktionalitäten	216

13.5.2.2	Arbeitsraum	216
13.5.2.3	Tool Center Point und Hüllkugel	217
13.5.2.4	Schnittstellen	218
13.5.2.5	Beispiel zur Beschaltung des Funktionsblock	219
13.5.3	SF_Antivalent-Funktion	220
13.5.3.1	Fehlererkennung	221
13.5.3.2	Fehlerverhalten	221
13.5.3.3	Funktionsblock-spezifische Fehler und Status-Codes	221
13.5.3.4	Status Diagramm SF_Antivalent	222
13.5.3.5	Zeit-Diagramm SF_Antivalent	223
13.5.4	SF_EDM-Funktion	224
13.5.4.1	Fehlererkennung	225
13.5.4.2	Fehlerverhalten	225
13.5.4.3	Besonderheiten	228
13.5.4.4	Status Diagramm SF_EDM	229
13.5.4.5	Zeit-Diagramm SF_EDM	230
13.5.5	SF_EmergencyStop-Funktion	231
13.5.5.1	Fehlererkennung	232
13.5.5.2	Fehlerverhalten	233
13.5.5.3	Status Diagramm SF_EmergencyStop	234
13.5.5.4	Zeit-Diagramm SF_EmergencyStop	235
13.5.6	SF_EnableSwitch-Funktion	236
13.5.6.1	Fehlererkennung	238
13.5.6.2	Fehlerverhalten	238
13.5.6.3	Status Diagramm SF_EnableSwitch	240
13.5.6.4	Zeit Diagramm SF_EnableSwitch	241
13.5.7	SF_Equivalent-Funktion	242
13.5.7.1	Fehlererkennung	243
13.5.7.2	Fehlerverhalten	243
13.5.7.3	Status Diagramm SF_Equivalent	244
13.5.7.4	Zeit-Diagramm SF_Equivalent	245
13.5.8	SF_ESPE-Funktion	246
13.5.8.1	Fehlererkennung	247
13.5.8.2	Fehlerverhalten	248

13.5.8.3	Status Diagramm SF_ESPE	249
13.5.8.4	Zeit-Diagramm SF_ESPE	250
13.5.9	SF_GuardLocking-Funktion	251
13.5.9.1	Betriebssequenz	252
13.5.9.2	Fehlererkennung	253
13.5.9.3	Fehlerverhalten	254
13.5.9.4	Status Diagramm SF_GuardLocking	256
13.5.9.5	Zeit-Diagramm SF_GuardLocking	257
13.5.10	SF_GuardMonitoring-Funktion	258
13.5.10.1	Fehlererkennung	259
13.5.10.2	Fehler- und Reset-Verhalten	260
13.5.10.3	Status-Diagramm SF_GuardMonitoring	262
13.5.10.4	Zeit-Diagramm SF_GuardMonitoring	263
13.5.11	SF_ModeSelector-Funktion	264
13.5.11.1	Fehlererkennung	266
13.5.11.2	Fehlerverhalten	266
13.5.11.3	Besonderheiten	267
13.5.11.4	Status-Diagramm SF_ModeSelector	268
13.5.11.5	Zeit-Diagramm SF_ModeSelector	268
13.5.12	SF_MutingPar-Funktion	270
13.5.12.1	Fehlererkennung	272
13.5.12.2	Fehlerverhalten	272
13.5.12.3	Status Diagramm SF_MutingPar	277
13.5.12.4	Zeit-Diagramm SF_MutingPar	278
13.5.13	SF_MutingPar_2Sensor-Funktion	279
13.5.13.1	Fehlererkennung	281
13.5.13.2	Fehlerverhalten	281
13.5.13.3	Status-Diagramm SF_MutingPar_2Sensor	284
13.5.13.4	Zeit-Diagramm SF_MutingPar_2Sensor	284
13.5.14	SF_MutingSeq-Funktion	285
13.5.14.1	Fehlererkennung	287
13.5.14.2	Fehlerverhalten	287
13.5.14.3	Status-Diagramm SF_MutingSeq	290
13.5.14.4	Zeit-Diagramm SF_MutingSeq	291

13.5.15	SF_Optional_Pwd-Funktion	292
13.5.15.1	Fehlererkennung	293
13.5.15.2	Fehlerverhalten	293
13.5.15.3	Status-Diagramm SF_Optional_Pwd	295
13.5.15.4	Zeit-Diagramm SF_Optional_Pwd	296
13.5.16	SF_Optional_PwdII-Funktion	297
13.5.16.1	Fehlererkennung	298
13.5.16.2	Fehlerverhalten	298
13.5.16.3	Status-Diagramm SF_Optional_PwdII	301
13.5.16.4	Zeit-Diagramm SF_Optional_PwdII	301
13.5.16.5	Sicherheitsmaßnahmen und -hinweise	301
13.5.17	SF_Optional_Switch-Funktion	303
13.5.17.1	Wertetabelle	303
13.5.18	SF_Optional_Switch_SDint	304
13.5.19	SF_Optional_Switch_Dint	305
13.5.20	SF_OutControl-Funktion	306
13.5.20.1	Allgemeines	307
13.5.20.2	Optionale Bedingungen zur Prozesskontrolle (ProcessControl)	307
13.5.20.3	Optional startup inhibits	307
13.5.20.4	Fehlererkennung	307
13.5.20.5	Fehlerverhalten	307
13.5.20.6	Status-Diagramm SF_OutControl	309
13.5.20.7	Zeit-Diagramm SF_OutControl	310
13.5.21	SF_SafetyRequest-Funktion	311
13.5.21.1	Fehlererkennung	312
13.5.21.2	Fehlerverhalten	312
13.5.21.3	Status-Diagramm SF_SafetyRequest	314
13.5.21.4	Zeit-Diagramm SF_SafetyRequest	315
13.5.22	SF_TestableSafetySensor-Funktion	316
13.5.22.1	Test-Modus	318
13.5.22.2	Optionale Startup inhibits	318
13.5.22.3	Fehlererkennung	318
13.5.22.4	Fehlerverhalten	318
13.5.22.5	Status-Diagramm SF_TestableSafetySensor	323

13.5.22.6	Zeit-Diagramm SF_TestableSafetySensor	324
13.5.23	SF_TwoHandControlTypell-Funktion	325
13.5.23.1	Fehlererkennung	325
13.5.23.2	Fehlerverhalten	325
13.5.23.3	Status-Diagramm SF_TwoHandControlTypell	328
13.5.23.4	Zeit-Diagramm SF_TwoHandControlTypell	329
13.5.24	SF_TwoHandControlTypeIII-Funktion	330
13.5.24.1	Fehlererkennung	330
13.5.24.2	Fehlerverhalten	330
13.5.24.3	Besonderheiten	332
13.5.24.4	Status Diagramm SF_TwoHandControlTypeIII	333
13.5.24.5	Zeit-Diagramm SF_TwoHandControlTypeIII	334
13.5.25	SF_HGW_Login	335
13.5.25.1	Fehlererkennung	337
13.5.25.2	Fehlerverhalten	337
13.5.25.3	SF_HGW_Login Status Diagram	341
13.5.26	Komplexe numerische Funktionsblöcke	341
13.5.26.1	SF_LimitComparator	341
13.5.26.2	SF_LimitComparator_Dint	345
13.5.26.3	SF_Numeric_Selector	346
13.5.26.4	SF_RampMonitor	350
13.5.26.5	SF_DirectionMonitor	354
13.5.26.6	SF_SkewMonitor	358
13.6	Funktionsblock-Makros	362
13.6.1	Beschreibung	362
13.6.2	Begriffe	362
13.6.3	Erstellen von Makros	363
13.6.3.1	Zusammenstellen eines Makros	363
13.6.3.2	Erstellen eines Makros aus vorhandenen Elementen	365
13.6.3.3	Einfügen der Anschlüsse in das Makro-Netzwerk	366
13.6.3.4	Erstellen von eingebetteten Makros	368
13.6.3.5	Erstellen einer Kopie eines Makros	368
13.6.4	Makro-Bibliotheken	369
13.6.4.1	Erstellen von Makro-Bibliotheken	369

13.6.4.2	Speichern von Makro-Bibliotheken	370
13.6.4.3	Anzeigen von Makro-Bibliotheken	371
13.6.4.4	Verwenden von Makros aus Bibliotheken	371
13.6.4.5	Lokale Makro-Bibliothek	372
13.6.4.6	Sperren von Makro-Bibliotheken	373
13.6.4.7	Vergleich von Makros	375
13.6.5	Debuggen von Funktion Block Macros	377
14.	Safety-Module	378
14.1	Beschreibung Safety-CPU-Modul XS-102-C00-000	378
14.2	Beschreibung Safety Analog-Eingangsmodul XS-322-4AI-I2 ...	379
14.3	Beschreibung Safety Digital-Eingangsmodul XS-322-10DI-PF ..	380
14.4	Beschreibung Safety Digital-Ausgangsmodul XS-322-8DO-P20	381
14.5	Beschreibung Safety Digital-Ein-/Ausgangsmodul XS-322-8DIO-PF20	382
14.6	Beschreibung Safety Relais-DC-Ausgangsmodul XS-322-2DO-RNODC	383
14.7	Beschreibung Safety Relais-Ausgangsmodul XS-322-2DO-RNOAC	384
14.8	Beschreibung Safety Inkrementalgeber-Modul XS-322-2INC ...	385
15.	Fehler-Codes	386
	Anhang	446
A.1	Weitere Nutzungsinformationen	447
A.2	Shortcuts - Tastenkombinationen und Sondertasten	448
A.2.1	Funktionstasten und Tastenkombinationen	448
A.2.1.1	Tastenkombinationen im Menü FILE	448
A.2.1.2	Tastenkombinationen im Menü EDIT	448
A.2.1.3	Funktionstaste im Menü BUILD	448
A.2.1.4	Funktionstasten und Tastenkombination im Menü DEBUG	448
	Stichwortverzeichnis	449

0.1 Zu dieser Dokumentation

Die vorliegende Dokumentation beinhaltet die Informationen, die für einen korrekten und sicheren Umgang mit der modularen Sicherheitssteuerung XN Safety notwendig sind.

Diese Anwenderhilfegilt als Bestandteil der Programmiersoftware **XN Safety Designer**. Die Programmiersoftware **XN Safety Designer** ist integrierter Bestandteil des modularen Sicherheitssteuerungssystem XN Safety. Mit dieser Software werden die zugehörigen Safety-Projekte erstellt.

Arbeiten Sie mit der aktuellsten Dokumentation zu den XN Safety-Modulen, siehe → Abschnitt "Weitere Nutzungsinformationen", Seite 447

Bitte senden Sie Ihre Kommentare, Empfehlungen oder Anregungen zu diesem Dokument an: DocumentationEGBonn@eaton.com

0.1.1 Änderungsprotokoll

Gegenüber den früheren Ausgaben hat es folgende wesentliche Änderungen gegeben:

Redaktionsdatum	Seite	Stichwort	neu	Änderung	entfällt
01/2025	ff.	Ausgabe zur Zertifizierung	✓		
06/2025	ff.	Auflage zur Produkteinführung	✓		

0.1 Zu dieser Dokumentation

0.1.2 Zielgruppe

Diese Anwenderhilfe enthält alle Informationen, die Sie für den SafetyDesigner_DE zur Konfiguration der Safety-Module benötigen.

Diese Beschreibung richtet sich an Fachkräfte der Elektrotechnik und an Personen, die mit elektrotechnischen Installation vertraut sind und die das XN Safety Designer einsetzen möchten:

- Projektplaner, Automatisierungstechniker und -ingenieure
- Monteure, Inbetriebnahmetechniker
- Maschinenbediener
- Instandhalter/Prüftechniker

Es werden fundierte Kenntnisse auf dem Gebiet der Automatisierungstechnik vorausgesetzt.

Sie erhalten weitere Hilfe sowie Informationen und passendes Zubehör auf unserer Website Eaton.com.

Bei Fragen steht Ihnen natürlich auch gerne unser Support-Team zur Verfügung. Notfalltelefon sowie Geschäftszeiten entnehmen Sie bitte unserer Website.

Ein XN Safety-Modul darf nur von Fachkräften der Elektrotechnik und Personen, die mit elektrotechnischen Installation vertraut sind, montiert und angeschlossen werden.



VORSICHT



Installation erfordert Elektro-Fachkraft



Beachten Sie die Sicherheitsvorschriften zum modularen Sicherheitssteuerung XN Safety !

Vor dem Arbeiten mit dem XN Safety CPU-Modul muss das Kapitel zu den Sicherheitsvorschriften von allen Personen, die mit dem Safety-Modulen arbeiten, gelesen und verstanden worden sein.

0.1.3 Haftungsausschluss

Alle Angaben in diesem Handbuch wurden nach bestem Wissen und Gewissen sowie nach dem Stand der Technik gemacht. Dennoch können Unrichtigkeiten nicht ausgeschlossen werden. Technische Änderungen, die der Verbesserung der Geräte dienen, sind vorbehalten.

Für die Richtigkeit und Vollständigkeit der Angaben wird keine Haftung übernommen. Die vorliegende Betriebsanleitung stellt eine reine Produktbeschreibung dar. Die Angaben enthalten insbesondere keine Zusicherung bestimmter Eigenschaften im Sinne des Gewährleistungsrechts.

Der Maschinenhersteller ist für den sachgemäßen Einbau sowie die Produktkonfiguration verantwortlich. Der Maschinenbediener ist für einen sicheren Umgang sowie die sachgemäße Bedienung verantwortlich.

XN Safety darf nur in Kenntnis und Verständnis dieses Handbuches betrieben werden.

Die Kenntnisse aus den Handbüchern zur Implementierung der Sicherheitssteuerung in den Automatisierungsprozess werden vorausgesetzt.

Sofern die sicherheitsrelevanten Hinweise nicht beachtet werden, insbesondere die Installation und Inbetriebnahme der Sicherheitssteuerung durch nicht hinreichend qualifiziertes Personal erfolgt oder die Sicherheitssteuerung sachwidrig verwendet werden, können von der Steuerung ausgehende Gefahren nicht ausgeschlossen werden. Für Schäden, die aufgrund einer Nichtbeachtung dieser Anleitungen oder der jeweiligen Vorschriften entstehen, übernimmt Eaton keine Haftung.

Für die Nutzung von Beispielprogrammen sowie für die Verwendung von in der Dokumentation verwendeten Abbildungen von Programmteilen gelten folgende Hinweise und Nutzungsregeln:

1. Die zur Verfügung gestellten Beispielprogramme wurden nach bestem Wissen und Gewissen sowie unter Berücksichtigung des heutigen Standes der Technik erstellt. Dennoch können Fehler nicht ausgeschlossen werden und die bereitgestellten Beispielprogramme decken nicht sämtliche Funktionsbausteine und Anwendungen ab, die für die Sicherheitssteuerung zur Verfügung stehen.
2. Für die Programmerstellung und Inbetriebnahme der Sicherheitssteuerung werden elektrotechnische Fachkenntnisse vorausgesetzt. Ist die Sicherheitssteuerung falsch angeschlossen oder fehlerhaft konfiguriert und aktive Komponenten wie Motoren oder Druckzylinder werden angesteuert, sind Personen und/oder Anlagenteile gefährdet.
3. Bei der Nutzung der zur Verfügung gestellten Beispielprogramme und bei der Programmerstellung mit dem XN Safety Designer sind von Ihnen eigenverantwortlich einzuhalten:

0.1 Zu dieser Dokumentation

- Sämtliche relevanten Regeln zur Erstellung von Schaltplänen für die Sicherheitssteuerung gemäß den jeweils aktuellen Dokumenten zu XN Safety.
 - Sämtliche für die Inbetriebnahme, Schaltplan-Erstellung und den Einsatz der Sicherheitssteuerung für die von Ihnen geplante Anwendung maßgeblichen Richtlinien, Normen und Vorschriften der Arbeitssicherheit und der Unfallverhütung, insbesondere von Berufsgenossenschaften.
 - Der anerkannte Stand von Wissenschaft und Technik.
 - Alle sonstigen allgemeinen Sorgfaltspflichten zur Verhütung von Schäden an Leib, Leben und Gesundheit von Personen und von Sachschäden.
4. Der Hersteller übernimmt keine Haftung für Schäden, gleich welcher Art, die dadurch verursacht werden, dass Kunden die zur Verfügung gestellten Beispielprogramme entgegen den Nutzungsbedingungen, die hier unter Ziffer 1 bis 3 aufgeführt sind, eingesetzt haben.



Das aktuelle Handbuch ist verfügbar im Download Center,
Suchwort: MN050028.



Eaton.com/documentation

0.1.4 Kurzbezeichnungen

Nachfolgend werden die folgenden Kurzbezeichnungen eingesetzt:

Kurzbezeichnung	Erklärung
XN Safety	gesamte Serie, Zusammenfassung aller Module in der Produktfamilie
Safety-CPU, SafeCPU	XN Safety CPU-Modul XS-102-C00-000
XN Safety-Modul	Erweiterung zum XN Safety CPU-Modul
XN Safety Designer	XN Safety Designer



Die genaue Bezeichnung Ihrer Safety-Module entnehmen Sie dem Aufdruck auf den Modulen.

0.1.5 Lesekonventionen

Tab. 1: Darstellungsmittel in dieser Dokumentation

Auszeichnung	Bedeutung
dickengleicheSchrift	kennzeichnet Display-Anzeigen, Elemente auf Datei-Ebene, Quellcode Befehlszeilen
	gibt die Tasten-Beschriftungen an, am Gerät und in der Software
Beschriftung	gibt die Beschriftungen in der Software-Oberfläche an
Menüpfad\Untermenü...\Eintrag	Pfad-Angaben zu Ansichten und Dialogen in der Software
Menü/Befehl	Kennzeichnet einen Befehl aus einem Menü
<name>	Spitze Klammern kennzeichnen variable Werte, für die Sie eigene Werte einsetzen müssen

0.1.5.1 Warnhinweise

Warnung vor Personenschäden



GEFAHR

Gefahr bedeutet, dass der Tod oder schwere Verletzungen eintreten, wenn die angegebenen Maßnahmen nicht getroffen werden.

- Beachten Sie alle Hinweise, um Tod oder schwere Verletzungen zu vermeiden.



WARNUNG

Warnung bedeutet, dass der Tod oder schwere Verletzungen eintreten können, wenn die angegebenen Maßnahmen nicht getroffen werden.

- Beachten Sie alle Hinweise, um Tod oder schwere Verletzungen zu vermeiden.



VORSICHT

Vorsicht bedeutet, dass mittelschwere bis leichte Verletzungen eintreten können, wenn die angegebenen Maßnahmen nicht getroffen werden.

- Beachten Sie alle Hinweise, um mittelschwere bis leichte Verletzungen zu vermeiden.

Warnung vor Sachschäden



ACHTUNG

warnt vor möglichen Sachschäden.

Verbote



Verbot

Verbotszeichen untersagen Handlungen oder den Gebrauch von bestimmten Gegenständen

Gebote



Gebot

Gebotszeichen fordern zu einem bestimmten Verhalten auf

0.1 Zu dieser Dokumentation

Hinweise



zusätzliche Informationen, Wissenswertes, nützliches Zusatzwissen

Liefert wichtige Hinweise über das Produkt, die Handhabung oder relevante Teile der Dokumentation, auf welche besonders aufmerksam gemacht werden soll.



▶ zeigt Handlungsanweisungen an

0.1.5.2 Weitere Nutzungsinformationen

Dokumente, wie zum Beispiel Handbücher, werden mit dem entsprechenden Namen und der Eaton-Nummer aufgeführt hinter dem Symbol .



Titel der Publikation

zur Identifizierung die Eaton Publikationskennung

Verlinkungen zu externen Internet-Adressen, diese werden hinter dem Symbol  angezeigt.



Zieladresse ohne http(s)://www.

Links im Text zu externen Inhalten werden **blau** angezeigt.

1. Bedienung Programmiersoftware XN Safety Designer

1.1 Begriffsklärung: Projekt

Mit dem XN Safety Designer kann ein Projekt erstellt oder geladen werden. Ein Projekt enthält die Konfigurationsdaten des gesamten Safety-Systems. Außerdem sind alle Safety-Module (Safety-CPU und Safety-IO-Module) in dem Projekt enthalten. Eine Änderung des Projekts kann nur offline erfolgen, d.h. sobald eine Online-Verbindung zur Safety-CPU (XS-102-C00-000) besteht, kann das Projekt so lange nicht geändert werden bis die Online-Verbindung geschlossen wird.

Bei der Safety-CPU XS-102-C00-000 im Stand-Alone-Betrieb fungiert die jeweilige Safety-CPU als Root-Element.

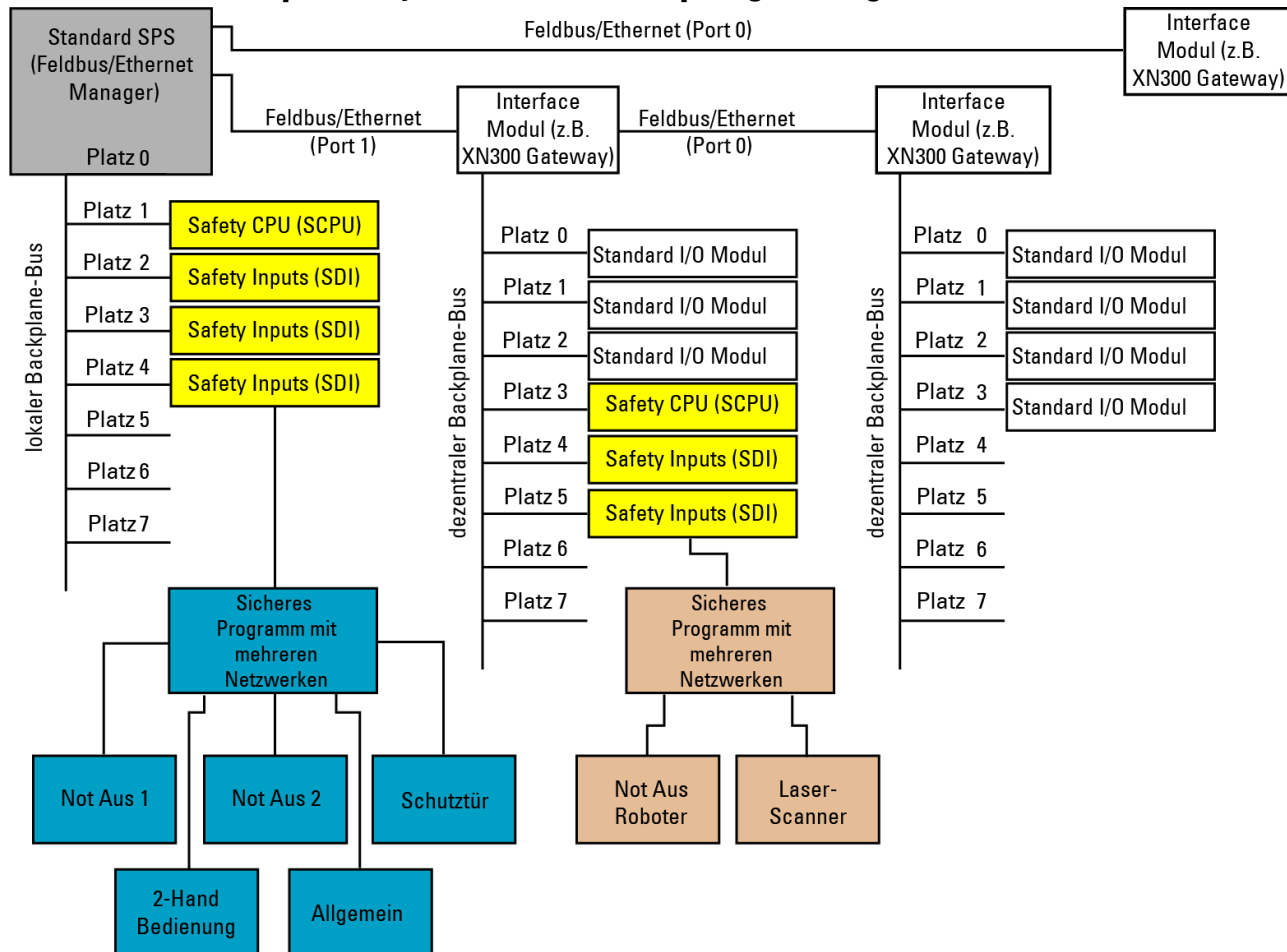
Pro Safety-CPU wird ein Programm abgearbeitet das wiederum aus mehreren graphischen Netzwerken (Teilprogramme) bestehen kann.

Bei der Safety-CPU XS-102-C00-000 können max. 16 Erweiterungsmodule (Eingangskarten, Ausgangskarten, Relais, ...) angesteckt werden. Das Programm kann aus mehreren graphischen Netzwerken (Teilprogrammen) bestehen.

1. Bedienung Programmiersoftware XN Safety Designer

1.2 Übersichtsbeispiel Projekt, Hardware-Topologie, Programmnetzwerke

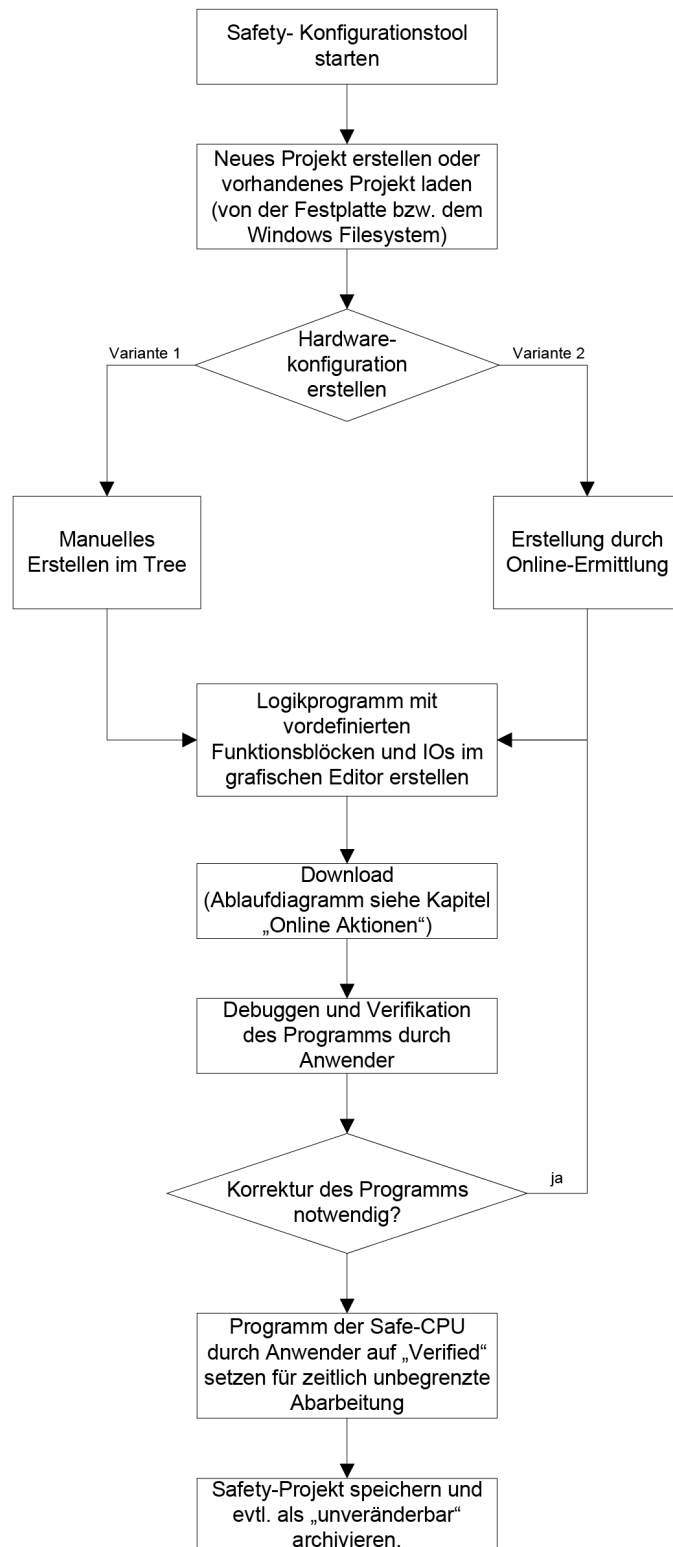
1.2 Übersichtsbeispiel Projekt, Hardware-Topologie, Programmnetzwerke



1.3 Schematische Darstellung einer Projekterstellung

1. Bedienung Programmiersoftware XN Safety Designer

1.3 Schematische Darstellung einer Projekterstellung

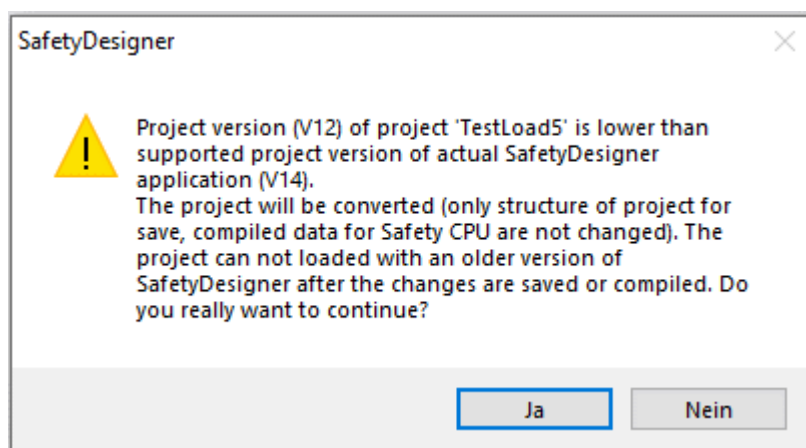


1. Bedienung Programmiersoftware XN Safety Designer

1.4 Laden von Projekten mit einer neueren Version von XN Safety Designer

1.4 Laden von Projekten mit einer neueren Version von XN Safety Designer

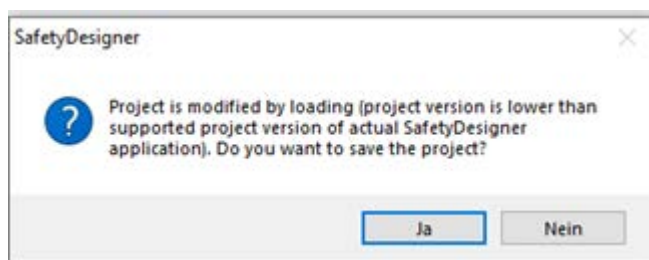
Projekte, welche mit der ersten Version des XN Safety Designers erstellt wurden, können mit einer späteren Version geladen werden. Beim Laden wird der Anwender darauf hingewiesen, dass sich die Version von XN Safety Designer geändert hat und sich das Projekt nach dem Speichern nicht mehr in der alten Version laden lässt.



Der Anwender kann entscheiden, ob er das Projekt laden will.

Die Unterschiede zwischen den Versionen von XN Safety Designer beziehen sich nur auf die Struktur des Projektaufbaus (z.B. es gibt zusätzliche Einstellmöglichkeiten im Property Browser). Das Kompilat und somit die CRC des Kompilats auf der Safety-CPU ändert sich nicht.

Nimmt der Anwender keine Änderungen im Projekt vor, wird beim Speichern bzw. Kompilieren des Projektes gefragt, ob gespeichert werden soll.



Zusammenhang zwischen Projektversion und Versionsnummer

Projektversion	XN Safety Designer	Bemerkung
V15	11.01.076	Erstes Build

2. Bedingungen und Anforderungen für den Anwender

1. Der XN Safety Designer ist bestimmungsgemäß zu verwenden. Darunter ist unter anderem zu verstehen, dass alle Parameter der einzelnen Funktionsblöcke, Safety-Gruppen, Konstanten usw. vom Anwender immer auf Richtigkeit geprüft werden.
2. Ein weiterer wichtiger Punkt ist, dass ein Projekt in sich geschlossen ist. Darunter ist zu verstehen, dass ein Safety-Projekt nicht auf Safety-Bausteine eines anderen Projekts zugreifen darf, z.B. eine Safety-CPU von Projekt A greift auf ein Safety-Input-Modul von Projekt B zu. Dies ist verboten!
3. Der Anwender darf kein eigenständiges Update der Firmware an den XN Safety-Modulen vornehmen. Dies ist verboten!
4. Während des Betriebs des Sicherheitssystems ist die Verbindung zum XN Safety Designer zu trennen.
Der XN Safety Designer darf nur zur Konfiguration bzw. zu Diagnosezwecken mit dem Sicherheitssystem verbunden sein.

3. Erstellen der Hardware-Konfiguration

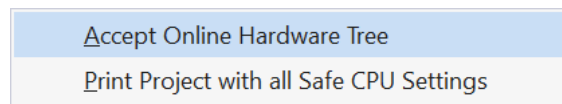
3.1 Ermittlung der Hardware-Konfiguration durch Online-Verbindung

3. Erstellen der Hardware-Konfiguration

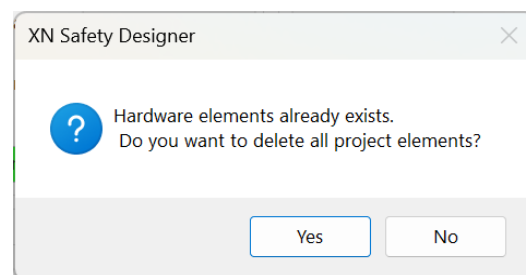
3.1 Ermittlung der Hardware-Konfiguration durch Online-Verbindung

Die physikalisch vorhandenen Hardware-Module können durch Herstellung einer Online-Verbindung zur Steuerung ausgelesen werden. Die ausgelesenen Module können über ein Kontextmenü im Hardware Tree in ein leeres Safety-Projekt übernommen werden.

Klicken Sie hierfür im Hardware Tree mit der rechten Maustaste und wählen Sie aus dem Kontextmenü „Accept Online Hardware Tree“.



Bei einem bestehenden Safety-Projekt (z.B. es ist bereits ein Hardware Tree vorhanden), wird dem Anwender mitgeteilt, dass das bestehende Projekt gelöscht wird. Der Anwender kann das Löschen des bestehenden Projekts abbrechen.



Nach dem Ausführen des Kommandos sind durch Öffnen der Unterordner im Hardware Tree die eingefügten Module ersichtlich. Es werden nur die Module eingefügt, die eine Relevanz für das Safety-Projekt haben (Safety-Module, Interface-Module).

Der Anwender muss den Hardware Tree entsprechend anpassen, bevor eine Applikation erstellt werden kann.

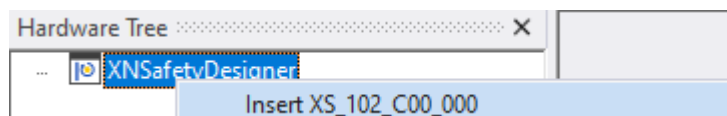
Die Verbindung wird über die USB-Schnittstelle der Safety-CPU hergestellt.

3. Erstellen der Hardware-Konfiguration

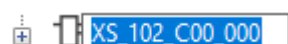
3.2 Manuelles Editieren einer XS-102-C00-000 im Stand-Alone-Betrieb

3.2 Manuelles Editieren einer XS-102-C00-000 im Stand-Alone-Betrieb

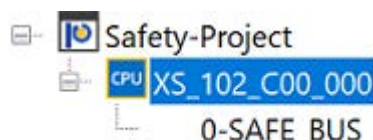
Es besteht die Möglichkeit eine XS-102-C00-000 im Stand Alone-Betrieb zu konfigurieren. Mit dieser Methode kann eine Hardware-Konfiguration ohne Online-Verbindung zum Modul XS-102-C00-000 erstellt werden. Der Aufbau des Baums erfolgt über das Kontextmenü im Hardware Tree, welches Sie über einen Klick mit der rechten Maustaste erreichen. Als erster Schritt muss beim Root-Element im Hardware Tree über das Kontextmenü die XS-102-C00-000 eingefügt werden.



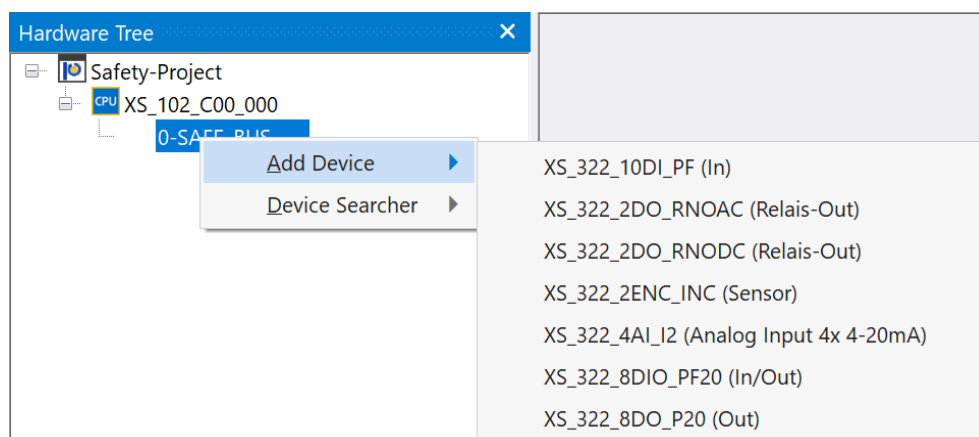
Nach dem Ausführen des Befehls werden Sie aufgefordert einen Namen für die XS-102-C00-000 in das Edit-Feld einzugeben, bzw. den vorgeschlagenen Namen zu übernehmen.



Als nächstes wird eine Baumstruktur mit einem Erweiterungs-Steckplatz aufgeklappt. Es können bis zu 16 Erweiterungsmodule eingefügt werden. Es wird immer nur ein leerer Erweiterungs-Steckplatz am Ende angezeigt; sind alle 16 Plätze belegt, wird kein leerer Steckplatz mehr angezeigt.



Mittels Kontextmenü können die leeren Elemente im Baum mit Modulen belegt werden. Es werden nur jene Module angezeigt, die für diese Position erlaubt sind.



Sie können auf die beschriebene Weise Ihren gesamten Aufbau mit den benötigten Modulen zusammenstellen. Innerhalb des Hardware Trees lassen sich die Module per Drag & Drop an die gewünschte Position verschieben. Es wird über ein Cursor-

3. Erstellen der Hardware-Konfiguration

3.2 Manuelles Editieren einer XS-102-C00-000 im Stand-Alone-Betrieb

Symbol angezeigt, ob die neue Position für das Modul erlaubt ist (dunkles Rechteck für erlaubt, rotes Kreuz für verboten).

Wurde aus Versehen ein falsches Modul eingefügt, können Sie es mittels Kontextmenüpunkt „Delete Module“ oder mit der „[DEL]-Taste“ löschen.

Bei den Erweiterungsmodulen gibt es eine maximale Anzahl von Relais-Modulen und Sensoren. Es können max. 8 XS_322_2DO_RNOAC, XS_322_2DO_RNODC und/oder XS_322_2ENC_INC-Module platziert werden, da jedes dieser Module 2 Plätze belegt.

Nach dem Ausführen des Befehls zum Einfügen eines Moduls werden Sie aufgefordert, einen Namen für das Modul in das Edit-Feld einzugeben, bzw. den vorgeschlagenen Namen zu übernehmen.

Des weiteren können Sie bei Safety-Modulen mittels den Kontextmenüpunkten „Replace Device“ bzw. „Replace Searcher“ das jeweilige Modul (z.B. XS_322_10DI_PF auf XS_322_8DO_P20) austauschen. Es werden nur Module angezeigt, welche zum Austauschen erlaubt sind. Bestehende Elemente in den Netzwerken, welche durch das Ersetzen betroffen sind, werden überprüft und gegebenenfalls gelöscht. Der Anwender wird darüber informiert und muss das Ersetzen explizit bestätigen.

3.3 Einfaches Ermitteln der Daten auf der Safety-CPU

Die Namen und Versionsnummern der geladenen Projekte können ermittelt werden, indem eine Online-Verbindung zu einer Safety-PLC hergestellt wird, ohne dass im XN Safety Designer ein Projekt geladen ist. Es erscheint ein Fenster mit den physikalisch vorhandenen Modulen, wobei nach jeder Safety-CPU in eckigen Klammern der Name und die Versionsnummer des geladenen Projekts stehen. An der Farbe der Schrift kann man den Zustand der Safety-CPU ablesen:

- Grün = Operation Mode
- Violett = Temporary Operation Mode
- Gelb = Service Mode
- Rot = Error
- Grau = Sonstiges

Die Firmware-Version und die Sicherheitsnummer der Safety-CPU ist einfach zu ermitteln. Dazu muss der Online Hardware Tree geöffnet sein. Wenn im Online-Hardware Tree dann der Mauszeiger über die Safety-CPU bewegt wird, werden im Tooltip die besagten Daten angezeigt.

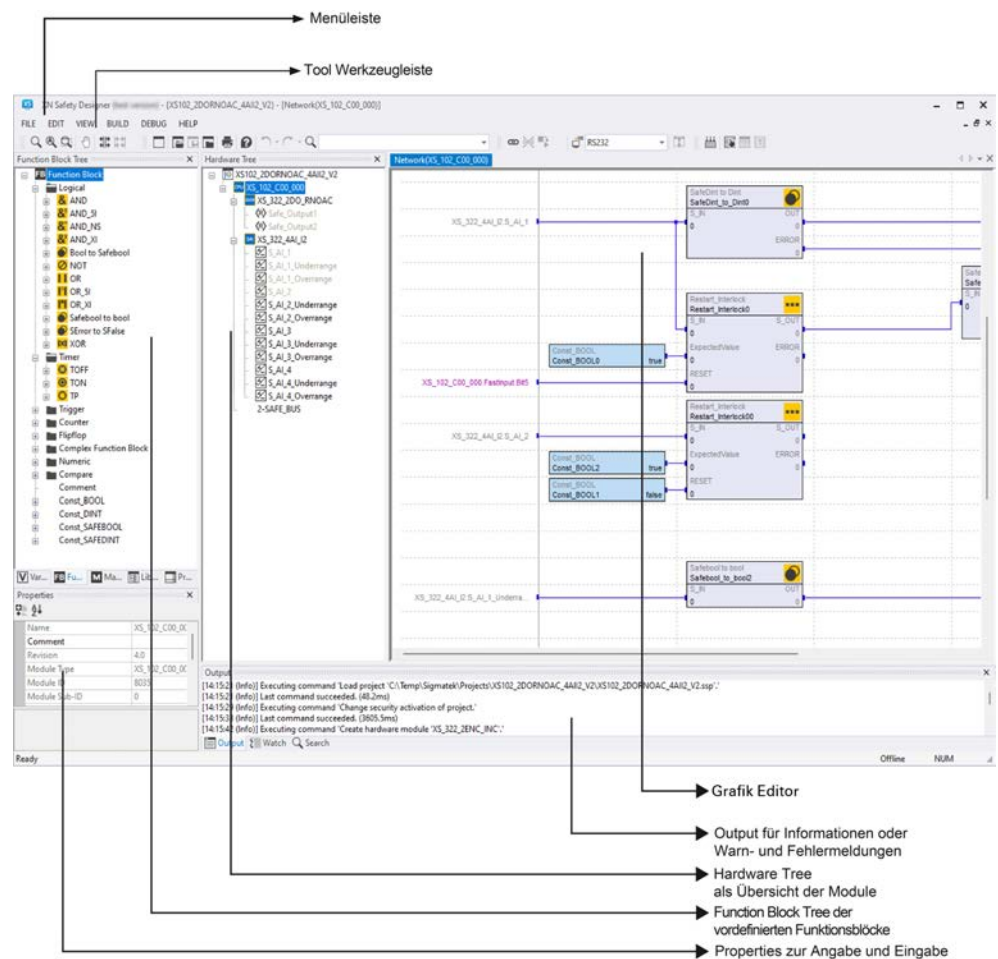
4. Grafik Editor

Der Grundaufbau und die Bedienung sind an die übliche Oberfläche angelehnt. Der Ablauf-Code für die Safety-CPU wird graphisch im Netzwerk der Safety-CPU erstellt. Die benötigten Parameter der einzelnen Elemente werden im Property Browser eingestellt.

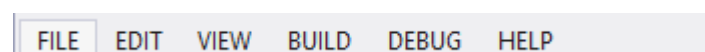


Änderungen am Projekt sind nur möglich, wenn keine Online-Verbindung zur XS-102-C00-000 hergestellt ist.

4.1 Oberfläche



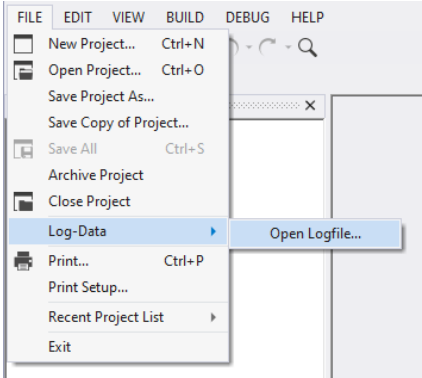
4.2 Menüleiste



4. Grafik Editor

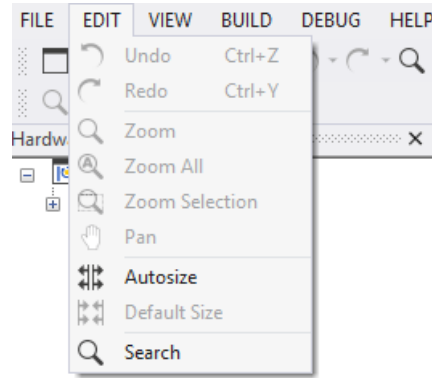
4.2 Menüleiste

4.2.1 File



Übersicht		
	New Project	Öffnet den Dialog Create New Project zum Erstellen eines neuen Projektes.
	Open Project	Öffnet ein bereits bestehendes Projekt. Im Dialog Open a Safety Project File kann das gewünschte Projekt (*.ssp) ausgewählt und geöffnet werden.
	Save Project As	Das derzeit geöffnete Projekt kann unter anderem Namen und anderem Pfad gespeichert werden. Das Projekt ist anschließend mit dem neuen Namen/Pfad geöffnet.
	Save Copy of Project	Das derzeit geöffnete Projekt kann unter anderem Namen und anderem Pfad gespeichert werden.
	Save All	Speichert das derzeit geöffnete Projekt.
	Archive Project	Archiviert das derzeit geöffnete Projekt. Nach dieser Aktion ist das Projekt nicht mehr veränderbar!
	Close Project	Schließt das derzeit geöffnete Projekt.
	Log Data	Gespeicherte Log-Dateien anzeigen. Im Untermenü werden vorhandene Log-Dateien des Projekts angezeigt.
	Print	Druckt das gesamte Projekt mit allen Netzwerken.
	Print Setup	Druckmenü-Einstellungen
	Recent Project List	Zeigt eine Liste der zuletzt geöffneten Projekte.
	Exit	Schließt den XN Safety Designer . Falls Änderungen vorgenommen wurden, die noch nicht gespeichert sind, erscheint vor dem Schließen eine Aufforderung das Projekt zu speichern.

4.2.2 Edit

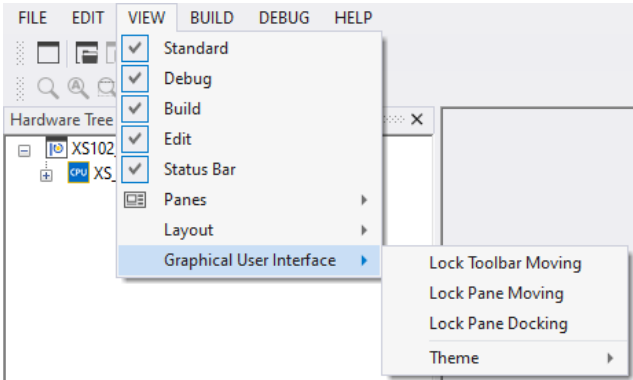


Übersicht		
	Undo	Macht den letzten Vorgang (die letzte Aktion) rückgängig.
	Redo	Stellt die letzte rückgängig gemachte Aktion wieder her.
	Zoom	Zoom-Werkzeug zum Vergrößern bzw. Verkleinern von Objekten im Netzwerk.
	Zoom All	Zoomt alle Objekte im Netzwerk so, dass alle sichtbar sind.
	Zoom Selection	Zoomt einen gewünschten Bereich im Netzwerk.
	Pan	Handwerkszeug zum Navigieren im Netzwerk.
	Autosize	Spaltenbreite eines Netzwerks automatisch berechnen.
	Default Size	Spaltenbreite des Netzwerks auf den Default-Wert setzen.
	Search	Ein Dialog für die globale Suchfunktion wird geöffnet.

4. Grafik Editor

4.2 Menüleiste

4.2.3 View



Übersicht		
	Standard Debug Build Edit Status Bar	Aktivieren / Deaktivieren Sie das jeweilige Häkchen um die einzelnen Werkzeugleisten (Standard, Debug, Build, Edit und Status Bar) anzuzeigen oder auszublenden. Eine übersichtliche Auflistung und Definition der Werkzeugleisten finden Sie unter → Abschnitt "Werkzeug-Symboleiste", Seite 38.
	 Panes	Aktivieren / Deaktivieren Sie das jeweilige Häkchen, um folgende Fenster anzuzeigen oder auszublenden. Unter → Abschnitt "Tree-Ansichten", Seite 40 sind die einzelnen Fenster genauer erklärt. Function Block Macro Tree Pane Library Tree Pane Variable Tree Pane Online Hardware Tree Pane Function Block Pane Hardware Tree Pane Project Tree Pane Properties Pane Watch Pane Output Pane
	Layout	Menüpunkt zum Laden und Speichern der Fenster Layouts. Reset Pane Layout Export Pane Layout Import Pane Layout

Übersicht

Graphical
User Inter-
face

Aktivieren / deaktivieren Sie das jeweilige Häkchen, um folgende Befehle zu sperren oder freizugeben.

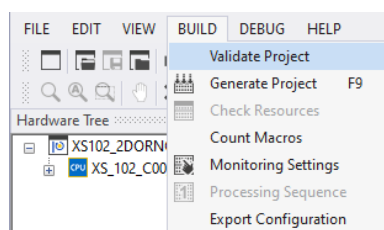
Lock Toolbar Moving

Lock Pane Moving

Lock Pane Docking

Theme – visuelle Gestaltung der Oberfläche, fest eingestellt auf Eaton.

4.2.4 Build



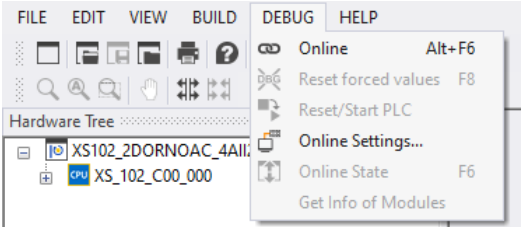
Übersicht

	Validate Project	Überprüft (validiert) das aktuell geöffnete Projekt.
	Generate Project	Generiert die Download-Daten des Projekts.
	Check Resources	Öffnet einen Dialog mit voraussichtlichem Speicherverbrauch, Zykluszeiten und Reaktionszeiten.
	Count Macros	Ermittelt alle Verwendungen von Funktionsblöcken und Macros und gibt die Anzahl der Verwendungen aus.
	Monitoring Settings	Öffnet einen Dialog und berechnet die voraussichtlichen Zykluszeiten und Übertragungszeiten.
	Processing Sequence	zeigt die Abarbeitungsreihenfolge in Netzwerken an, erst verfügbar, wenn das Projekt erfolgreich kompiliert ist
	Export Configuration	Exportiert eine Binärdatei mit den Download-Daten.

4. Grafik Editor

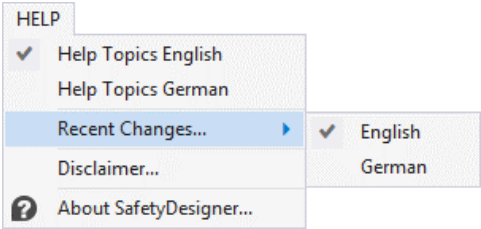
4.2 Menüleiste

4.2.5 Debug



Übersicht		
	Online	Stellt eine Online-Verbindung zwischen dem PC (Projekt) und der Steuerung her und generiert die Download-Daten.
	Reset all forced values	Setzt die geforcten Werte auf der Safety-CPU zurück.
	Reset/Start PLC	Reset und Start des PLC vor dem Start der Safety-Applikation.
	Online Settings...	Öffnet das Fenster XN Online configurations.
	Online State	Öffnet und schließt die Online Statusanzeige.

4.2.6 Help



Übersicht		
	Help Topics English	Startet die XN Safety Designer Hilfe in Englisch.
	Help Topics German	Startet die XN Safety Designer Hilfe in Deutsch.
	Recent Changes...	Öffnet eine Release Note-Datei mit den Änderungen dieser Version in Deutsch oder Englisch.
	Disclaimer...	Haftungsausschluss
	About XN Safety Designer...	Zeigt ein Informationsfenster über den XN Safety Designer mit Datum und Versionsstand an.

4.3 Werkzeug-Symbolleiste

Übersicht			
	New Project	Erstellt ein neues Projekt (öffnet das Fenster <i>Create New Project</i>).	Standard-Werkzeugleiste
	Open Project	Öffnet ein bereits bestehendes Projekt. Im Fenster <i>Open a Safety Projectfile</i> kann das gewünschte Projekt ausgewählt und geöffnet werden.	
	Save Project	Das derzeit geöffnete Projekt kann unter anderem Namen und anderem Pfad gespeichert werden. Das Projekt ist anschließend mit dem neuen Namen/Pfad geöffnet.	
	Close Project	Schließt das derzeit geöffnete Projekt.	
	Print	Druckt das gesamte Projekt mit allen Netzwerken.	
	About XN Safety Designer	Zeigt das Impressum mit Datum und Versionsstand an.	
	Undo	Rückgängig, macht die letzte Aktion rückgängig.	Build-Werkzeugleiste
	Redo	Wiederherstellen, stellt die letzte <i>Undo</i> Aktion wieder her.	
	Generate Project	Generiert die Download-Dateien des Projekts.	
	Monitored Settings	Öffnet einen Dialog und berechnet die voraussichtlichen Zykluszeiten und Übertragungszeiten.	
	Check Resources	Öffnet einen Dialog mit voraussichtlichem Speicherverbrauch, Zykluszeiten und Reaktionszeiten.	Debug-Werkzeugleiste
	Online	Stellt eine Online-Verbindung zwischen dem PC (Projekt) und der Steuerung her und generiert die Download-Daten.	
	Reset forced values	Setzt die geфорcten Werte auf der Safety-CPU zurück.	
	Reset/Start PLC	Reset und Start der PLC vor dem Start der Safety-Applikation.	
	Online Settings...	Öffnet das Fenster <i>XN Online configurations</i> .	
	Current Online Configuration	Online Verbindung zur Safety-CPU.	
	Online State	Öffnet und schließt die Online Statusanzeige.	

4. Grafik Editor

4.3 Werkzeug-Symboleiste

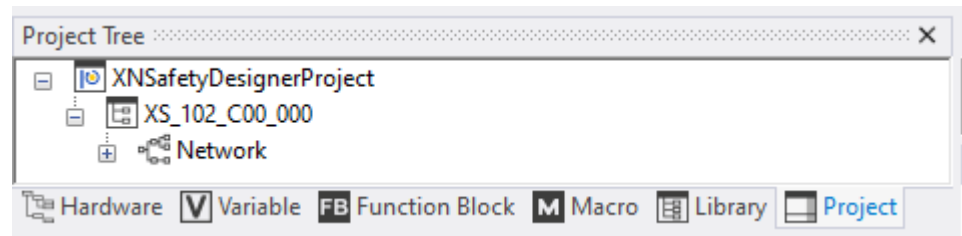
Übersicht			
	Zoom	Zoom-Werkzeug zum Vergrößern bzw. Verkleinern von Objekten im Netzwerk.	Edit-Werk- zeugleiste
	Zoom All	Zoomt alle Objekte im Netzwerk so, dass alle sichtbar sind.	
	Zoom Selection	Zoomt einen gewünschten Bereich im Netzwerk.	
	Pan	Handwerkszeug zum Navigieren im Netzwerk.	
	Autosize	Spaltenbreite eines Netzwerks automatisch berechnen.	
	Default Size	Spaltenbreite des Netzwerks auf den Defaultwert setzen.	

4.4 Tree-Ansichten

Die Trees (Baumstrukturen) sind standardmäßig auf der linken Seite des Bearbeitungstools platziert und umfassen 6 Bereiche:

- → Abschnitt "Project Tree", Seite 41
- → Abschnitt "Hardware Tree ", Seite 43
- → Abschnitt "Function Block Tree", Seite 46
- → Abschnitt "Variable Tree", Seite 48
- → Abschnitt "Library Tree", Seite 51
- → Abschnitt "Function Block Macro Tree", Seite 52

Jeder dieser Bereiche kann durch Registerkarten angewählt werden.



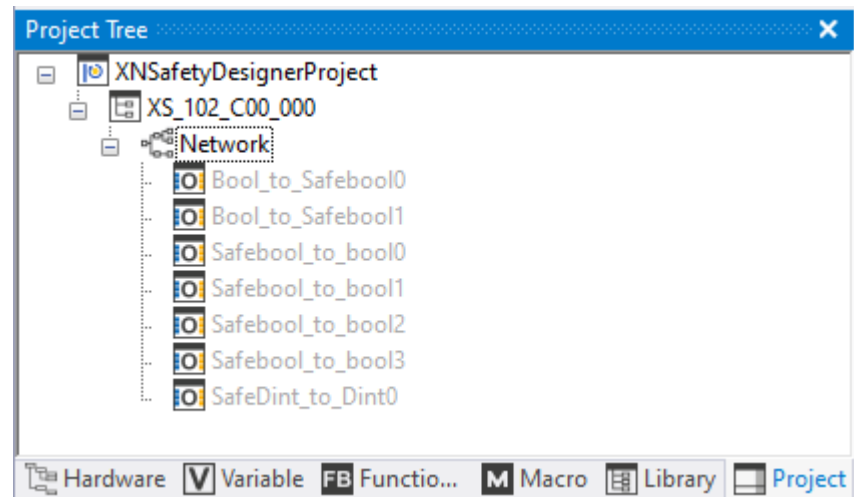
Bis auf den **Function Block Tree**, können alle Trees angepasst werden. Bei der Selektion des Root-Elements steht die Eigenschaft „Temporary Operation Time“ zur Verfügung. Diese Zeit (Einheit Minuten) gibt an, wie lange der temporäre Operation Mode läuft bis die Safety-CPU abschaltet, wenn nicht vorher die Verifizierung des Programms mit „Set Verified“ durch den Anwender abgeschlossen wurde. Der Wertebereich für diese Einstellung liegt zwischen 5 Minuten und 480 Minuten (8 Stunden) (→ Abschnitt " Einstellparameter im Projekt", Seite 81).

4. Grafik Editor

4.4 Tree-Ansichten

4.4.1 Project Tree

In dieser Baumstruktur ist der allgemeine Projektaufbau ersichtlich. Die Safety-CPU, die dazugehörigen Netzwerken und die platzierten Funktionsblöcke werden aufgelistet.



Das Root-Element zeigt den Projektnamen an. Über das Kontextmenü kann ein komplettes Projekt mit der Safety-CPU und den Netzwerken ausgedruckt werden.

[Print Project with all Safe CPU Settings](#)

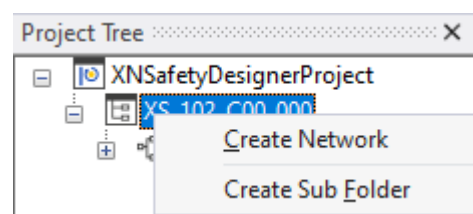
Über das Kontextmenü kann auch eine Revisionshistorie in eine XML-Datei geschrieben werden.

[Export Revision Documentation](#)

Die Safety-CPU wird als Unterelement des Root-Elements angezeigt.

Der Namen der Safety-CPU kann hier nicht umbenannt werden; ein neuer Name kann nur im **Hardware Tree** beim gleichnamigen Element gesetzt werden. Der Name wird dann automatisch in den **Project Tree** übernommen.

Im Kontextmenü der Safety-CPU kann mit **Create Network** ein Netzwerk erstellt werden.



Ebenso kann das Netzwerk mittels der [Einfg]-Taste erstellt werden.

Das Netzwerk wird als Unterelement der Safety-CPU eingefügt. Durch einen Doppelklick auf den Netzwerknamen im Tree wird der Netzwerkeditor geöffnet.

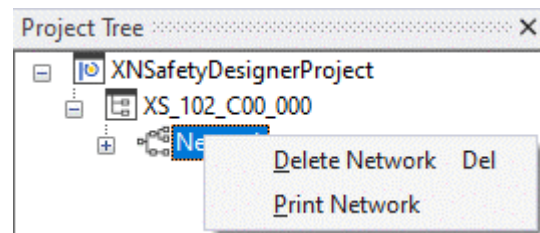
4. Grafik Editor

4.4 Tree-Ansichten

Die platzierten Funktionsblöcke werden als Unterelemente der Netzwerke angezeigt. Das Einfügen dieser Elemente erfolgt automatisch durch Platzieren im Netzwerkeditor.

Der Name des Netzwerkes kann im Baum geändert werden, muss aber innerhalb der Safety-CPU eindeutig sein.

Das Netzwerk kann mittels Kontextmenü gelöscht oder ausgedruckt werden.



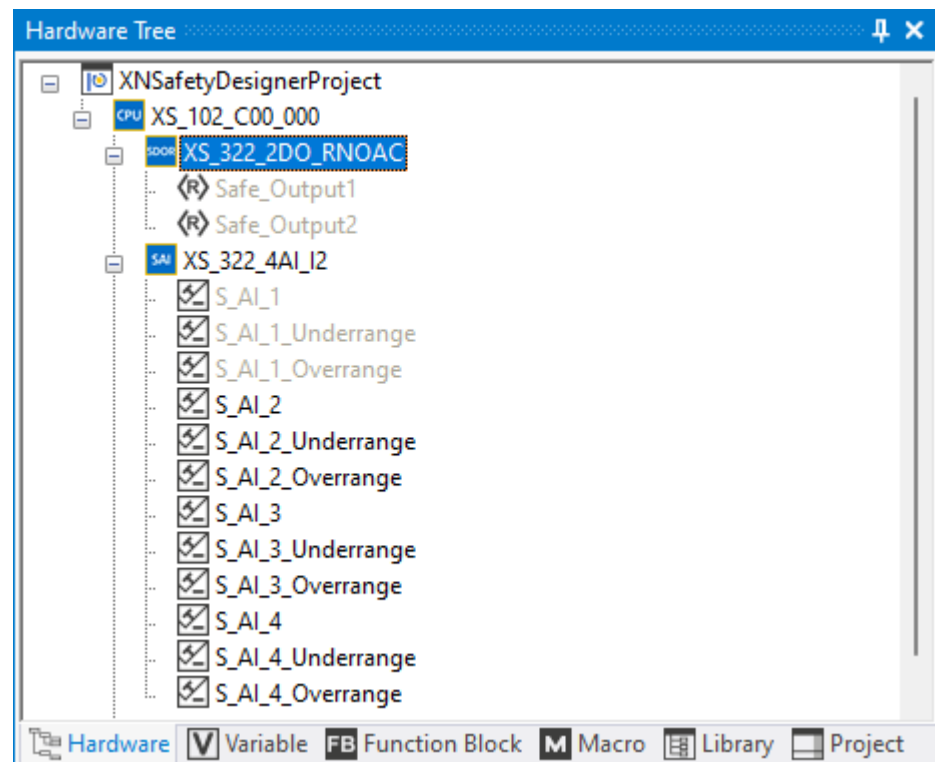
Durch eine Selektion der Elemente (Projekt, Safety-CPU, Netzwerk, Funktionsblock-Instanz) können deren Eigenschaften im Property-Browser angezeigt werden.

4. Grafik Editor

4.4 Tree-Ansichten

4.4.2 Hardware Tree

In dieser Baumstruktur ist die Topologie der Safety-Module mit den entsprechenden Ein- und Ausgängen ersichtlich. Die Topologie kann mittels Upload von der Steuerung ermittelt werden. Falls dies nicht möglich ist, kann mittels Kontextmenü eine Topologie manuell angelegt werden (siehe → Abschnitt "Erstellen der Hardware-Konfiguration", Seite 28).



Die Unterelemente spiegeln die Topologie der Safety-Module wieder. Alle Module haben als Unterelemente ihre physikalischen Ein- und Ausgänge aufgelistet.

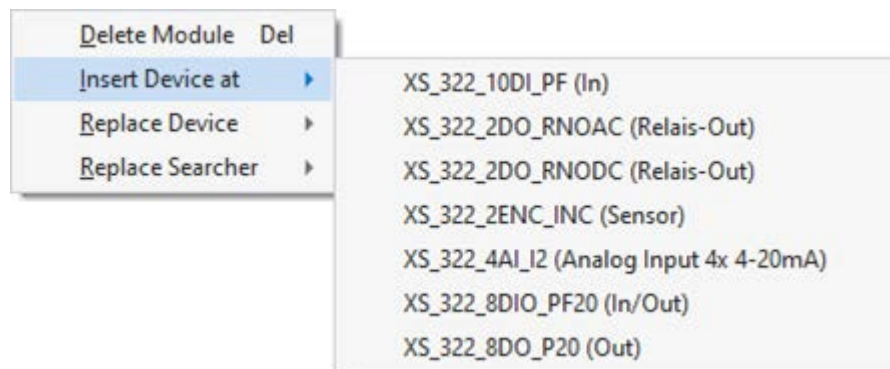
Die Elemente (Module sowie Eingänge/Ausgänge) können mit der Taste F2 bzw. mit einem verzögerten Klick umbenannt werden. Die Namen müssen innerhalb des Projekts eindeutig sein.

Verwendete Eingänge und Ausgänge werden im Tree durch eine hellere Farbe (hellgrau) dargestellt.

Hardwaremodule können mittels Drag & Drop im Tree verschoben werden. Die Module können nur innerhalb des für sie vorgesehenen Bereichs verschoben werden, dabei darf der neue Platz noch nicht belegt sein.

Die Eigenschaften von Eingängen bzw. Ausgängen eines Moduls und deren Verwendungen in Netzwerken können mittels Drag & Drop im Tree auf ein anderes Modul übertragen werden. Es muss dafür der Eingang/Ausgang (von welchem die Daten übernommen werden sollen) auf den neuen Eingang bzw. Ausgang gezogen werden. Die Eingänge und Ausgänge können nur getrennt verschoben werden. Es wird über ein Cursor-Symbol angezeigt, ob ein Verschieben erlaubt ist (dunkles Rechteck für erlaubt, rotes Kreuz für verboten). Die Eigenschaften (z.B. Querschuss) der verschobenen Elemente werden auf die neuen Eingänge und Ausgänge übertragen. Die Namen der verschobenen Elemente werden übernommen (Ausnahme: Verschieben im eigenen Modul). Die Verwendungen der verschobenen Eingänge/Ausgänge in den Netzwerken werden ersetzt. Die bestehenden Verbindungen in den Netzwerken bleiben dabei erhalten. Und-Verknüpfungen in den Eingangsspalten werden nur umgesetzt, wenn beide Eingänge bzw. Ausgänge verschoben werden.

Des Weiteren besteht die Möglichkeit sichere Hardware-Module eines bestimmten Typs durch einen anderen Typ zu ersetzen. Im Kontextmenü werden alle Module angezeigt, die für das Ersetzen zur Verfügung stehen.

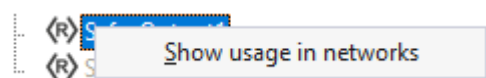


Der Anwender muss das Ersetzen bestätigen, da Ein- und Ausgänge, die auf dem neuen Modul nicht mehr vorhanden sind, gelöscht werden.

Die Eingänge/Ausgänge der Safety-Module können mittels Drag & Drop im → Abschnitt " Editor", Seite 58 platziert werden.

Durch Selektion eines Elements können dessen Eigenschaften im **Properties Pane** angezeigt und editiert werden.

Über das Kontextmenü der Ein- und Ausgänge besteht die Möglichkeit mit dem Befehl **Show usage in networks** zu jenen Stellen im Network (Netzwerk) zu springen, wo das entsprechende Element platziert ist.



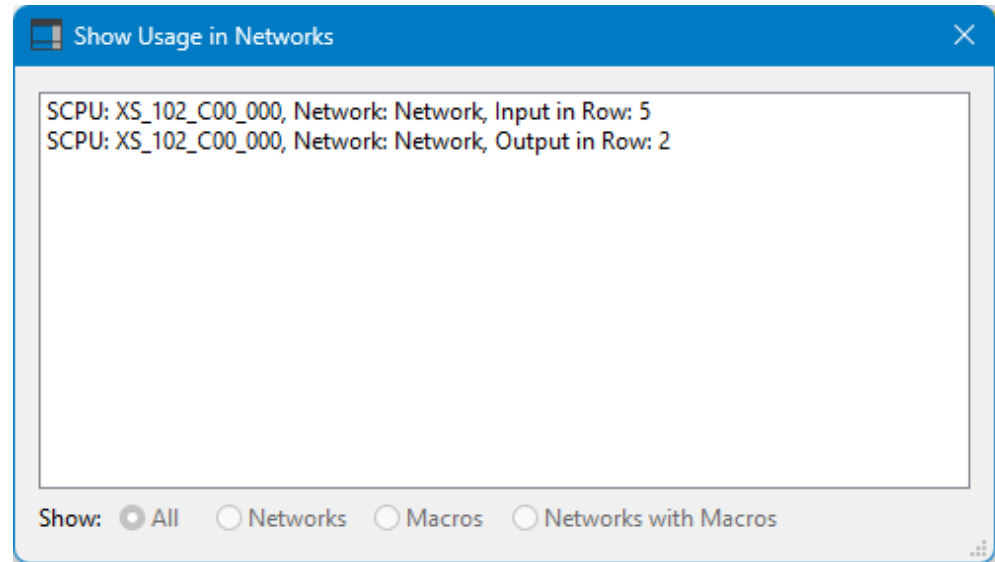
4. Grafik Editor

4.4 Tree-Ansichten

Das Kontextmenü wird nur angezeigt, wenn der entsprechende Ein- oder Ausgang in einem Netzwerk verwendet wird.

Bei einer einmaligen Verwendung des Kanals wird sofort an die jeweilige Stelle im Netzwerk gesprungen.

Wird der Kanal mehrfach verwendet kann im Dialog *Show Usage in Networks* die gewünschte Stelle durch Doppelklick auf die Zeile im Dialog ausgewählt werden.

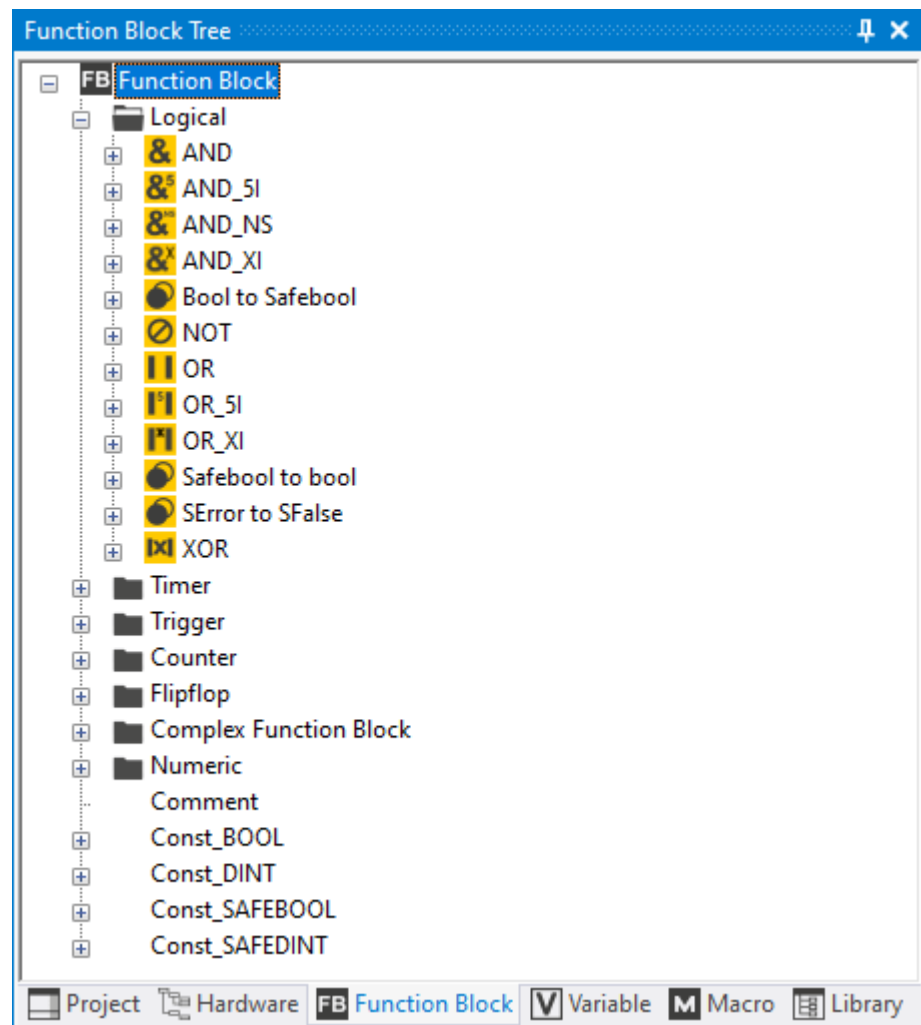


4.4.3 Function Block Tree

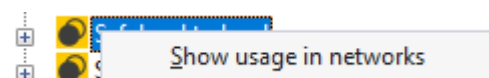
In dieser Baumstruktur sind die vordefinierten Funktionsblöcke abgelegt.

Die Elemente können mittels Drag & Drop als Instanz im Netzwerk→ Abschnitt " Editor", Seite 58 platziert werden.

Die verschiedenen Funktionsblöcke sind thematisch in Ordner aufgeteilt (→ Abschnitt "Funktionsblöcke", Seite 164).



Über das Kontextmenü bei den Funktionsblöcken besteht die Möglichkeit zu der Stelle im Netzwerk zu springen, wo die Funktion verwendet wird.

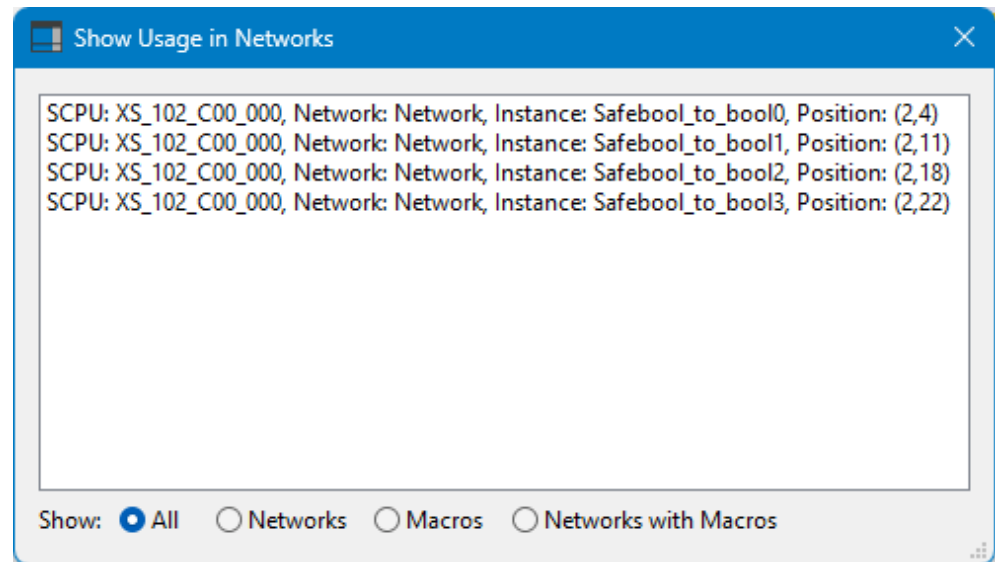


4. Grafik Editor

4.4 Tree-Ansichten

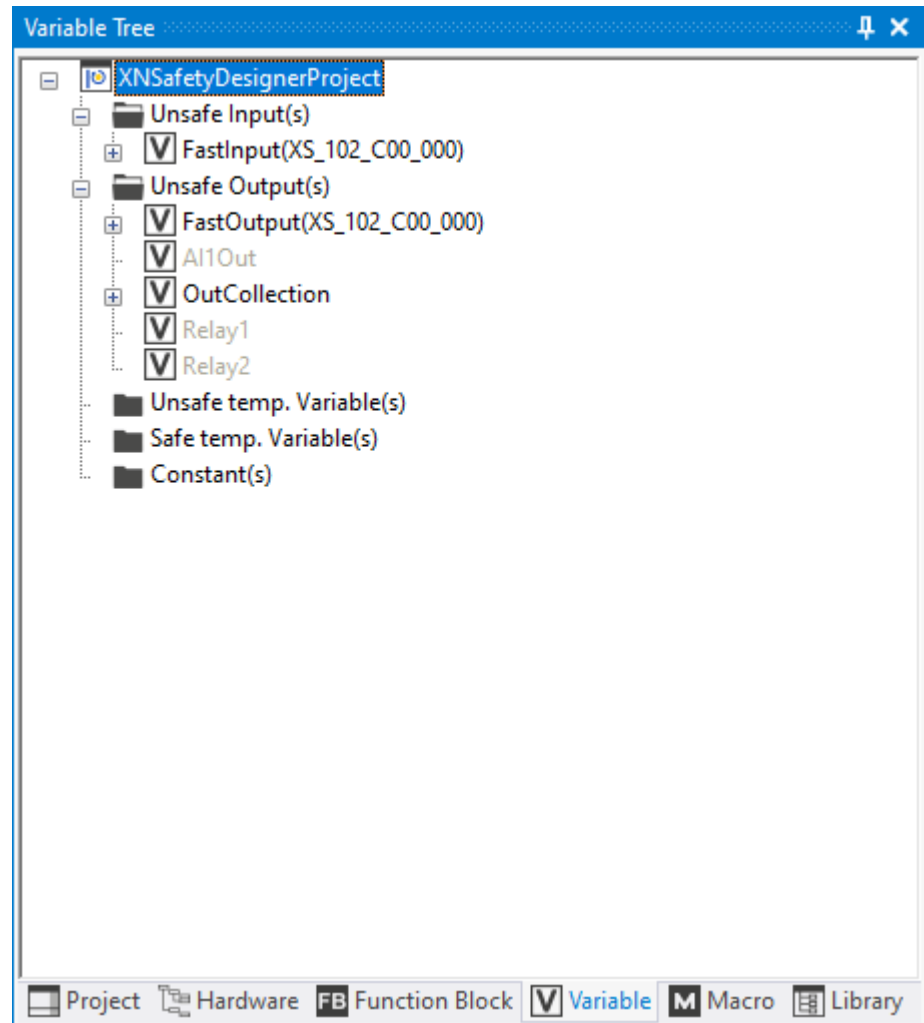
Das Kontextmenü wird nur angezeigt, wenn der Funktionsblock verwendet wird.
Bei einer einmaligen Verwendung des Funktionsblocks wird sofort an die jeweilige Stelle im Netzwerk gesprungen.

Wird der Funktionsblock mehrfach verwendet, kann im Dialog *Show Usage in Networks* die gewünschte Stelle durch Doppelklick auf die Zeile im Dialog ausgewählt werden.

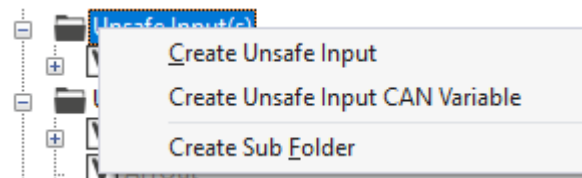


4.4.4 Variable Tree

In dieser Baumstruktur werden die unsicheren Variablen, die temporären Variablen sowie Konstanten verwaltet.



Als erste Unterelemente sind die Ordner für unsichere Ein- und Ausgänge angelegt. Diese Variablen dienen zum Datenaustausch von und zur Standard-SPS. Die Eingänge/Ausgänge können per Kontextmenü oder Taste **[Einfg]** angelegt werden.



Der im Variable Tree vergebene Name ist der Symbolname der Variable. Dieser dient ausschließlich zur leichteren Erkennung im Network, Tree und Watch.

Der symbolische Name kann vom Anwender vergeben werden.

Vorgeschlagen wird immer ein `obj.server` (Objekt.Server-Name), welcher durch den

4. Grafik Editor

4.4 Tree-Ansichten

„.“ getrennt wird. Der Name wird bei der ersten Eingabe interpretiert und falls möglich als Objekt- und Server-Namen übernommen. Wird beim Erstellen kein „.“ im Namen vergeben, muss der Objekt- und Server-Name manuell eingetragen werden.



Wird der Symbolname einer Variable verändert, ändern sich der Objekt- und Server-Name nicht. Es erfolgt keine Prüfung der Objekt- und Server-Namen auf Gültigkeit.

Eine unsichere Variable kann zwei verschiedene Arten von Variablen unterstützen. Die Standard-Variable repräsentiert ein Objekt mit dazugehörigem Server im unsicheren Applikationsprojekt. Die I/O-Element-Variable repräsentiert ein I/O-Element im unsicheren Applikationsprojekt. Die Einstellung der Variablen-Art erfolgt unter **Properties** zur Variablen.

Properties	
Comment	
Object	obj
Server	server1
BDINT Variable	false
Type of variable	Variable
	Variable
	IO-Element

Ein bestimmter Datentyp wird hier nicht angegeben, alle Server entsprechen einem 4Byte-Wert. Je nach Verwendung bei einem bestimmten Funktionsblock wird der Wert dann als BOOL oder DINT interpretiert. Falls keine unsicheren Variablen angelegt werden können, wird ein entsprechender Tooltip angezeigt:

Tooltip	Beschreibung
Variables are not allowed in a stand alone Safety CPU	Der Stand-Alone-Betrieb einer Safety-CPU unterstützt keine unsicheren Variablen.

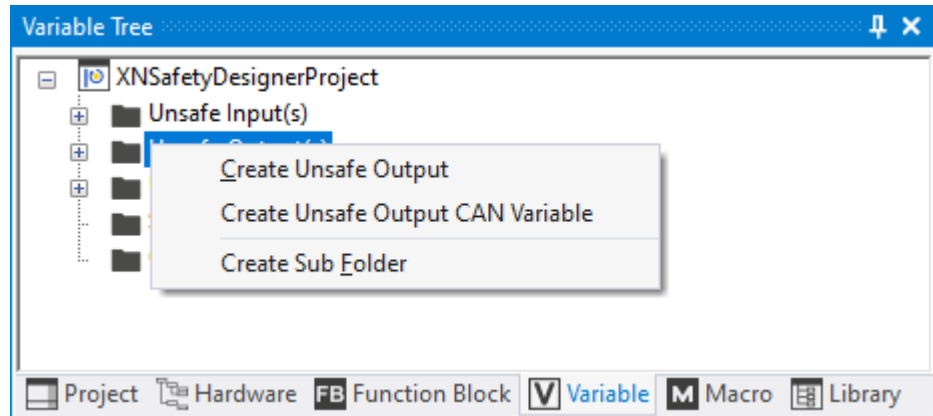
4.4.4.1 Bit-Variablen

Die erstellten Variablen können über die **Properties** in eine Bit-Variable umgestellt werden. Bei den Bit-Variablen werden 32 einzelne Bits angelegt, welche im Netzwerk platziert werden können. Die Anzeige der Bit-Nummern erfolgt über ein Icon mit der entsprechenden Zahl. Zusätzlich kann für jede Safety-CPU über das Kontextmenü eine schnelle 32-Bit Eingangs- und Ausgangsvariable angelegt werden.

Create Fast Unsafe Variable for 'XS_102_C00_000'

4. Grafik Editor

4.4 Tree-Ansichten



Die nächsten Unterelemente sind die Ordner für temporäre sichere und nicht sichere Variablen. Diese Variablen sind Marker, d.h. sie werden in einem bestimmten Abschnitt im Netzwerk gesetzt und können später im Programm wieder verwendet werden. Dabei wird überprüft, ob in der Reihenfolge der Abarbeitung die temporäre Variable als erstes gesetzt und anschließend verwendet wird.

Die Variablen können per Kontextmenü angelegt werden.

Die Namen müssen innerhalb des Projekts eindeutig sein. Die Elemente können als Eingang und Ausgang verwendet werden (konfigurierbar über die [Properties](#)).



Die temporären Variablen sind nur Platzhalter in den Netzwerken. Beim Erstellen der Applikation werden die Platzhalter entfernt und ein direkter Befehl zum Kopieren der Werte zwischen zwei Funktionsblöcken eingefügt. Eine temporäre Variable kann nur innerhalb einer Safety-CPU verwendet werden.

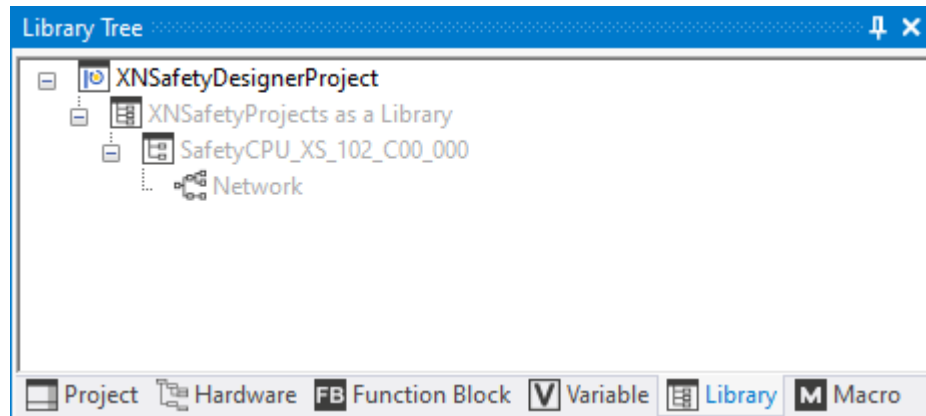
Die Elemente können mittels Drag & Drop im Netzwerk → Abschnitt "Editor", Seite 58 platziert werden.

4. Grafik Editor

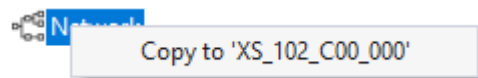
4.4 Tree-Ansichten

4.4.5 Library Tree

In dieser Baumstruktur können andere Safety Designer-Projekte als Library geladen werden. Es werden die einzelnen Safety-CPU's und deren Netzwerke angezeigt.



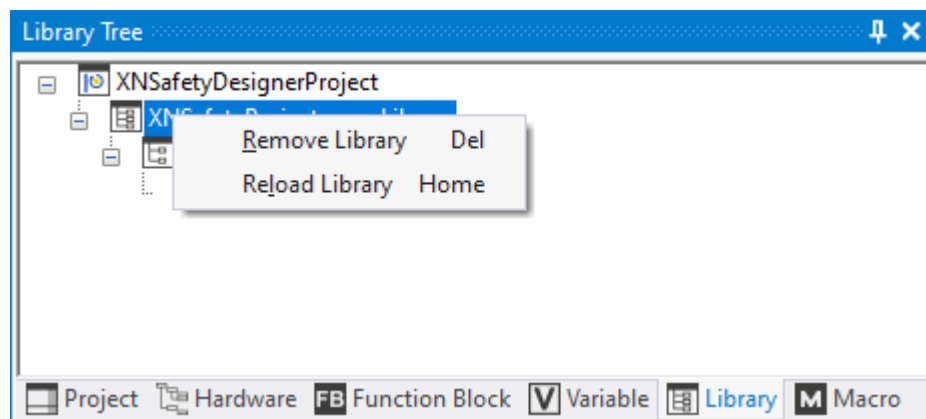
Die Netzwerke können über das Kontextmenü direkt in das Safety Designer-Projekt übernommen werden.



Bei der Übernahme wird überprüft, ob alle Daten korrekt übernommen werden können. Bei festgestellten Konflikten werden die entsprechenden Elemente im Netzwerk durch Platzhalter ersetzt.

Durch einen Doppelklick auf den Netzwerknamen im Tree wird der Netzwerkeditor geöffnet (nur lesend). Elemente dieses Netzwerks können kopiert und in ein anderes Netzwerk eingefügt werden, siehe → Abschnitt "Kopieren, Ausschneiden und Einfügen von Elementen", Seite 74.

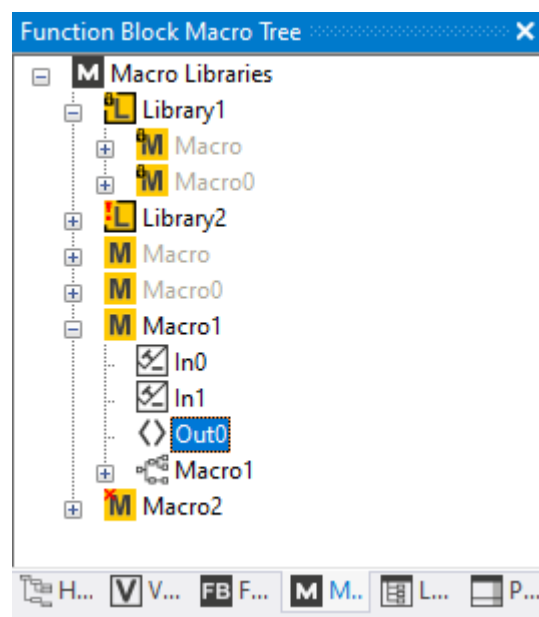
Wird ein Projekt verändert, welches in einem anderen Projekt als Library geladen ist, kann über den Kontext Menü Eintrag **Reload Library** die veränderte Library neu geladen werden.



4.4.6 Function Block Macro Tree

In dieser Baumstruktur können aus Funktionsblöcken Makros erstellt und zu Libraries zusammengefasst werden. Unter Makros wird die Kombination einzelner Funktionsblöcke verstanden, die wie einzelne Funktionsblöcke in den Netzwerken verwendet werden können. Die Makros können in Libraries verwaltet werden, die unabhängig von den einzelnen Projekten sind.

Die Makros können mittels Drag & Drop als Instanz im Netzwerk → Abschnitt "Editor", Seite 58 platziert werden.



Im Tree sind zuerst die Libraries platziert, welche mit einem eingerahmten „L“ als Symbol dargestellt werden. Danach folgen die im aktuellen Projekt verwendeten Makros, die Makros werden mit einem „M“ symbolisiert. Bei der Verwendung der Makros in einem Netzwerk wird der Namen im Tree grau dargestellt. In diesem Fall kann man, wie bei einem Funktionsblock, über das Kontextmenü zu der Stelle im Netzwerk springen (siehe → Abschnitt "Function Block Tree", Seite 46).

Mittels Drag & Drop können Makros auf Libraries gezogen werden und können somit in die Library kopiert werden. Zeigt ein Symbol in der Linken oberen Ecke ein kleines Schloss, so ist das Element geschützt, man kann es verwenden, aber nicht verändern.

Klappt man ein Makro auf, werden die Ein- und Ausgänge wie bei einem Funktionsblock angezeigt. Darunter ist das Symbol des Makro-Netzwerkes. Es trägt den Namen des Makros und kann durch Doppelklick geöffnet werden. Wird das Macro verwendet, so erscheint wie bei den Funktionsblöcken ein Ordner Instance(s), in dem die Vorkommen in den Netzwerken aufgelistet werden. Mit Doppelklick oder via Kon-

4. Grafik Editor

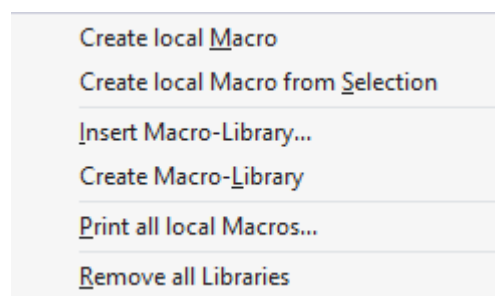
4.4 Tree-Ansichten

textmenü kann von diesen Einträgen zu den Instanzen in den Netzwerken gelangt werden.

Eine Library mit einem roten Rufzeichen in der linken oberen Ecke wurde verändert, aber noch nicht (wieder) gespeichert.

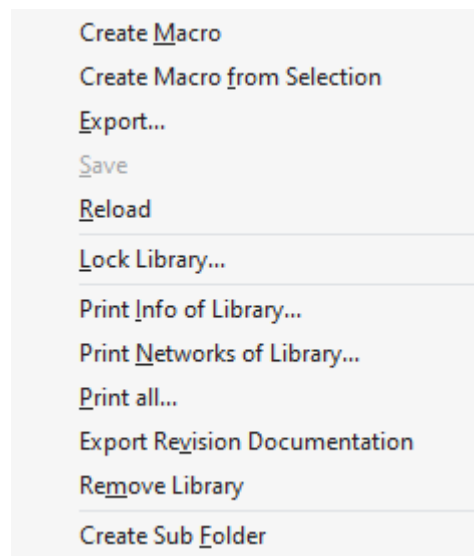
Ein Makro mit einem roten „X“ in der linken oberen Ecke ist ungültig, d.h. es hat nicht mindestens einen Ein- und Ausgang und kann somit nicht verwendet werden.

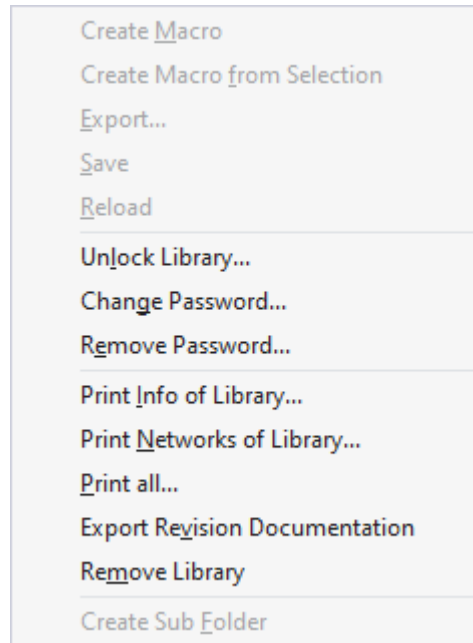
Das Kontextmenü des Basiselements Macro Libraries bietet Funktionen zum Erstellen von Makros und Libraries. Mit InsertMacro-Library... kann man eine Library von einer Datei laden.



Mit Print all local Macros werden die Netzwerke aller Macros die zum Projekt gehören ausgedruckt. Schließlich entfernt Remove all Libraries alle Bibliotheken von der Ansicht. Dabei wird bei jeder geänderten Library gefragt, ob sie gespeichert werden soll.

Das Kontextmenü der Libraries bietet neben den Funktionen, um Makros zu erstellen, auch die Möglichkeit mit „Export“, die Bibliothek in einer neuen Datei abzuspeichern. Wurde sie einmal exportiert können mit „Save“ Änderungen gespeichert werden.

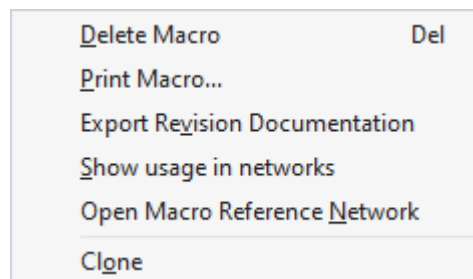




Mit „Set Password“ kann die Bibliothek mit einem Passwort gegen weitere Veränderungen geschützt werden. Im Kontextmenü werden zusätzliche Punkte angezeigt, mit „Unlock Library“ lässt sich die Bibliothek vorübergehend für Modifikationen entsperren. Die Library ist entsperrt, bis mit „Relock Library“ wieder gesperrt oder neu geladen wird. Mit „Change Password“ und „Remove Password“ stehen weitere Funktionen für die Password-Verwaltung zur Verfügung.

„Print Info of Library“ druckt Informationen, Kommentare, Versions-Dokumentationen etc. der Bibliothek und aller enthaltenen Makros. Die Netzwerke aller Makros kann man mit „Print Networks of Library“ ausdrucken lassen. „Print all...“ druckt alle Informationen mit allen Makro-Netzwerken. „Export Revision Documentation“ schreibt die Versions-Dokumentation (falls vorhanden) in eine XML-Datei. Schließlich kann man mit „Remove Library“ die Bibliothek aus der Ansicht entfernen.

Im Kontextmenü der Makros kann man mit „Print Macro...“ das Makro-Netzwerk ausdrucken und mit „Export Revision Documentation“ die Versions-Dokumentation in eine XML-Datei schreiben.

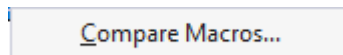


Mit **Clone** kann man im Projekt oder in einer Library eine Kopie des Makros mit anderem Namen erstellen. Das Kontextmenü der Makros in Bibliotheken hat noch den zusätzlichen Punkt **Copy to local library** mit dem das Makro in das Projekt kopiert wird.

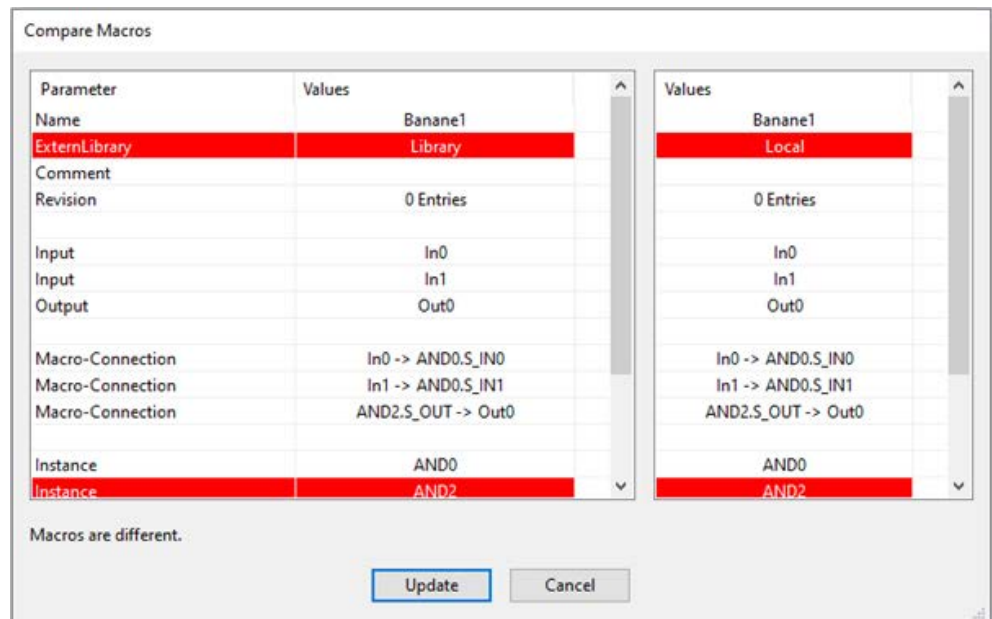
4. Grafik Editor

4.4 Tree-Ansichten

Markiert man ein Makro des Projekts und ein Makro gleichen Namens in einer Bibliothek gemeinsam (durch Halten der Taste **STRG**) so kann man ein spezielles Kontextmenü abrufen.



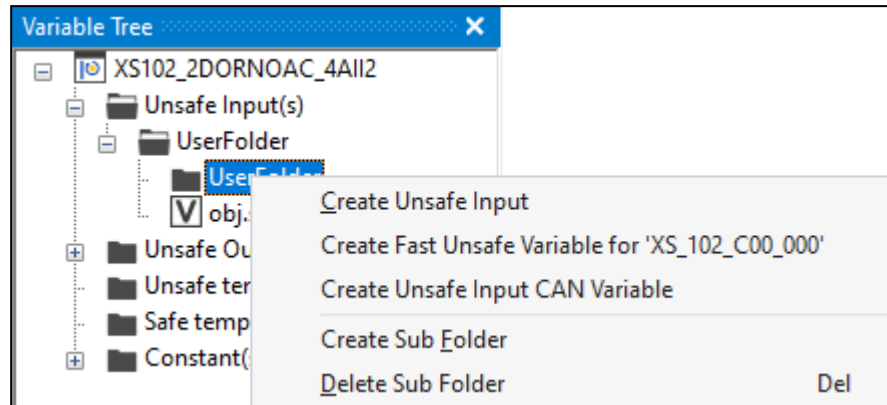
Damit wird ein Dialog aufgerufen, der Unterschiede in roten hinterlegten Zeilen anzeigt.



Wird das Makro im Projekt nicht verwendet, dann kann es mit **Update** durch jenes aus der Bibliothek ersetzen werden.

4.4.7 Unterordner in der Tree-Ansicht

Ordner dienen der Struktur und Organisation von Variablen, Interfaces, Funktionsblock-Macros und Netzwerken. Erstellt werden Ordner über **Create Sub Folder** im Kontextmenü mit einem Standard-Namen **<UserFolder>** bzw. **<LibFolder>**. Ordner können Objekte aus dem Bereich in dem sie sich befinden, z.B. „Unsafe Input(s)“ sowie weitere Ordner als Unterordner enthalten.



→ Die Unterordner eines Ordners müssen unterschiedliche Namen haben.

→ Beim Löschen eines Ordners werden alle enthaltenen Elemente und Unterordner mit ihren Elementen aus dem Projekt entfernt.
Ist ein Element in Verwendung ist ein Löschen nicht möglich.

Ordner übernehmen das Kontextmenü des Bereiches, in dem sie platziert sind, z.B. ein Ordner im Ordner „Unsafe Input(s)“ hat alle Menü-Punkte für unsichere Eingänge. Mit der Drag&Drop Funktion können Ordner und Elemente eines Ordners verschoben werden.

→ Enthält der Zielordner bereits einen Unterordner gleichen Namens, ist ein Verschieben nicht möglich.

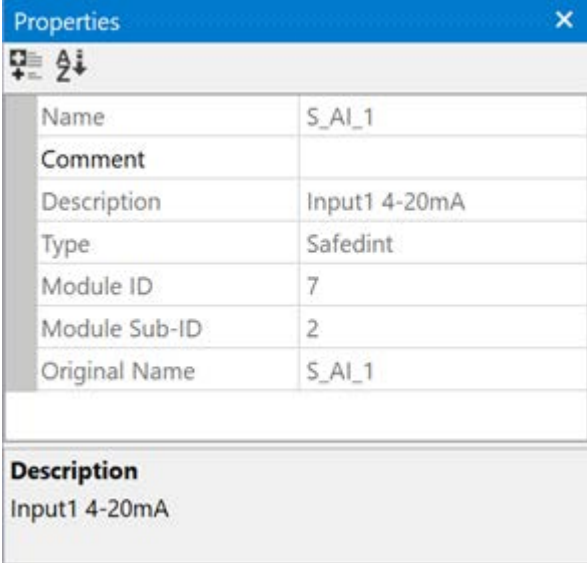
→ Elemente eines Bereichs, z.B. unsichere Eingänge, in unterschiedlichen Ordnern, müssen trotzdem eindeutige (unterschiedliche) Namen haben.

4. Grafik Editor

4.5 Property Browser

4.5 Property Browser

Im Property Browser werden je nach Selektion im Project Tree oder Netzwerkeditor die entsprechenden Eigenschaften des Elements angezeigt. Die Elemente sind entweder schreibgeschützt oder bearbeitbar. Der Property Browser ist die zentrale Stelle für das Setzen von Parametern.



Properties	
Name	S_AI_1
Comment	
Description	Input1 4-20mA
Type	Safedint
Module ID	7
Module Sub-ID	2
Original Name	S_AI_1

Description
Input1 4-20mA

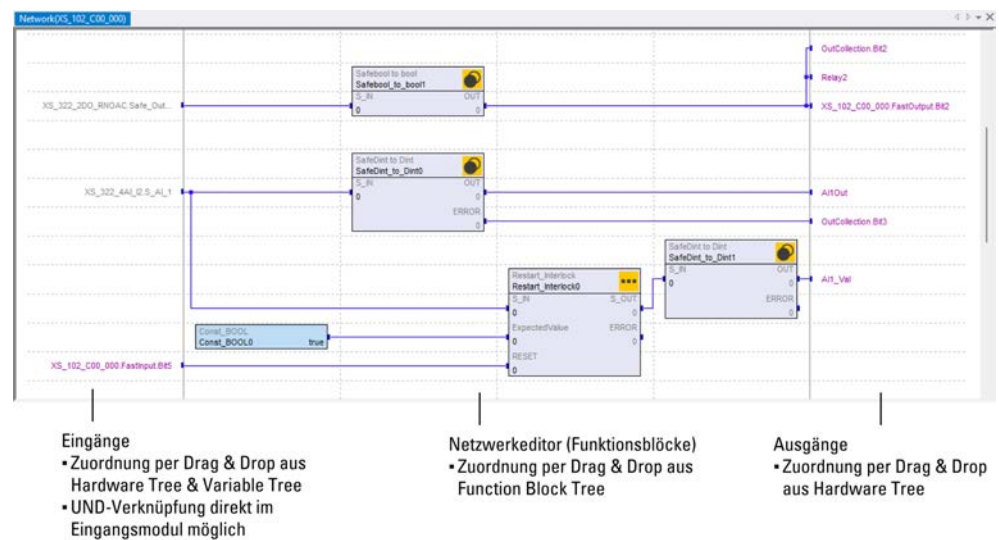
Im Online-Betrieb sind alle Einstellungen ausgegraut (disabled).

4.6 Editor

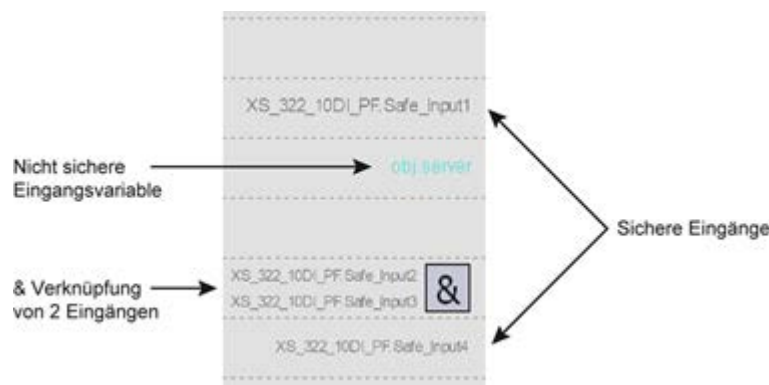
Im Netzwerk-Editor wird mittels Drag & Drop der Ablaufcode der Safety-CPU graphisch erstellt. Es werden dazu Eingänge und Ausgänge von Safety-Modulen mit zertifizierten Funktionsblöcken logisch verknüpft.

Der Netzwerk-Editor ist in 3 Bereiche aufgeteilt:

- Spalte für Eingänge (linker Bereich des Editors)
- Netzwerkeditor (Bereich zwischen Eingangs- und Ausgangsspalte)
- Spalte für Ausgänge (rechter Bereich des Editors)



4.7 Spalte für Eingänge



Es können in den einzelnen Zeilen Eingänge, Ausgänge, nicht sichere Variablen und Bit-Variablen platziert werden.

Die Elemente werden mittels Drag & Drop aus dem Hardware Tree bzw. Variable Tree eingefügt. In einer Zeile kann auch eine UND-Verknüpfung von zwei sicheren Eingängen vorgenommen werden (für z.B. Zweikanaligkeit von Not-Halt). Für die Verknüpfung wird ein Eingang mittels Drag & Drop platziert, anschließend kann ein zweiter Eingang in die entsprechende Zeile gezogen werden. Der zweite Eingang kann entweder aus dem Hardware Tree kommen oder ein Eingang aus der Eingangsspalte sein. Solche Verknüpfungen in der Eingangsspalte werden direkt im jeweiligen Eingangsmodul ausgewertet, so dass nur das Ergebnis der Verknüpfung zur Safety-CPU übertragen wird. Dies bedeutet, dass die UND-verknüpften Eingänge vom gleichen Safe-Modul stammen müssen.

- ➔ Bei der UND-Verknüpfung in der Eingangsspalte werden die beiden Eingänge defaultmäßig über den AND-Funktionsblock miteinander verbunden
- ➔ Sollten Sie eine Verknüpfung mit einer Diskrepanzzeit-Überwachung benötigen, darf diese UND-Verknüpfung nicht verwendet werden. In diesem Fall müssen Sie eine eigene Verknüpfung über den Äquivalenz-Funktionsblock erstellen. Die UND-Verknüpfung der Eingangsspalte kann mit Diskrepanzzeit-Überwachung im Property Browser eingestellt werden. Es kann dazu zwischen einer Verknüpfung mit einem AND- und einem Äquivalenz-Funktionsblock unterschieden werden.

Alle Eingänge, Ausgänge und Variablen können mehrfach in der Eingangsspalte platziert werden.

Das Platzieren eines Ausgangs in der Eingangsspalte bedeutet, dass der aktuelle Ausgangszustand als Eingangswert verwendet wird.

Durch das Drücken der Taste **[SHIFT]** können mittels Drag & Drop aus dem **Hardware Tree** bzw. **Variable Tree**, bestehende Eingänge im Netzwerk ersetzt werden. Es können nur gleichwertige Elemente ersetzt werden (z.B. unsichere Variablen durch unsichere Variablen). Die Verbindungen zum Eingang bleiben bestehen.

Die Eigenschaften der Elemente können unter **Properties** verändert werden.

Äquivalenz-Verknüpfung in der Eingangsspalte

Eine UND-Verknüpfung kann mit einem Äquivalenz-Funktionsblock verwendet werden (Verknüpfung mit Diskrepanzzeit-Überwachung). Die Art der Und-Verknüpfung kann unter **Properties** der Und-Verknüpfung eingestellt werden. Beim **Property Function block** kann über eine Combobox zwischen einem AND- und einem Äquivalenz-Funktionsblock ausgewählt werden (Default-Einstellung AND). Bei der Auswahl des Äquivalenzblocks muss zusätzlich die Diskrepanzzeit eingestellt werden. Die Einstellung erfolgt über das Property **Discrepancy Time**; die Einstellung ist nur beim Äquivalenz-Block sichtbar. Der Defaultwert ist 0; der Grenzbereich liegt zwischen 0 und 86400000 ms. Die Diskrepanzzeit ist die maximale Verzögerungszeit welche beim Setzen der beiden Eingänge auftreten darf; beim Überschreiten dieser Zeit geht der Äquivalenz-Funktionsblock in den Fehlerzustand.

Properties	
 	
Position	(-1,26)
Comment	
Module	XS_322_8DIO_PF20
Name	Safe_Input1
Cross Circuit Detectio	0
Filter Time	0
Module 2	XS_322_8DIO_PF20
Name 2	Safe_Input2
Cross Circuit Detectio	0
Filter Time 2	0
IsSafe	true
Function block	SF_Equivalent
Discrepancy Time	5

Bei der Auswahl des Äquivalenz-Funktionsblocks wird für die Abarbeitung der SF-Äquivalenz-Funktionsblock verwendet. Die einzelnen Eingänge und Ausgänge des Funktionsblocks können im Online-Betrieb im Watch-Fenster angezeigt werden. Die Fehlerzustände werden am DiagCode-Ausgang angezeigt und können in der

4. Grafik Editor

4.7 Spalte für Eingänge

Beschreibung des Funktionsblocks nachgelesen werden. Fehler können durch ein Rücksetzen der sicheren Eingänge quittiert werden.

Die Darstellung der beiden Verknüpfungen erfolgt im Netzwerk mit unterschiedlichen Symbolen:

AND:

XS_322_10DI_PF.Safe_I ...
XS_322_10DI_PF.Safe_I ... 

Aquivalenz:

XS_322_10DI_PF.Safe_I ...
XS_322_10DI_PF.Safe_I ... 

4.8 Spalte für Ausgänge



Die Elemente werden mittels Drag & Drop aus dem **Hardware Tree** bzw. **Variable Tree** eingefügt.

Ein spezifischer **SafetyOutput** kann nur einmal in einem Safety-Projekt in einer Ausgangsspalte platziert werden. Die Verwendung von nicht sicheren Variablen in der Ausgangsspalte ist erlaubt; diese können ebenfalls nur einmal platziert werden. Die Verwendung von schreibenden Interfaces und Bit-Variablen ist erlaubt; diese können ebenfalls nur einmal platziert werden.

Es kann bei den sicheren Ausgängen eine Und-Verknüpfung mit einem unsicheren Freigabesignal eingestellt werden. Unter dem Freigabesignal wird eine unsichere Variable verstanden, welche nicht im Safety-Projekt vorhanden ist.

DO.Safe_Output2



Die Darstellung von Relais-Ausgängen erfolgt in zwei Zeilen, dies soll verdeutlichen, dass die Ausgänge in der Hardware doppelt ausgeführt sind.

XS_322_2DO_RNOAC.Safe_Output1_1
XS_322_2DO_RNOAC.Safe_Output1_2

Durch das Drücken der Taste **[SHIFT]** können mittels Drag & Drop aus dem **Hardware Tree** bzw. **Variable Tree**, bestehende Ausgänge im Netzwerk ersetzt werden.

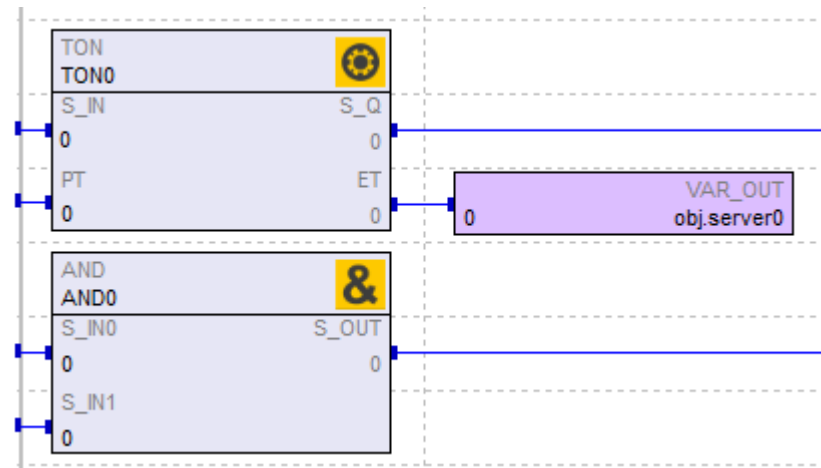
Es können nur gleichwertige Elemente ersetzt werden (z.B. unsichere Variablen durch unsichere Variablen).

Die Verbindung zum Ausgang bleibt bestehen.

Die Eigenschaften der Elemente können im **Properties Pane** verändert werden.

4.9 Network Editor

Im Netzwerkeditor werden die verknüpften Elemente eines Safe-Netzwerks angezeigt.

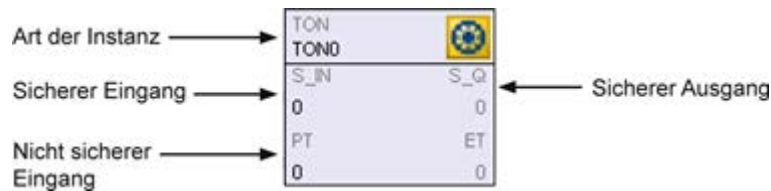


Der Netzwerkeditor hat folgende Bestandteile:

- Hintergrundraster (zur besseren Orientierung), Raster entspricht 1 Zeilen und 1 Spalte
- Instanzen von Funktionsblöcken
- Konstanten
- Variablen
- Temporäre Variablen
- Bit-Variablen
- Verbindungen
- Kommentare

4.9.1 Netzwerkelemente

Folgendes Element im Netzwerkeditor repräsentiert eine Instanz eines Funktionsblocks. Die Elemente werden mittels Drag & Drop aus dem Function Block Tree eingefügt. Sichere Ein- und Ausgänge sind daran erkennbar, dass sie immer mit S beginnen.



Zusätzlich können Blöcke für Konstanten, unsichere Variablen und temporäre Variablen platziert werden. Diese Blöcke besitzen keine Kopfzeile. Blöcke für Konstanten haben nur einen Ausgang (können als Eingangsparameter für Funktionsblöcke verwendet werden).

Blöcke für unsichere Variablen können im Tree entweder als Eingangsvariable oder als Ausgangsvariable angelegt werden. Es können keine sicheren Variablen angelegt werden. Die Elemente werden mittels Drag & Drop aus dem Variablen Tree eingefügt. Nicht sichere Variablen werden aus der unsicheren Applikation der Standard-SPS übernommen. Die Werte der nicht sicheren Eingangsvariablen können vom Anwender in der unsicheren Applikation gesetzt werden. Die nicht sicheren Eingangsvariablen können nur mit nicht konstanten unsicheren Eingängen von Funktionsblöcken verbunden werden; dies ist z.B. der Activate-Eingang der komplexen Funktionsblöcke. Es ist sichergestellt, dass ein sicherer Ausgang nicht alleine durch eine unsichere Variable gesetzt werden kann. In der Firmware werden die verwendeten nicht sicheren Variablen mit 0 vorbelegt.

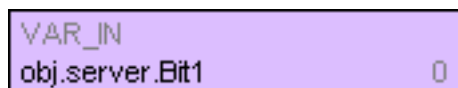


4. Grafik Editor

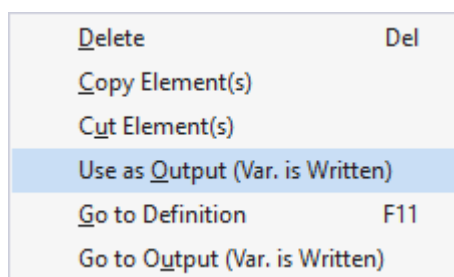
4.9 Network Editor

4.9.1.1 Bit-Variablen

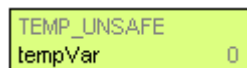
Blöcke für unsichere Bit-Variablen können im Tree entweder als Eingangsvariable oder als Ausgangsvariable angelegt werden. Die Variablen können schnelle 32 schnelle Bit-Variablen einer Safety-CPU oder 32 Bit-Variablen sein. Die Elemente werden mittels Drag & Drop aus dem Variablen Tree eingefügt.



Blöcke für temporäre Variablen können als Ausgang oder Eingang verwendet werden. Die Festlegung erfolgt mittels Kontextmenü



bei der Instanz im Netzwerk oder nach Selektion über das **Properties Pane**. Die Elemente werden mittels Drag & Drop aus dem **Variablen Tree** eingefügt.

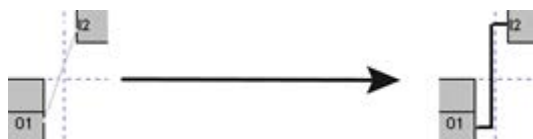


Alle Eingänge werden auf der linken Seite der Instanz angezeigt, alle Ausgänge auf der rechten Seite. Die Höhe des Kopfes, bzw. eines Ein- oder Ausganges entspricht einer Zeile im Editor. Nicht sichere Variablen, sowie Konstanten, temporäre Variablen werden in einer gesonderten Farbe (Konstanten = blau; Unsichere Variablen = violett; Temporäre Variablen = grün) dargestellt, um die Elemente besser unterscheiden zu können.

Durch das Drücken der Taste **[SHIFT]** können mittels Drag & Drop aus dem Variable Tree, bestehende Funktionsblöcke für unsichere Variablen und temp. Variablen im Netzwerk ersetzt werden. Es können nur gleichwertige Elemente ersetzt werden (z.B. unsichere Variablen durch unsichere Variablen). Die Verbindungen zum Element bleiben bestehen.

Verbindungen zwischen den verschiedenen Netzwerkelementen können mit der Maus gezogen werden.

Es muss nur der Start- und der Endpunkt der Verbindung festgelegt werden, der Linienerverlauf wird vom **XN Safety Designer** selbstständig generiert.



- Es können nur Verbindungen gezogen werden wenn die Datentypen (z.B. SAFEBOOL) der Endpunkte übereinstimmen bzw. kompatibel sind.
- Jedes Element der Ausgangsspalte kann nur einmalig verbunden werden.
- Eingänge von Instanzen können nur einmalig verbunden werden.
- Kreuzungen zwischen Verbindungen sind erlaubt. Abzweigungen werden mittels eines Knotenpunkts angezeigt.
- Der Anwender wird durch einen entsprechenden Mauscursor hingewiesen, ob eine Verbindung möglich ist.

Gültige Verbindung:



Ungültige Verbindung:



Das Element Kommentar wird mittels Drag & Drop aus dem Function Block Tree eingefügt. Das Kommentarfeld kann im gesamten Bereich platziert werden, wenn die Position nicht bereits belegt ist. Das Cursorsymbol zeigt an, ob ein Einfügen erlaubt ist (dunkles Rechteck für erlaubt, rotes Kreuz für verboten).

Kommentar

4. Grafik Editor

4.10 Allgemeine Arbeitsweise des Editors

4.10 Allgemeine Arbeitsweise des Editors

Im Folgenden wird die Bedienung des Editors beschrieben.

Nicht gültige Aktionen werden entweder durch ein Cursor-Symbol, farblich oder durch eine Fehlermeldung im Output-Fenster angezeigt.

Beim Drag & Drop und Verschieben von Elementen wird mittels Symbol angezeigt, ob die Einfügen des Elements an dieser Stelle möglich ist.

4.10.1 Füllen der Eingangsspalte

Aus dem Hardware Tree/Variablen Tree können Eingänge/Ausgänge von Safety-Modulen, unsichere Variablen, temporäre Variablen mittels Drag & Drop in die einzelnen Zellen platziert werden. Das Cursor-Symbol zeigt an, ob ein Element an dieser Stelle eingefügt werden kann. Je nach Typ des platzierten Elements (Eingang/Ausgang, unsichere Variable, temporäre Variable) wird die Zelle farblich anders dargestellt (Eingänge/Ausgänge = grau, Unsichere Variablen = violett; Temporäre Variablen = grün).

Bei Selektion einer Zelle werden die Eigenschaften im Property Browser angezeigt. Die gesamte Parametrierung der Elemente erfolgt im Property Browser.

4.10.2 Füllen der Ausgangsspalte

Aus dem Hardware Tree / Variablen Tree können Ausgänge von Safety-Modulen, unsichere Variablen, temporäre Variablen mittels Drag & Drop in die einzelnen Zellen platziert werden. Das Cursor-Symbol zeigt an, ob ein Element an dieser Stelle eingefügt werden kann. Je nach Typ des platzierten Elements (Ausgang, unsichere Variable, temporäre Variable) wird die Zelle farblich anders dargestellt (/Ausgänge = grau, Unsichere Variablen = violett; Temporäre Variablen = grün).

Jeder physikalische Ausgang darf nur einmal in einem Safety Projekt platziert werden.

Bei Selektierung einer Zelle werden die Eigenschaften im Property Browser angezeigt. Die gesamte Parametrierung erfolgt im Property Browser.

4.11 Arbeiten im Netzwerkeditor

4.11.1 Einfügen von Netzwerkelementen

Netzwerkelemente können aus dem Function Block Tree/Variablen Tree mittels Drag & Drop in den Netzwerkeditor gezogen werden. Instanzen von Funktionsblöcken werden aus dem Function Block Tree, Variablen, temporäre Variablen werden aus dem Variablen Tree eingefügt.

Ein gestricheltes Rechteck gibt die Größe des einzufügenden Elements an, das Cursor-Symbol zeigt an, ob das Element an dieser Stelle eingefügt werden kann.

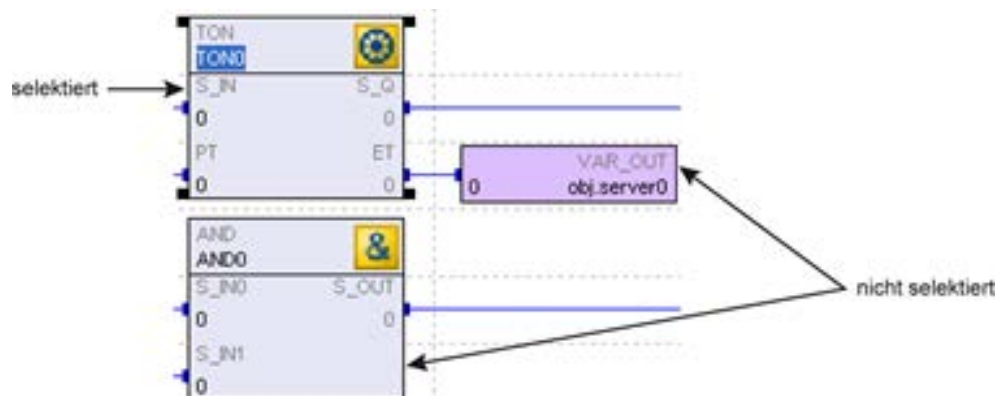
Darstellung

- Roter Cursor (kann an dieser Stelle nicht eingefügt werden)
- Graues rechteckiges Symbol (kann hier eingefügt werden)

Die Platzierung ist nur möglich, wenn an der ausgewählten Stelle genügend Platz vorhanden ist. Die Ein- bzw. Ausgänge eines Funktionsblocks liegen jeweils auf einer eigenen Zeile. Sollte der Platz nicht ausreichen, müssen die bereits eingefügten Objekte verschoben werden. Dies muss vom Anwender manuell durchgeführt werden.

4.11.2 Selektion von Netzwerkelementen

Die Selektion erfolgt mittels Mausklick auf das Netzwerkelement. Die selektierten Elemente werden mit gefüllten Rechtecken an den Eckpunkten angezeigt.



Es können mehrere Elemente gleichzeitig selektiert werden. Die Multiselektion erfolgt entweder mit der [SHIFT]-Taste oder durch das Aufziehen eines umgebenden Rechtecks. Die Eigenschaften der selektierten Elemente werden im Property Browser angezeigt. Bei einer Multiselektion werden nur die übereinstimmenden Eigenschaften angezeigt.

4. Grafik Editor

4.11 Arbeiten im Netzwerkeditor

4.11.3 Verschieben von Netzwerkelementen

Selektierte Netzwerkelemente (Instanzen, Elemente der Eingangs- und Ausgangsspalte), können im Netzwerkeditor verschoben werden. Verbindungen können nicht direkt verschoben werden, sondern nur, wenn ein anderes Element verschoben wird. Während des Verschiebens wird die aktuelle Position durch einen strichlierten Rahmen angezeigt. Für den Verschiebevorgang gelten dieselben Einschränkungen wie im Punkt "Einfügen von Netzwerkelementen" beschrieben.

4.11.4 Löschen von Netzwerkelementen

Selektierte Netzwerkelemente können über das Kontextmenü bzw. über die [DEL]-Taste gelöscht werden. Alle angehängten Verbindungen werden ebenfalls entfernt.

4.11.5 Parametrierung von Netzwerkelementen

Die Parameter der einzelnen selektierten Netzwerkelemente können im Property Browser geändert werden. Nicht änderbare Parameter werden grau dargestellt. Einstellungen können entweder durch Eingabe der Werte bzw. durch eine Auswahl aus einer Liste mit vordefinierten Einträgen durchgeführt werden.

4.11.6 Ziehen von Verbindungen

Verbindungen werden mit der Maus gezogen. Dabei sind zwei Mausklicks notwendig, die Maustaste muss während des Ziehens losgelassen werden. Dem Anwender wird durch einen entsprechenden Cursor angezeigt ob eine Verbindung gezogen werden kann. Die Datentypen der beiden Endpunkte müssen übereinstimmen (z.B. SAFEBOOL auf SAFEBOOL) oder kompatibel sein (z.B. CONST auf BOOL). Der Verlauf der Verbindung ergibt sich automatisch, der Anwender hat darauf keinen Einfluss.



Verbindungen zur Eingangs- bzw. Ausgangsspalte können auch über mehrere Editorspalten laufen, wenn der dazwischen liegende Bereich nicht durch ein platziertes Netzwerkelement versperrt ist.

Beim Erstellen einer ungültigen Verbindung kann ein Tooltip angezeigt werden (am Pin, zu welchem die Verbindung gezogen wird).

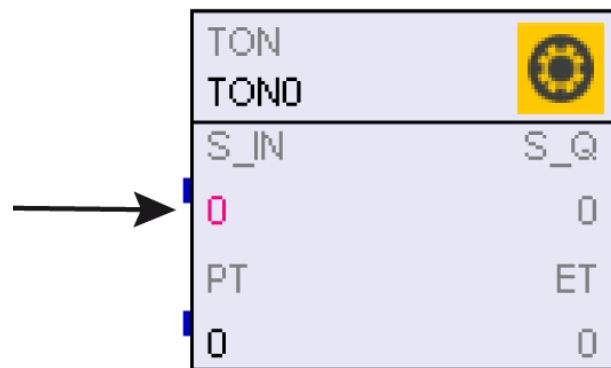
Mögliche Tooltips und deren Erklärung:

Tooltip	Beschreibung
Connection is crossing another connection or an instance	Die neue Verbindung schneidet eine andere Verbindung oder führt über eine Instanz eines Funktionsblocks.
Connection with a dummy module	Es soll eine Verbindung mit einem Dummy-Element erzeugt werden.
Input of instance is connected	Der Eingang einer Instanz eines Funktionsblocks ist bereits verbunden.
Element in output column is connected	Das Element in der Ausgangsspalte ist bereits verbunden.
Connection between identical network elements (e.g. output column with output column)	Es sollen zwei gleiche Elemente verbunden werden (z.B. Element der Ausgangsspalte mit einem Element der Ausgangsspalte).
Connection between different data types	Es sollen zwei verschiedene Datentypen miteinander verbunden werden (z.B. bool mit safebool).
Value of constant element is outside of allowed borders	Der Wert einer Konstante ist außerhalb der erlaubten Grenzen.
Sense of direction of connections must be left to right	Die Verbindung muss immer von links nach rechts laufen.
Connection between two constant elements	Es werden Konstanten mit Konstanten verbunden.

4. Grafik Editor

4.11 Arbeiten im Netzwerkeditor

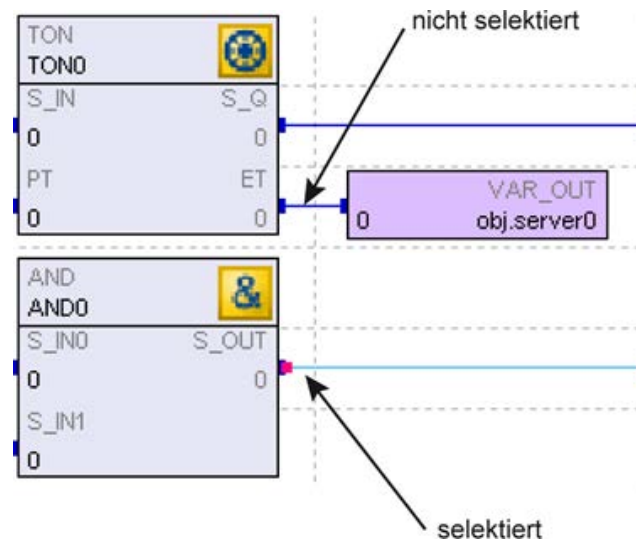
Eingangsparameter eines Funktionsblocks müssen immer verbunden sein (nicht benötigte Eingangsparameter müssen mit Konstanten belegt werden), bei Ausgängen ist eine Verbindung optional. Ist ein notwendiger Parameter nicht gesetzt, wird dies im Netzwerkelement farblich gekennzeichnet.



4.11.7 Selektion von Verbindungen

Die Selektion erfolgt durch einen Mausklick auf die Verbindung.

Die selektierten Verbindungen werden durch ein gefülltes Rechteck an den Enden dargestellt und besitzen eine andere Farbe.

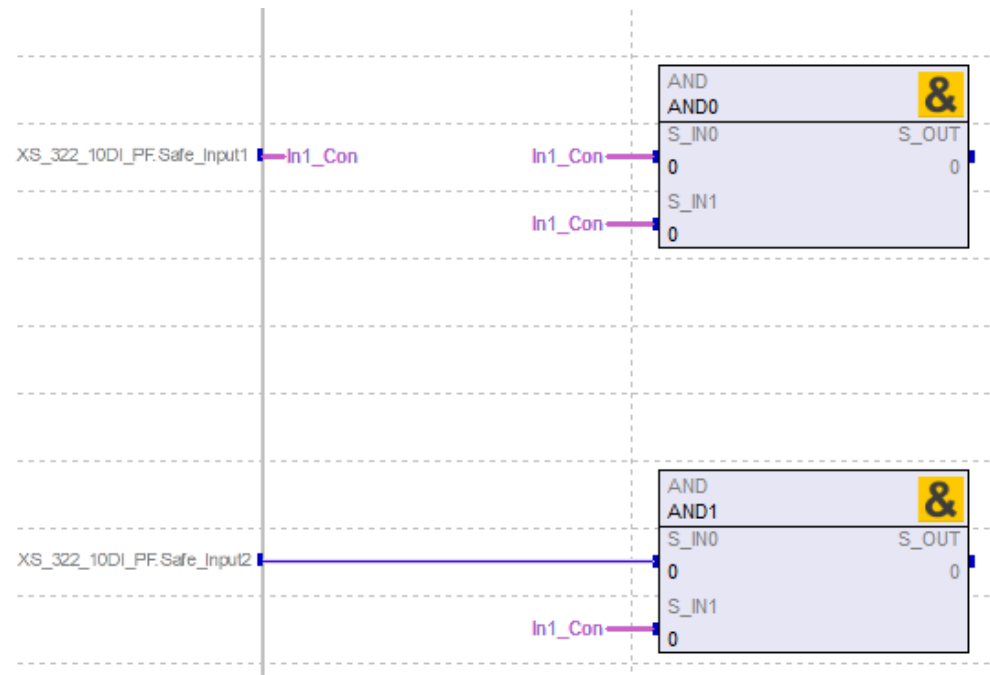


4.11.8 Löschen von Verbindungen

Selektierte Verbindungen können über das Kontextmenü bzw. über die [DEL]-Taste gelöscht werden.

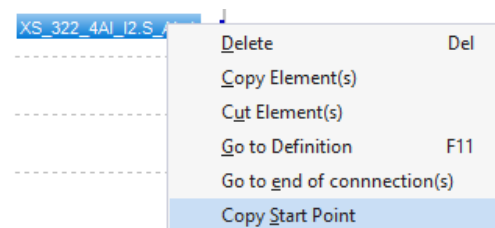
4.11.9 Globale Verbindungen

Bei globalen Verbindungen handelt es sich um eine Verbindung mit einem Label-Namen. Die Verbindungslinien werden nicht mehr gezeichnet. Die Verbindungsstellen werden mit dem Label-Namen gekennzeichnet. In komplexen Netzwerken ist damit eine bessere Übersichtlichkeit zu erreichen.



Erstellen von globalen Verbindungen

Es gibt mehrere Möglichkeiten die globalen Verbindungen zu erstellen und zu verbinden.



Über die Punkte **Copy Start Point** und **Paste Connection** lassen sich Verbindungen einfach an den Verbindungsstellen einfügen.

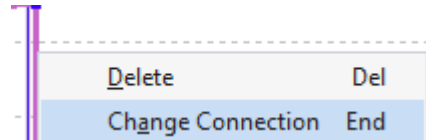
Create Connection erstellt eine temporäre globale Verbindung an der ausgewählten Stelle.

Alle weiteren Verbindungen, die von diesem Punkt weg gehen werden als globale Verbindung erstellt. Die Verbindungen können über die Punkte **Copy** und **Paste** erstellt werden oder über das Ziehen einer Verbindung.

4. Grafik Editor

4.11 Arbeiten im Netzwerkeditor

Über das Kontextmenü auf eine Verbindung kann auch die Verbindungsart gewechselt werden mit **Change Connection**.



Beim Löschen einer globalen Verbindung wird gefragt, ob alle Verbindungen mit dem Label-Namen gelöscht werden sollen.

4.11.10 Einfügen von Kommentaren

Kommentare können aus dem Function Block Tree mittels Drag & Drop in den Netzwerkeditor gezogen werden. Ein strichliertes Rechteck gibt die Größe des einzufügenden Kommentars an, das Cursor-Symbol zeigt an, ob das Element an dieser Stelle eingefügt werden kann.

Darstellung

- Roter Cursor (kann an dieser Stelle nicht eingefügt werden)
- Graues rechteckiges Symbol (kann hier eingefügt werden)

4.11.11 Selektion von Kommentaren

Die Selektion erfolgt mittels Mausklick auf den Kommentar. Der selektierte Kommentar wird mit gefüllten Rechtecken an den Eckpunkten angezeigt.

4.11.12 Verschieben von Kommentaren

Selektierte Kommentare können im Netzwerkeditor verschoben werden. Während des Verschiebens wird die aktuelle Position durch einen strichlierten Rahmen angezeigt. Für den Verschiebevorgang gelten dieselben Einschränkungen wie in → "Einfügen von Kommentaren", Seite 73 beschrieben.

4.11.13 Löschen von Kommentaren

Selektierte Kommentare können über das Kontextmenü bzw. über die [DEL]-Taste gelöscht werden.

4.11.14 Ändern der Kommentargröße

Selektierte Kommentare können in der Größe geändert werden.

Der Cursor muss dazu auf die rechte unter Ecke des Kommentarfeldes gestellt werden; es erscheint ein Pfeil zum Ändern der Größe.

Es kann nun mittels Linksklick und Ziehen der Maus die Größe verändert werden. Die neue Größe wird über einen strichlierten Rahmen angezeigt; sollte die Maus in ungültige Bereiche gezogen werden, wird der strichlierte Rahmen nur für die nächst mögliche gültige Größe angezeigt.

Durch das Loslassen der Maustaste wird die Größe des strichlierten Bereichs übernommen.

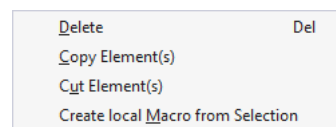


4.11.15 Einfügen von Kommentartexten

Der Text des selektierten Kommentars kann im Property Browser oder mittels [F2]-Taste im Netzwerk geändert werden.

4.11.16 Kopieren, Ausschneiden und Einfügen von Elementen

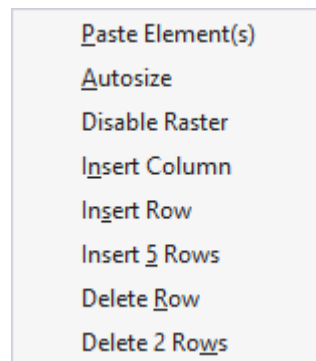
Selektierte Elemente können mittels Kontextmenü oder [STRG][C] aus einem Netzwerk kopiert werden. Ebenso können selektierte Elemente mittels Kontextmenü oder [STRG][X] aus einem Netzwerk ausgeschnitten werden. Der Kopierbefehl ist auch in den Netzwerken einer Bibliothek möglich.



Die kopierten bzw. ausgeschnittenen Elemente können über das Kontextmenü des Netzwerkes oder [STRG][V] in ein Netzwerk eingefügt werden.

4. Grafik Editor

4.12 Aufruf der Definition eines Elements



Die einzufügenden Elemente hängen am Mauscursor und werden gestrichelt dargestellt. Das Cursor-Symbol gibt an, ob an einer Stelle eingefügt werden kann. Elemente, welche beim Einfügen nicht mehr eindeutig sind (z.B. ein sicherer Ausgang eines Safety-Moduls wird ein zweites Mal platziert), werden als Platzhalter eingefügt. Die Darstellung der Platzhalter erfolgt in der Farbe orange.

4.11.17 Ersetzen von Platzhaltern

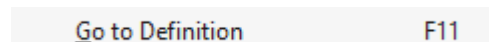
Platzhalter können mittels Drag & Drop von Elementen aus dem Hardware Tree / Function Block Tree oder Variablen Tree ersetzt werden. Das Cursor-Symbol zeigt an, ob ein Platzhalter ersetzt werden kann.

4.11.18 Ersetzen von bestehenden Elementen

Bestehende Elemente im Netzwerk können mittels Drag & Drop von Elementen aus dem Hardware Tree bzw. Variablen Tree und das Drücken der [SHIFT]-Taste ersetzt werden. Das Cursor-Symbol zeigt an, ob ein Element ersetzt werden kann. Instanzen von Funktionsblöcken (z.B. AND-Funktionsblock) können nicht ersetzt werden.

4.12 Aufruf der Definition eines Elements

Es kann mittels Kontextmenü oder [F11] die Definition eines Elements im entsprechenden Tree angesprungen werden.



Dieser Befehl kann für die Elemente der Eingangsspalte, die Instanzen von Funktionsblöcken und die Elemente der Ausgangsspalte aufgerufen werden.

Bei den Elementen der Eingangsspalte/Ausgangsspalte wird entweder der Kanal eines Hardwaremoduls oder eine Variable / temp. Variable im entsprechenden Tree selektiert.

Bei den Elementen in den Netzwerken wird je nach Elementart entweder der Funktionsblock oder die Variable / temp. Variable im entsprechenden Tree selektiert.

4.13 Aufruf der Startpunkte/Endpunkte einer Verbindung

4.13 Aufruf der Startpunkte/Endpunkte einer Verbindung

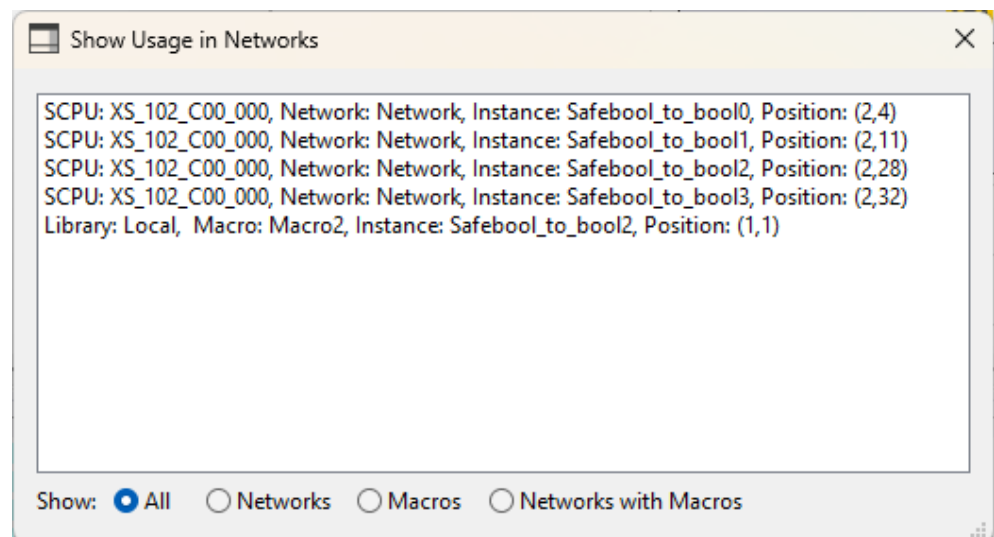
Es kann mittels Kontextmenü ein Anspringen des Startpunktes bzw. ein Anspringen der Endpunkte einer Verbindung ausgeführt werden.

Go to start of connection

Go to end of connection(s)

Das Anspringen des Startpunktes einer Verbindung kann in der Ausgangsspalte oder bei den Eingängen der Funktionsblöcke ausgelöst werden.

Das Anspringen der Endpunkte einer Verbindung kann in der Eingangsspalte oder bei den Ausgängen der Funktionsblöcke ausgelöst werden; sind mehrere Verbindungen vorhanden, wird ein Auswahldialog angezeigt. Bei einer einzelnen Verbindung wird sofort an die jeweilige Stelle im Netzwerk gesprungen. Werden mehrere Verbindungen verwendet, dann kann der Anwender in einem Dialog die gewünschte Stelle auswählen. Die Auswahl erfolgt mittels Doppelklick auf eine Zeile im Dialog.



Um dem Benutzer ein ständiges Aufrufen des Kontextmenüs zu ersparen, kann man alternativ mit den Pfeiltasten durch die Verbindungen navigieren. Mit den Tasten Pfeil nach links bzw. rechts gelangt man zum Anfang bzw. Ende einer Verbindung, aber auch von der Eingangs- zur Ausgangsseite einer Instanz. Die einzelnen Anschlüsse auf einer Seite kann man mit den Tasten Pfeil nach unten bzw. oben erreichen und so weiter durchs Netzwerk wandern. Ist man in der Eingangs- bzw. Ausgangsspalte gelandet, gelangt man mit den Tasten Pfeil nach unten bzw. oben zu einem anderen Element.

Will man von einem Anschluss einer Instanz starten reicht es diese mit der Maus auszuwählen. Bei Konstanten und anderen einzeiligen Elementen kann man sofort weiter navigieren; ansonsten gelangt man mit der Taste Pfeil hinunter in den Bereich mit den Anschlüssen und kann mit den Pfeiltasten einen ansteuern.

4. Grafik Editor

4.14 Aufruf der temporären Variablen

4.14 Aufruf der temporären Variablen

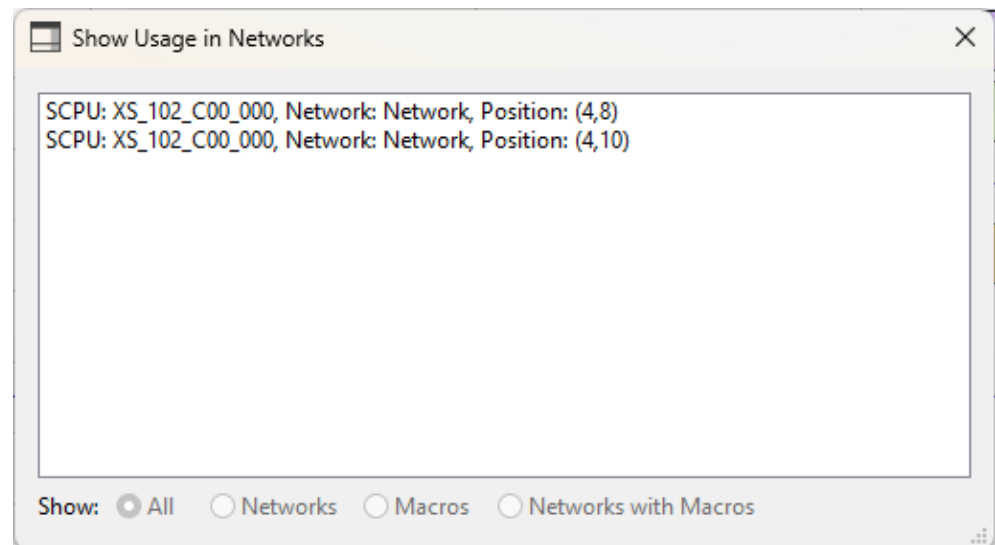
Es kann mittels Kontextmenü ein Anspringen der schreibenden bzw. lesenden Seite von temporären Variablen ausgeführt werden.

Go to Input (Var. is Read)

Go to Output (Var. is Written)

Das Anspringen der schreibenden Seite kann über eine lesende temp. Variable in der Eingangsspalte oder im Netzwerk ausgeführt werden.

Das Anspringen der lesenden Seite kann über eine schreibende temp. Variable in der Ausgangsspalte oder im Netzwerk ausgeführt werden; sind mehrere lesende Elemente vorhanden, wird ein Auswahldialog angezeigt. Bei einer einzelnen Verwendung wird sofort an die jeweilige Stelle im Netzwerk gesprungen. Werden die Variablen mehrfach verwendet, dann kann der Anwender in einem Dialog die gewünschte Stelle auswählen. Die Auswahl erfolgt mittels Doppelklick auf eine Zeile im Dialog.



Ähnlich wie bei den Verbindungen im Netzwerk gelangt man auch hier mit den Pfeiltasten von der schreibenden zur lesenden Seite einer temp. Variable und umgekehrt. Man navigiert auf der Seite ohne Anschluss einfach weiter und landet beim Gegenstück dazu, das funktioniert sowohl im Instanz-Bereich als auch in den Ein- und Ausgangsspalten. Sind mehrere lesende Elemente vorhanden, erscheint auch dabei obiger Dialog.

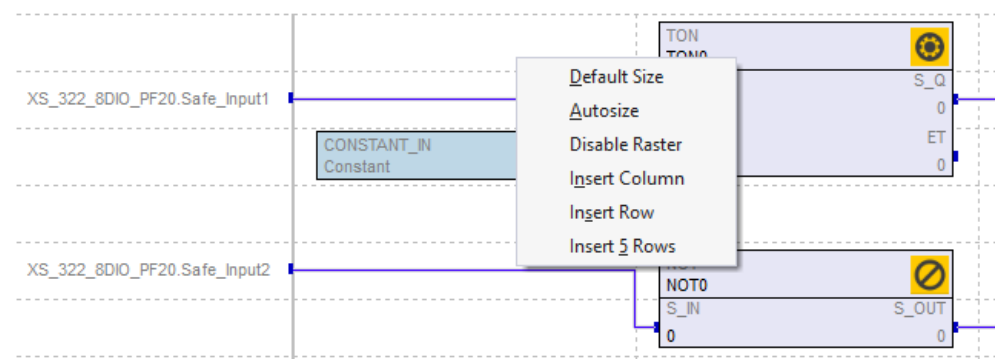
4.15 Zoom - Pan

Für jedes Netzwerk können Zoom- und Verschiebe-Funktionen verwendet werden. Die Auswahl erfolgt über die Edit-Toolbar, → Abschnitt "Werkzeug-Symbolleiste", Seite 38.

Zoom	Bei eingeschalteten Zoom kann mittels linken Mausklick oder  -Taste die Ansicht vergrößert (Zoom In) werden; mittels rechten Mausklick oder  -Taste kann die Ansicht verkleinert (Zoom Out) werden.
Zoom All	Beim Auswählen der Funktion wird das Netzwerk auf eine ganze Seite gezoomt.
Zoom Selection	Beim Auswählen der Funktion muss ein Rechteck zur Selektion des Zoombereichs aufgezogen werden. Es wird auf den selektierten Bereich gezoomt.
Pan	Beim Auswählen der Funktion mittels linken Mausklick und einem Bewegen der Maus kann der sichtbare Bereich verschoben werden.

4.16 Ändern der Spaltenbreite

Es kann mittels Toolbar bzw. Kontextmenü im Netzwerk die Spaltenbreite eines jeden Netzwerks geändert werden. Die Spaltenbreite der Netzwerke kann verschieden sein. Es kann zwischen der *Default Size* (Default-Breite) und *Autosize*, einer automatisch berechneten optimalen Breite, gewählt werden. Die Default-Breite ist zugleich die Mindestbreite einer Spalte. Bei der automatischen Berechnung der Breite werden alle Texte im Netzwerk berücksichtigt; es soll in der Default-Darstellung (ohne Zoom) kein Text mehr abgekürzt werden.



4. Grafik Editor

4.17 Fehlermeldungen beim Einfügen eines Elements ins Netzwerk

4.17 Fehlermeldungen beim Einfügen eines Elements ins Netzwerk

Beim Einfügen eines Elements aus einem ins Netzwerk wird über ein Cursor-Symbol angezeigt, ob das Einfügen möglich ist. Wenn das Einfügen nicht erlaubt ist, wird zusätzlich ein Infotext über den Grund angezeigt.

Mögliche Meldungen und deren Erklärung:

Infotext	Beschreibung
Element is only allowed in input column	Das Element kann nur in der Eingangsspalte eingefügt werden.
SafeDint elements are not allowed in this version of Safety CPU	Das Element enthält den Datentyp SafeDint.
Position is not empty or not allowed	Die Position im Netzwerk ist bereits belegt, bzw. die Position ist nicht erlaubt.
Element is different to dummy element	Das Element passt nicht zum zu ersetzenden Dummy-Element (z.B. unterschiedlicher Datentyp).
Wrong second module for AND-Connection	Bei einer Und-Verknüpfung in der Eingangsspalte passen die Hardwaremodule nicht zusammen (Elemente müssen vom selben Modul sein).
Element is only allowed in input or output column	Das Element kann nur in der Eingangs- oder Ausgangsspalte eingefügt werden.
Element is not allowed in input or output column	Das Element darf nicht in der Eingangs- oder Ausgangsspalte eingefügt werden.
Function blocks with SafeDint data are not allowed in this version of Safety CPU	Der Funktionsblock enthält den Datentyp SafeDint.
Replace element is different to placed element	Beim Replace eines Elements passen die Elemente nicht zusammen (z.B. unterschiedlicher Datentyp).
Output variable can only used once	Eine Ausgangsvariable ist bereits gesetzt und kann nur einmal gesetzt werden.
Temp. variable can only used once as output	Eine temp. Variable ist bereits gesetzt und kann nur einmal gesetzt werden.
Variables are not allowed in a stand alone Safety CPU	Variablen können in einer Stand-Alone Safety-CPU nicht verwendet werden.
Variable is not allowed in input column	Die Variable ist in der Eingangsspalte nicht erlaubt.
Variable is not allowed in output column	Die Variable ist in der Ausgangsspalte nicht erlaubt.
Temp. Variables can only used in one Safety CPU	Temp. Variablen können nur innerhalb einer Safety-CPU verwendet werden.
Write interface variable can only used once	Eine schreibende Interface Variable ist bereits gesetzt und kann nur einmal gesetzt werden.
Element is only allowed in output column	Das Element kann nur in der Ausgangsspalte eingefügt werden.
Output of a module can only used once as output	Ein Ausgang eines Moduls ist bereits gesetzt und kann nur einmal gesetzt werden.

4.17 Fehlermeldungen beim Einfügen eines Elements ins Netzwerk

Infotext	Beschreibung
Outputs of a module can only used by 2 Safety CPUs	Die Ausgänge eines Moduls dürfen nur von zwei verschiedenen Safety-CPU's gesetzt werden.
Multi modules are not allowed in this version of Safety CPU	Multimodule dürfen in dieser Version der Safety-CPU nicht verwendet werden.
A Safety CPU can only used by one multi module	Es kann pro Safety-CPU nur ein Multimodul verwendet werden.
An unsafe variable can only used with object/I/O-Element and server	Eine unsichere Variable kann nur verwendet werden, wenn der Objektnamen / I/O-Elementnamen sowie der Servernamen gesetzt sind.
HGW Modules only allowed in own Safety CPU	Elemente von HGW-Modulen können nur in der eigenen Safety-CPU verwendet werden.
AND-Connections are not allowed for node channels	Und-Verknüpfungen sind für Node-Module nicht erlaubt
Element is not allowed in macro network	Das Element darf in kein Makro-Netzwerk eingefügt werden
Macro must have at least one input and one output	Ein Makro muss mindestens einen Eingang und einen Ausgang haben
Constant variables are not allowed in this version of Safety CPU	Konstanten sind in dieser Version der Safety-CPU nicht erlaubt
Module only allowed in one Safety CPU	Drive-Module sind nur in einer Safety-CPU erlaubt
Write enable of a section can only used once	Das Write-Enable-Flag der Parameterliste darf nur einmalig verwendet werden
Only BDINT or CAN sub variables are allowed	Es sind nur drei Subvariablen von BDINT- und CAN-Variablen erlaubt
Parameters from parameter list are not allowed in this version of Safety CPU	Parameter von Parameterlisten sind in dieser Version der Safety-CPU nicht erlaubt
Function blocks with Array data not allowed in this version of Safety CPU	Funktionsblöcke mit Arrays sind in dieser Version der Safety-CPU nicht erlaubt
Arrays not allowed in this version of Safety CPU	Arrays sind in dieser Version der Safety CPU nicht erlaubt

4. Grafik Editor

4.18 Einstellparameter im Projekt

4.18 Einstellparameter im Projekt

4.18.1 Temporary Operation Time

Einheit	Minuten (5 Minuten bis 8 Stunden, Standardwert 1 Stunde)
Bedeutung	Der Zeitraum, in dem die Safety-CPU im temporären Operationmode läuft. Nach Ablauf der Zeitspanne schaltet die Safety-CPU ab, wenn nicht vorher verifiziert wurde.
Einstellort	Root-Element des Hardware Trees, Project Trees und Variable Trees

4.18.2 Max. Cycle Time

Einheit	Millisekunden (1 ms bis 50 ms, Standardwert 3 ms)
Bedeutung	Überwachte Zykluszeit einer Safety-CPU
Einstellort	Safety-CPU Modul(e) im Hardware Tree und Project Tree

4.18.3 Maximum Transmission Time

Einheit	Millisekunden (1 ms bis 100 ms, Standardwert 10 ms)
Bedeutung	Wert wird zur Überprüfung des maximalen Alters von übertragenen Informationen herangezogen. Der Wert gibt die maximale Zeit für die Busübertragung an, für das maximale Alter werden zusätzlich noch die Zykluszeiten von Sender und Empfänger eingerechnet.
Einstellort	Safety-Module im Hardware Editor bzw. sichere Eingänge und Ausgänge im grafischen Editor

4.18.4 Maximum Transmission Time (Output as Input)

Einheit	Millisekunden (1 ms bis 100 ms, Standardwert 10 ms)
Bedeutung	Wert wird zur Überprüfung des maximalen Alters von rücklesenden Ausgängen herangezogen. Der Wert gibt die maximale Zeit für die Busübertragung an, für das maximale Alter werden zusätzlich noch die Zykluszeiten von Sender und Empfänger eingerechnet.
Einstellort	Safety-Module im Hardware Editor bzw. sichere rücklesende Ausgänge im grafischen Editor

4.18.5 Filter Time

Einheit	Millisekunden (0 ms bis 100 ms, Standardwert 0 ms)
Bedeutung	Zeitwert für Softwarefilter bei sicheren Eingängen
Einstellort	Sichere Eingänge im Hardware Tree bzw. im grafischen Editor

4.18.6 Cross Circuit Detection

Einheit	einheitenlos, Auswahl von 0, A oder B (Standardwert 0)
Bedeutung	Querschlusserkennungskreis bei sicheren Eingängen (keine Querschlusserkennung, Kreis A oder Kreis B)
Einstellort	Sichere Eingänge im Hardware Tree bzw. im grafischen Editor

4.18.7 Hardware Revision

Einheit	einheitenlos, Auswahl von Hardware-Revisionen - kleiner als HW: 4.0 bedeutet Hardware vor der Revision 4.0 - größer oder gleich HW: 4.0 bedeutet Hardware ab der Revision 4.0
Bedeutung	Berechnung der überwachten Zykluszeit einer Safety-CPU in Abhängigkeit von der eingestellten Hardware Revision. Die Formeln für die Berechnungen sind gleich. Ab der Revision 4.0 wird die berechnete Zeit um 15 % erhöht.
Einstellort	Safety-CPU Modul(e) im Hardware Tree und Project Tree. Auswahl steht nur bei Safety-CPU's zu Verfügung, bei welchen eine unterschiedliche Berechnung erforderlich ist.

4.18.8 Parameter Revision

Einheit	einheitenlos, Auswahl von Parameter-Revisionen - kleiner als HW: 2.0 bedeutet Parameter vor der Hardware Revision 2.0 - größer oder gleich HW: 2.0 bedeutet Parameter ab der Hardware Revision 2.0
Bedeutung	Die eingestellte Parameter Revision gibt an, welche Parameter für ein Modul zur Verfügung stehen. Die Parameter werden im Property Browser entsprechend angezeigt.
Einstellort	Safety-Modul(e) im Hardware Tree. Die Auswahl steht nur bei Safety-Modulen zu Verfügung, bei welchen unterschiedliche Parameter eingestellt werden können.

4. Grafik Editor

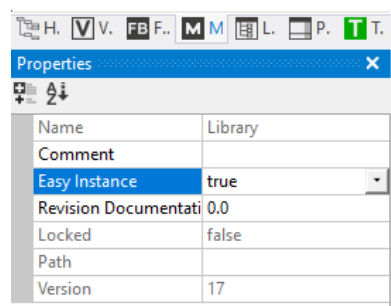
4.18 Einstellparameter im Projekt

4.18.9 Vereinfachte Darstellung ausgewählter Funktionsblöcke

Es besteht die Möglichkeit einige Funktionsblöcke in einer vereinfachten Form darzustellen. Die Darstellung erfolgt dann ohne die Namen.



Die Einstellung kann im Property Browser des Projekts durchgeführt werden. Der Parameter „Easy Instance“ muss dazu auf true gestellt werden. Die Einstellung kann auch in der Library von Macros vorgenommen werden.



Folgende Funktionsblöcke können in vereinfachter Form dargestellt werden.

Funktionsblock	Darstellung
AND	&
OR	
XOR	⊕
NOT	⊘
Bool to Safebool	B2SB
Dint to SafeDint	D2SD
SError to SFalse	E2SBF
Safebool to bool	SB2B
SafeDint to Dint	SD2D
SafeDintError to SafeDint	SDE2SD



Aus einer Easy-Instance-Macro-Library kann man keine Macros in ein normales Projekt ziehen, man muss vorher Macro-Library oder Projekt umstellen.



Will man das Projekt oder die Macro-Library auf normale Darstellung der Funktionsblöcke zurückstellen, muss sichergestellt sein, dass in den Netzwerken genügend Platz für die Header der Funktionsblöcke vorhanden ist.

4.18.10 Aktivieren des Schreibens der CRC

Einheit	einheitenlos, Auswahl von TRUE oder FALSE (Standardwert TRUE)
Bedeutung	Flag, ob die CRCs für Applikation und Hardware-Pfad auf die Safety-CPU geschrieben werden soll
Einstellort	Root-Element des Hardware Tree, Project Tree und Variable Tree

4.18.11 Aktivieren der Easy Instance Einstellung für Instanzen

Einheit	einheitenlos, Auswahl von TRUE oder FALSE (Standardwert FALSE)
Bedeutung	Flag, ob ausgewählte Instanzen ohne Header dargestellt werden
Einstellort	Root-Element des Hardware Tree, Project Tree und Variable Tree

4.18.12 Aktivieren der Einstellung für einen Download mit Zusatzinformationen

(entweder als Text oder nur als CRC über den Text); Bei großen Projekten werden dadurch die Downlaod-Zeiten verringert und der Flash auf der Safety-CPU weniger belastet.

Einheit	einheitenlos, Auswahl von TRUE oder FALSE (Standardwert FALSE)
Bedeutung	Flag, ob Zusatzinfo als kompletter Text oder nur als CRC mit gespeichert wird
Einstellort	Root-Element des Hardware Tree, Project Tree und Variable Tree

4.18.13 Deaktivieren des Blinkens einer LED (lokale Module)

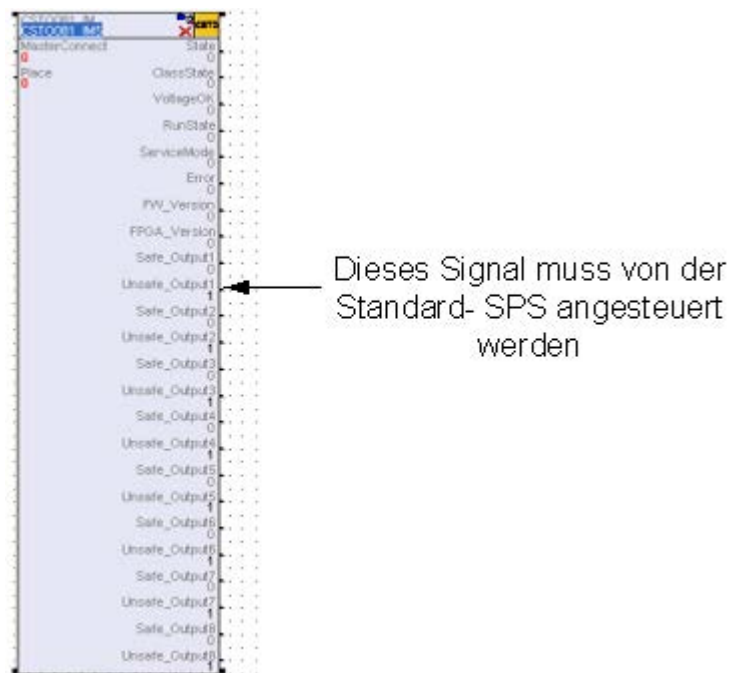
Einheit	einheitenlos, Auswahl von TRUE oder FALSE (Standardwert FALSE)
Bedeutung	Flag, ob das Blinken einer roten LED beim Modul deaktiviert wird; nur erlaubt, wenn lokale optionale Module nach der Verifikation fehlen.
Einstellort	Safety-Modul(e) im Hardware Tree. Die Auswahl steht nur bei Safety-Modulen zur Verfügung, bei welchen das Blinken der LED deaktiviert werden kann.

4. Grafik Editor

4.19 Enable Signal von sicheren Ausgängen

4.19 Enable Signal von sicheren Ausgängen

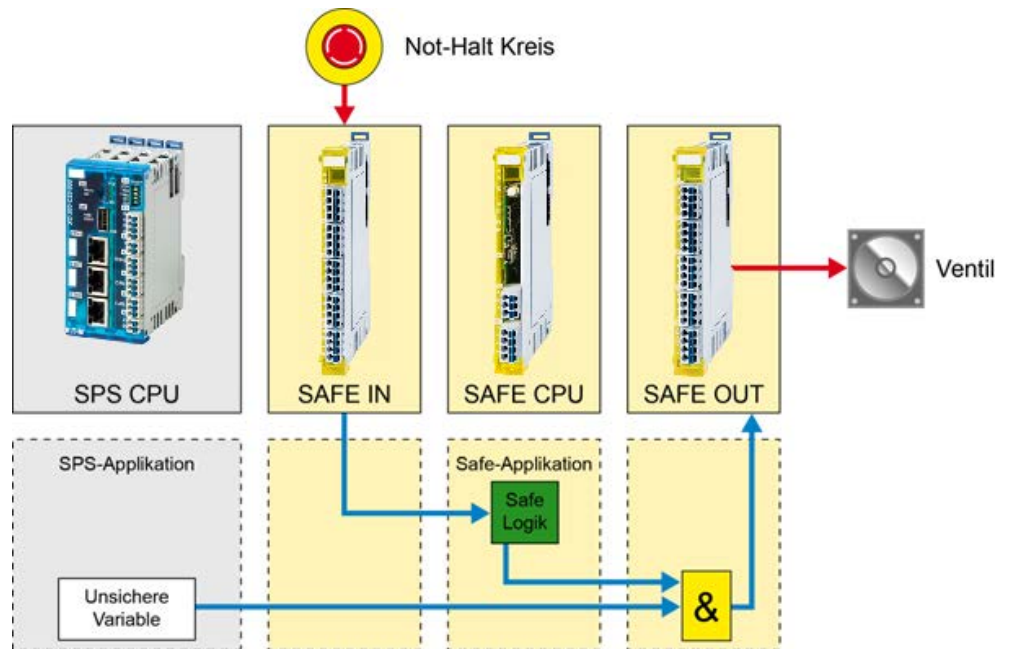
Es kann bei den sicheren Ausgängen im Safety Designer eine Und-Verknüpfung mit einem unsicheren Freigabesignal eingestellt werden. Unter dem Freigabesignal wird eine unsichere Variable verstanden, welche nicht im Safety-Projekt vorhanden ist, sondern direkt in der unsicheren Applikation beim entsprechenden Hardware-Objekt eingestellt wird.



Das unsichere Freigabesignal kann einen Ausgang nur in den sicheren Zustand versetzen (Ausgang wird abgeschaltet). Das unsichere Freigabesignal wird bei jedem Bus-Takt von den Hardware-Klassen im Safety-Modul gesetzt und dort in einem Interrupt abgearbeitet. Es ist somit möglich einen sicheren Ausgang über die unsicheren Applikationen schnell abzuschalten.

4. Grafik Editor

4.19 Enable Signal von sicheren Ausgängen



Im XN Safety Designer muss für die Verwendung des unsicheren Freigabesignals das Flag „Enable signal used“ im Property Browser auf true gesetzt werden. Der Wert kann entweder beim Ausgang im Hardware-Baum oder in der Ausgangsspalte des Netzwerkes gesetzt werden. Das Flag kann nur gesetzt werden, wenn der Ausgang verwendet wird. Der Ausgang wird im Netzwerk mit einem & Symbol gekennzeichnet.

DO.Safe_Output2

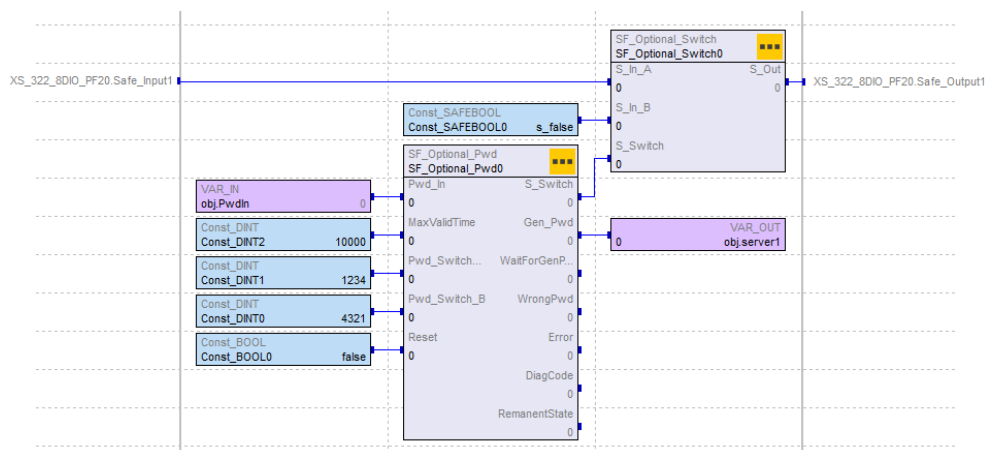


4.20 Optionale Module

Im XN Safety Designer können Safety-Module über den Property Browser auf optional geschaltet werden. Das bedeutet, dass einzelne Module für die Abarbeitung der Konfiguration auf der Safety-CPU nicht vorhanden sein müssen. Für die Daten dieser Module wird ein vom Anwender voreingestellter Wert verwendet. Die Verwendung von optionalen Modulen ist an Voraussetzungen geknüpft.

In der Sicherheitsapplikation werden nach den Eingängen des optionalen Moduls und der restlichen Logik spezielle Funktionsblöcke (SF_Optional_Pwd/SF_Optional_PwdII und SF_Optional_Switch) platziert, die im Falle eines fehlenden Moduls definierte Werte (auch solche, die dem nicht sicheren Zustand entsprechen) an die dahinter liegende Logik weiterreichen.

➔ Es ist hier zu beachten, dass eine gefährliche Situation eintreten könnte, wenn der Zustand dieser Funktionsblöcke (bezüglich des Vorhandenseins eines optionalen Moduls) nicht mit der Realität übereinstimmt.



Jeder verwendete Eingang eines optionalen Moduls oder eines lesenden Interfaces muss über eine Instanz des Funktionsblocks SF_Optional_Switch geschaltet werden. Die verwendeten Eingänge müssen mit dem ersten Eingang des Funktionsblocks verbunden werden. Der zweite Eingang des Funktionsblocks muss mit einem Defaultwert (z.B. Konstante) verbunden werden. Dieser Wert wird verwendet, wenn das optionale Modul weg geschaltet wurde. Der dritte Eingang muss mit dem ersten Ausgang des Funktionsblocks ST_Optional_Pwd/SF_Optional_PwdII verbunden werden und gibt an, ob der erste oder zweite Eingang verwendet werden soll. Mit Hilfe des Funktionsblocks SF_Optional_Pwd/SF_Optional_PwdII wird, durch Passwörter geschützt, die Umschaltung zwischen optional und nicht optional geprüft. Die Eingabe der Passwörter für die Umschaltung erfolgt über eine unsichere Variable aus

der unsicheren Applikation und muss vom Anwender über ein zusätzliches (vom Funktionsblock generiertes) Passwort bestätigt werden.



Das System ist nach jedem Wechsel zwischen optional und nicht optional vom Anwender auf Richtigkeit zu prüfen.



Der errechnete Code darf nicht kopierbar sein (beispielsweise durch „copy and paste“), sondern muss nach manueller Eingabe bestätigt werden.

Sicherheitsmaßnahmen und -hinweise

Durch die Verwendung des Funktionsblocks „SF_OPTIONAL_PWDII“ werden grundsätzlich alle Sicherheitsfunktionen betroffen; daher sind unbedingt die nachfolgenden Maßnahmen und Gefahrenhinweise zu berücksichtigen. Die Nichtbeachtung dieser Warnhinweise kann zu schweren Körperverletzungen oder zum Tod führen.



GEFAHR

Der Anwender der Maschine muss im Hinblick auf die möglichen Gefahren, die von der Verwendung des Funktionsblock „SF_OPTIONAL_PWDII“ ausgehen können, und unter Berücksichtigung der für die spezifische Anwendung geltenden normativen Anforderungen (beispielsweise hydraulische Gesenkbiegepressen) eine Risikoanalyse erstellen und die erforderlichen Maßnahmen der Risikominderung treffen.

Die Gültigkeitsdauer des von der Safety-CPU errechneten Codes (Anmeldezeit) muss in jedem Fall so kurz wie möglich sein; insbesondere muss sichergestellt werden, dass sie nicht schichtübergreifend wirksam ist.

Es muss gewährleistet sein, dass Passworteingabe und Bestätigung des automatisch errechneten Codes von ein und demselben autorisierten Einrichter vorgenommen wird.

Die Maschine muss während der Gültigkeitsdauer des automatisch errechneten Codes permanent im Blickfeld des Einrichters der Maschine sein.

Außer dem Einrichter der Maschine dürfen sich innerhalb der Grenzen der Maschine gemäß EN 12100 keine weiteren Personen aufhalten. Insbesondere muss der freie Zugang zur Maschine durch geeignete Maßnahmen verhindert werden.

Befehlsgeräte ohne NOT-HALT-Funktion sind so zu installieren, dass von jeder Position aus in Handreichweite jederzeit ein NOT-HALT ausgelöst werden kann.

4. Grafik Editor

4.21 Validierung und Generierung der Downloaddaten

4.21 Validierung und Generierung der Downloaddaten

Der Safety Designer erstellt für den Anwender die Download-Daten aus der Applikation. Zur Generierung der Daten wird die erstellte Applikation durch den Safety Designer auf Richtigkeit überprüft. Bei ermittelten Fehlern in der Applikation werden im Output-Fenster rot hinterlegte Fehlermeldungen ausgegeben.

Ein Doppelklick auf die Fehlermeldung selektiert in den meisten Fällen die entsprechende Stelle im Netzwerk;

bei Fehlern, welche keine direkte Selektion in einem Netzwerk ermöglichen, muss über die Fehlermeldung alleine auf das Problem rückgeschlossen werden.

Mögliche Fehlermeldungen und deren Erklärung:

Fehlermeldung	Beschreibung
SafeCPU=<Name>, Network=<Name>, Object=<Name>, Input=<Name> is not connected.	Der Eingang einer Instanz eines Funktionsblocks in einem Netzwerk einer Safety-CPU ist nicht verbunden. Maßnahme: Verbindung zum Eingang der Instanz erstellen.
SafeCPU=<Name>, Network=<Name>, Object= <Name>, Output=<Name> is not connected.	Der Ausgang einer Instanz eines Funktionsblocks in einem Netzwerk einer Safety-CPU ist nicht verbunden. Maßnahme: Verbindung zum Ausgang der Instanz erstellen.
SafeCPU=<Name>, Network=<Name>, Object=<Name> is not connected.	Eine Instanz eines Funktionsblocks in einem Netzwerk einer Safety-CPU ist nicht verbunden. Maßnahme: Verbindungen zu den Eingängen bzw. Ausgängen der Instanz erstellen.
SafeCPU=<Name>, Network=<Name>, Row= <Zeilennummer>, Module1=<Name> is not equal Module2 =<Name>.	Die Und-Verknüpfung in der Eingangsspalte ist fehlerhaft; die Eingänge für die Verknüpfung müssen von einem Modul stammen. Maßnahme: Und-Verknüpfung aus nur einem Modul erstellen.
SafeCPU=<Name>, Network=<Name>, Row= <Zeilennummer>, Name1=<Name> is equal Name2 =<Name>.	Die Und-Verknüpfung in der Eingangsspalte ist fehlerhaft; es wurde derselbe Eingang eines Moduls zweimal verwendet. Maßnahme: Und-Verknüpfung aus 2 verschiedenen Eingängen eines Moduls erstellen.
SafeCPU=<Name>, Network=<Name>, Input <Name> in row <Zeilennummer> has different configurations to other input.	Mehrfach verwendete Elemente in der Eingangsspalte eines Netzwerks einer Safety-CPU haben verschiedene Einstellungen (z.B. Querschluß, Filterzeit). Maßnahme: Einstellungen angleichen.
SafeCPU=<Name>, Network=<Name>, Input <Name> in row <Zeilennummer> is not connected.	Ein Element der Eingangsspalte ist nicht verbunden. Maßnahme: Verbindung zum Element der Eingangsspalte erstellen.

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
SafeCPU=<Name>, Network=<Name>, Output <Name> in row <Zeilennummer> is not connected.	Ein Element der Ausgangsspalte ist nicht verbunden. Maßnahme: Verbindung zum Element der Ausgangsspalte erstellen.
SafeCPU=<Name>, Network=<Name>, Output =<Name.Servername> is used multiple times.	Ein Ausgang ist mehrfach verwendet. Ausgänge dürfen nur einmal beschrieben werden. Maßnahme: Ausgang nur einmal verwenden.
SafeCPU=<Name>, Network=<Name>, Connection <Startposition to Endposition> is invalid, start column is higher than end column.	Eine Verbindung läuft von rechts nach links. Startpunkt rechts des Endpunktes. Maßnahme: Netzwerk löschen und neu erstellen.
SafeCPU=<Name>, Network=<Name>, Position = (Wert) instance not found.	Eine Instanz an der angegebenen Position ist nicht zu finden. Maßnahme: Netzwerk löschen und neu erstellen.
SafeCPU=<Name>, Network=<Name>, Connection <Startposition to Endposition>; Connection of constant block is invalid (delete connection and reconnect).	Es besteht eine Verbindung zwischen einer änderbaren Konstanten und einem konstanten Eingang. Maßnahme: Verbindung löschen und neu erstellen.
SafeCPU=<Name>, Network=<Name>, Connection <Startposition to Endposition> has different data types (delete connection).	Die Datentypen der Start- und Endposition der Verbindung passen nicht zusammen. Maßnahme: Verbindung löschen.
SafeCPU=<Name>, Network=<Name>, Instance= <Namen>; init value is out of range.	Der eingestellte Wert einer Konstanten ist außerhalb des gültigen Bereichs Maßnahme: Gültigen Init-Wert einstellen.
Temp. variable <Name> is never set.	Eine temp. Variable wird als Eingang benützt, ohne gesetzt worden zu sein. Maßnahme: Temp. Variable als Ausgang setzen.
SafeCPU=<Name>, Network=<Name>, Element of temp. variable <Name> with invalid sequence. Sequence of networks in Safety-CPU <Name> cannot be dissolved. Temp. variables are used before they are set.	Es kann keine eindeutige Abarbeitungsreihenfolge der Netzwerke mit temp. Variablen erstellt werden. Es müssen erst alle Netzwerke mit den temp. Variablen als Ausgang in den Download-Daten abgearbeitet werden. Wird eine temp. Variable als Eingang benötigt, bevor das Netzwerk mit der temp. Variablen als Ausgang abgearbeitet wurde, wird eine Fehlermeldung generiert. Es wird daher bei der Prüfung der temp. Variablen eine Reihenfolge der Netzwerke ermittelt. Die Reihenfolge muss eindeutig sein. Es wird die temp. Variable als ‚Fehlervariable‘ angegeben, welche als erstes nicht mehr bearbeitet werden kann. Es darf keine Überkreuzung bei der Verwendung der temp. Variablen in den Netzwerken vorkommen. Maßnahme: Verwendung aller temp. Variablen überprüfen und entsprechend neu setzen.

4. Grafik Editor

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
SafeCPU=<Name>, Network=<Name>; Using sequence of temp. variable <Name> is invalid. Temp variable is used before it is set.	Eine temp. Variable wird als Eingang verwendet, bevor die Variable als Ausgang gesetzt wurde. Maßnahme: Ausgang setzen, bevor die temp. Variable als Eingang verwendet wird.
Temp variables <Name> can only be used within one SafeCPU. Variables used in Safety-CPU: <Name>	Eine temp. Variable wird in mehreren Safety-CPU's verwendet. Eine temp. Variable darf nur in einer Safety-CPU verwendet werden. Maßnahme: Löschen der Instanzen der temp. Variablen aus den anderen Safety-CPU's.
SafeCPU=<Name>, Network=<Name>, Output=<Name> in Row= <Zeilennummer> has wrong domain number.	Eine unsichere Variable hat eine nicht vorhandene Domainnummer eingestellt. Maßnahme: Domain-Nummer bei der unsicheren Variablen ändern.
SafeCPU=<Name>, Network=<Name>, Instance=<Name> has wrong domain number.	Eine unsichere Variable hat eine nicht vorhandene Domainnummer eingestellt. Maßnahme: Domain-Nummer bei der unsicheren Variablen ändern.
PLC <Name> with domain number 0 has a invalid connection to module <Name>.	PLC mit der Domain-Nummer 0 ist nicht die oberste Standard-SPS. Maßnahme: Domain-Nummer ändern oder Aufbau der Hardware ändern.
PLC <Name> has no connection to a other PLC.	PLC hat eine Domain-Nummer ungleich 0, ist aber mit keiner anderen Standard-SPS verbunden. Maßnahme: Verbindung zu einer anderen Standard-SPS erstellen.
Domain number <Nummer> is used in PLC <Name> and <Name>.	Domain-Nummer ist in 2 Standard-SPS verwendet. Maßnahme: Domain-Nummer bei einer Standard-SPS ändern.
Domain number 0 must be used in project.	Die Domain-Nummer 0 ist im Projekt nicht vorhanden. Maßnahme: Domain-Nummer 0 bei der ersten Standard-SPS einstellen.
SafeCPU=<Name>, Network=<Name>, Input=<Name> in Row= <Zeilennummer> is a dummy element.	Ein Element der Eingangsspalte ist ein Platzhalter. Maßnahme: Platzhalter ersetzen.
SafeCPU=<Name>, Network=<Name>, Output=<Name> in Row= <Zeilennummer> is a dummy element.	Ein Element der Ausgangsspalte ist ein Platzhalter. Maßnahme: Platzhalter ersetzen.
SafeCPU=<Name>, Network=<Name>, Instance=<Name> is a dummy element.	Eine Instanz im Netzwerk ist ein Platzhalter. Maßnahme: Platzhalter ersetzen.

4. Grafik Editor

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
SafeCPU=<Name>, Network=<Name>, Object=<Name>, Out- put=<Name> is remanent. Rema- nent outputs are not supported by module=<Name>.	Eine Instanz eines Funktionsblocks hat einen remanenten Server. Rema- nente Server werden vom Modul nicht unterstützt. Maßnahme: Instanz löschen oder Safety-CPU durch eine Safety-CPU, wel- che remanente Server unterstützt, ersetzen.
SafeCPU=<Name>, Network=<Name>, Object=<Name> is a interface variable. Interface variables are not supported by modu- le=<Name>.	Eine Instanz eines Funktionsblocks ist ein Interface. Interface Variablen werden vom Modul nicht unterstützt. Maßnahme: Instanz löschen oder Safety-CPU durch eine Safety-CPU, wel- che Interface Variablen unterstützt, ersetzen.
SafeCPU=<Name>, Network=<Name>, Object=<Name> is a bit variable. bit variables are not supported by module=<Name>.	Eine Instanz eines Funktionsblocks ist eine Bit-Variablen. Bit-Variablen wer- den vom Modul nicht unterstützt. Maßnahme: Instanz löschen oder Safety-CPU durch eine Safety-CPU, wel- che Bit Variablen unterstützt, ersetzen.
SafeCPU=<Name>, Network=<Name>, Input <Name> is a bit variable. Bit variables are not supported by module=<Name>.	Ein Eingangselement ist eine Bit-Variablen. Bit-Variablen werden vom Modul nicht unterstützt. Maßnahme: Eingangselement löschen oder Safety-CPU durch eine Safety- CPU, welche Bit Variablen unterstützt, ersetzen.
SafeCPU=<Name>, Network=<Name>, Output <Name> is a bit variable. Bit variables are not supported by module=<Name>.	Ein Ausgangselement ist eine Bit-Variablen. Bit-Variablen werden vom Modul nicht unterstützt. Maßnahme: Ausgangselement löschen oder Safety-CPU durch eine Safety-CPU, welche Bit Variablen unterstützt, ersetzen.
Can not find interface module of interface(read) <Name>	Das eingestellte Interface-Modul kann im Projekt nicht gefunden werden. Maßnahme: Anderes Interface-Modul einstellen.
SafeCPU=<Name>, Network=<Name>, Instance=<Name> is a interface variable. Interface variables are not supported by standalone PLC <Name>.	Die Instanz ist eine Interface Variable. Interface Variablen werden von einer Stand-Alone PLC nicht unterstützt. Maßnahme: Instanz löschen.
SafeCPU=<Name>, Network=<Name>, Instance=<Name> is a variable. Variables are not supported by standalone PLC <Name>.	Die Instanz ist eine Variable. Variablen werden von einer Stand-Alone PLC nicht unterstützt. Maßnahme: Instanz löschen.
SafeCPU=<Name>, Network=<Name>, Input=<Name> in row= <Zeilennummen> is a variable. Variables are not supported by standalone PLC <Name>.	Das Eingangselement ist eine Variable. Variablen werden von einer Stand-Alone PLC nicht unterstützt. Maßnahme: Eingangselement löschen.

4. Grafik Editor

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
SafeCPU=<Name>, Network=<Name>, Output=<Name> in row= <Zeilennummen> is a variable. Variables are not supported by standalone PLC <Name>.	Das Ausgangselement ist eine Variable. Variablen werden von einer Stand-Alone PLC nicht unterstützt. Maßnahme: Ausgangselement löschen.
Module <Name> is no child of module <Name>.	Das Unterelement ist bei diesem Modul nicht erlaubt. Maßnahme: Elemente löschen.
Position of Module <Name> is wrong.	Die Position des Unterelements ist bei diesem Modul falsch. Maßnahme: Elemente löschen oder verschieben.
Module <Name> is wrong. It is not allowed to leave gaps bet- ween two modules of the same bus type.	Es ist nicht erlaubt, leere Positionen zwischen zwei Modulen am selben Bussystem zu haben. Maßnahme: Elemente löschen oder verschieben.
Module <Name> is wrong. Order of child elements is not correct.	Die Reihenfolge der Unterelemente ist falsch. Maßnahme: Elemente löschen oder verschieben.
Only one PLC as standalone module allowed, PLC <Name> is wrong.	Es ist nur eine Stand-Alone PLC erlaubt. Maßnahme: PLC löschen.
Module <Name> has no master module.	Bei einem Modul im Slave-Modus ist kein Modul im Master-Modus gesetzt. Maßnahme: Modul setzen.
Master <Name> of module <Name> not found.	Bei einem Modul im Slave-Modus ist ein falsches Modul im Master- Modus gesetzt. Maßnahme: Anderes Master-Modul setzen.
Module <Name> has different max. transmission times, only one time allowed.	Ein Modul hat bei den Eingängen/Ausgängen verschiedene max. Über- tragungszeiten eingestellt. Maßnahme: Zeiten angleichen.
Module <Name> has different max. transmission times for out- put as input, only one time allo- wed.	Ein Modul hat bei den rücklesenden Ausgängen verschiedene max. Über- tragungszeiten eingestellt. Maßnahme: Zeiten angleichen.
Invalid parameter <Name> of module <Name>. Value has to be within range <Wert> - <Wert>.	Der Wert eines dynamischen Parameters ist außerhalb der Grenzen. Maßnahme: Wert innerhalb der Grenzen angeben.
Invalid parameter <Name> of module <Name>. X-Value has to be within range <Wert> - <Wert>. Y-Value has to be within range <Wert> - <Wert>.	Die Werte eines dynamischen Parameters sind außerhalb der Grenzen. Maßnahme: Werte innerhalb der Grenzen angeben.
Invalid parameter <Name> of module <Name>. Unkown data type <Name>.	Der Datentyp eines dynamischen Parameters ist nicht bekannt. Maßnahme: Parameter durch einen anderen Parameter ersetzen.
Number of used enable signals of module <Name> is too high (max. number is 16).	Die Anzahl der sicheren Freigabesignale bei den Ausgängen ist zu hoch. Maßnahme: max. 16 Freigabesignale verwenden.

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
Number of bit variables are not supported by module <Name>. Only 8 bits allowed.	Die Anzahl der Bit-Variablen einer schnellen Variable ist zu hoch. Es sind max. 8 Bit erlaubt. Maßnahme: Variable löschen und erneut erstellen.
SafeCPU=<Name>, Network=<Name>, Interface=<Name> has a safe dint variable. Safe dint variables are not allowed.	Die Safety-CPU unterstützt keine SafeDint-Variablen. Maßnahme: Löschen des Interfaces im Netzwerk.
SafeCPU=<Name>, Network=<Name>, Module=<Name> has an safe dint channel. Safe dint channels not supported by module <Name>.	Die Safety-CPU unterstützt keine SafeDint-Variablen. Maßnahme: Löschen des Elements im Netzwerk.
SafeCPU=<Name>, Network=<Name> Object=<Name> has wrong revision of function block (must be lower than 2.0)	Die Safety-CPU unterstützt die Version des Funktionsblocks nicht. Maßnahme: Löschen der Instanz im Netzwerk.
Module<Name>: Number of modules at bus <Name> is too high (max. number: <Anzahl>). Module(s) <Namen> needs more than one place at bus. Delete one or more modules.	Es können nur eine bestimmte Anzahl von Modulen platziert werden. Es ist mindestens ein Modul dabei, welches mehr als einen Platz am Bus belegt (z.B. Relais). Maßnahme: Löschen von Modulen im Hardware Tree.
Module <Name> is not allowed in standalone mode.	Ein Modul ist im Stand Alone-Betrieb nicht erlaubt. Maßnahme: Löschen des Moduls.
Slave address <Number> of module <Name> is outside of allowed ranges.	Die Slave-Adresse liegt außerhalb der erlaubten Grenzen. Maßnahme: Adresse innerhalb der Grenzen vergeben.
Connection ID <Number> of module <Name> is out-side of allowed ranges.	Die Connection ID liegt außerhalb der erlaubten Grenzen. Maßnahme: Connection ID innerhalb der Grenzen vergeben.
Module <Name> has no FSoE master for slave address <Number>.	Es existiert kein FSoE-Master für eine angegebene Slave Adresse. Maßnahme: Modul löschen.
Connection ID <Number> in module <Name> is used multiple times.	Eine Connection ID ist mehrfach verwendet. Maßnahme: Eindeutige Connection ID's erzeugen.
Slave address <Number> in module <Name> is used multiple times.	Eine Slave-Adresse ist mehrfach verwendet. Maßnahme: Eindeutige Slave-Adressen erzeugen.

4. Grafik Editor

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
Slave addresses must be unique in whole system! Ranges of slave addresses in project <Name> and project of interface <Name> are not unique. Please change your ranges of slave addresses (range of project:<Wert>-<Wert> range of interface:<Wert>-<Wert>) in one of the projects.	Die Grenzwerte für die Slave-Adressen im Projekt und beim Projekt des schreibenden Interface überschneiden sich. Slave-Adressen müssen im ganzen System eindeutig sein. Maßnahme: Anpassen der Grenzwerte in einem der Projekte und eine entsprechende Anpassung Slave-Adressen.
Connection IDs must be unique in whole system! Ranges of Connection IDs in project <Name> and project of interface <Name> are not unique. Please change your ranges of Connection ID's (range of project:<Wert>-<Wert> range of interface:<Wert>-<Wert>) in one of the projects.	Die Grenzwerte für die Connection IDs im Projekt und beim Projekt des schreibenden Interface überschneiden sich. Connection IDs müssen im ganzen System eindeutig sein. Maßnahme: Anpassen der Grenzwerte in einem der Projekte und eine entsprechende Anpassung der Connection IDs.
The module <Name> can be only used by a SafeCPU with master mode. Please change your configuration.	Ein Modul (z.B. XS-322-2INC) wird hinter einer Safety-CPU im Slave-Modus eingefügt. Maßnahme: Safety-CPU auf den Master-Modus umstellen.
More than one function block <Name> has been found, only one function block is allowed.	Ein Funktionsblock (z.B. Sf_HGW_Login) ist mehrfach verwendet, der Funktionsblock darf nur einmalig verwendet werden. Maßnahme: Überflüssige Funktionsblöcke entfernen.
The flag of safety interface communication of SafeCPU <Name> is wrong. Module <Name> is used in SafeCPU. Please change your configuration.	Die Safety Interface Kommunikation wurde deaktiviert, obwohl ein Modul (HBG, SIB) diese benötigt. Maßnahme: Flag umstellen oder Module löschen.
Macro Network <Name>, Object <Name>, Input <Name> is not connected	Der Eingang einer Instanz eines Funktionsblocks in einem Makronetzwerk ist nicht verbunden. Maßnahme: Verbindung zum Eingang der Instanz erstellen.
Macro Network <Name>, Object <Name>, Out-put <Name> is not connected	Der Ausgang einer Instanz eines Funktionsblocks in einem Makro-Netzwerk ist nicht verbunden. Maßnahme: Verbindung zum Ausgang der Instanz erstellen.
Address of memory variable or constant <Name> cannot connected with <Name> in SafeCPU <Name>	Die Adresse der Memory-Variablen bzw. der Konstanten kann in der Safety-CPU nicht korrekt verbunden werden.
Address of memory variable or constant <Name> cannot connected with element in output row SafeCPU <Name>	Die Adresse der Memory-Variablen bzw. der Konstanten kann in der Safety-CPU nicht korrekt verbunden werden.
Adjustable CANY or DIM inputs must have same data type	Veränderbare CANY-Variablen oder Dimensionen müssen den gleichen Datentypen haben.
Adjustable inputs exceeds range	Veränderbare Eingänge eines Funktionsblocks müssen eine Grenze haben.

4. Grafik Editor

4.21 Validierung und Generierung der Downloaddaten

Fehlermeldung	Beschreibung
Wrong number of inputs	Die Anzahl der veränderbaren Eingänge stimmt nicht.
Input <Name> in function block not found	Der Eingang kann im Funktionsblock nicht gefunden werden.
<Name> of <Name> must be connected with dint array	Das Element einer Tabelle muss korrekt mit dem Array verbunden werden.
Please check the length and dimensions of array input	Die Größe oder Dimension von Arrays stimmt nicht.
Data type is different from 'StartAddress' in <Name>	Der Datentyp ist unterschiedlich zum Datentyp der Startadresse.
Memory variable <Name> must not be connected with address in SafeCPU <Name>	Die Merker-Variable darf nicht mit einer Adresse der Safety-CPU verbunden werden.
Parameter(s) of Section <Name> are not in correct order	Die Parameter innerhalb einer Sektion sind in einer falschen Reihenfolge
Type array <Name> of memory variable <Name> not found	Der eingestellte Array-Datentyp der Memory-Variablen kann nicht gefunden werden.
Type array <Name> of constant variable <Name> not found	Der eingestellte Array-Datentyp der Konstanten kann nicht gefunden werden.

4. Grafik Editor

4.22 Berechnung der Monitoring-Settings

4.22 Berechnung der Monitoring-Settings

Der XN Safety Designer unterstützt den Anwender bei der Berechnung der maximal benötigten Übertragungszeiten für die Module sowie bei der Ermittlung der Zykluszeit für die Safety-CPU. Durch den Aufruf der Funktion **Monitored Settings** öffnet sich folgender Dialog mit den automatisch ermittelten und vom Anwender eingestellten Daten.

Monitoring settings

Bus Time of system (Range: 1 - 10 [ms]): 1.0

Safe CPU:

Safe CPU	Process Time (calculated) [ms]	Process Time (monitored) [ms]
XS_102_C00_000	1.6	3.0

Used Modules:

Master-Slave	Transmission Time (calculated) [ms]	Transmission Time (monitored) [ms]

Accept calculated values in dialog OK Cancel Help

Der XN Safety Designer kann mit Hilfe dieser Daten und des erstellten Projekts die Zykluszeit der Safety-CPU ermitteln. Das Projekt muss für diese Zwecke erstellt werden. Die Werte im Dialog können nur richtig berechnet werden, wenn das Projekt korrekt erstellt werden kann. Falls das Projekt nicht erstellt werden kann, wird keine berechnete Zykluszeit angezeigt.

Die Werte für die **Monitored** Daten können vom Anwender manuell im Dialog geändert werden (Selektion der Zellen und F2 öffnet den Editor).

Der Button **Accept calculated values in dialog** übernimmt die vorgeschlagenen Daten in die Spalten für die **Monitored** Werte.

Der Button **OK** übernimmt die **Monitored** Daten in das Projekt.



Der Anwender muss die vom XN Safety Designer vorgeschlagenen Werte im laufenden System auf Richtigkeit überprüfen!

4.23 Ressourcenverbrauch und Reaktionszeit

Der XN Safety Designer unterstützt den Anwender durch Berechnung des Ressourcenverbrauchs, einer Schätzung der maximalen Zykluszeit sowie Berechnung der maximalen (überwachten) Gesamtreaktionszeit vom sicheren Eingang bis zum sicheren Ausgang. Diese Information steht dem Anwender nach der Programmgenerierung (Kompilervorgang) zur Verfügung und muss vom Anwender auch bestätigt werden. Beim Onlinegehen erscheint dieser Dialog ebenfalls, allerdings wird dann die Gesamtreaktionszeit nicht automatisch angezeigt, sondern nur nach Drücken des Buttons „Calculate Maximum Controlled Time“.

Resources of Safe CPU

Safe CPU resources:

Safe CPU	Size of RAM (actual, maxim...	Size of Flash (actual, maxim...	Process Time (calculated, m...	Number of FSoE connections (ac
XS_102_C00_000	488 Byte, 24576 Byte	2035 Byte, 229376 Byte	1.2 ms, 1.2 ms	0, 170

Maximum controlled time:

Input	Output	Maximum controlled Time [ms]
XS_322_10DI_PF	XS_322_8DO_P20	4

Calculate Maximum Controlled Time OK

Es werden zusätzliche Informationen über das Kompilat ausgegeben.

Dies sind:

- Anzahl der COPY-Aufrufe
- Anzahl der CALL-Aufrufe bei den Funktionsblöcken
- Größe des Interpretercodes in Bytes
- Größe des Codes der Funktionsblöcke in Bytes
- Größe des belegten Bereichs der remanenten Variablen
- Größe der Zusatzinfo



Die Belegung des Speichers (Flash und RAM) hängt unter anderem von den oben angeführten Daten ab. Jeder Funktionsblock, jede unsichere Variable, jede I/O-Karte, ... belegt einen Teil des Speichers. Der Flash-Speicher wird zusätzlich durch Zusatzinformationen wie der Anzahl der Netzwerke, den Namen der Netzwerke, den Namen der Funktionsblöcke, dem Code der Funktionsblöcke, ... belegt.

4. Grafik Editor

4.23 Ressourcenverbrauch und Reaktionszeit

4.23.1 Ressourcenverbrauch

Folgende Ressourcen werden bei den Berechnungen betrachtet:

Flash-Verbrauch durch Programmkonfiguration

Hier wird der vom Programm benötigte Flash-Speicher dem auf der Safety-CPU vorhandenen Speicher gegenüber gestellt. Überschreitet das Programm im XN Safety Designer die verfügbare Größe, wird eine Fehlermeldung bei der Kompilierung generiert, ein Download ist in diesem Fall nicht möglich.

RAM-Verbrauch durch Programmkonfiguration

Hier wird der vom Programm benötigte RAM-Speicher dem auf der Safety-CPU vorhandenen Speicher gegenüber gestellt. Überschreitet das Programm im XN Safety Designer die verfügbare Größe, wird eine Fehlermeldung bei der Kompilierung generiert, ein Download wird somit verhindert.

Es werden einige Anhaltspunkte für eine grobe Berechnung der RAM-Belegung aufgelistet, hiermit kann eine Vorabberechnung getroffen werden, wie groß die Auslastung der Safety-CPU werden kann, bzw. ob eine ausgewählte Safety-CPU ausreichend ist. Für die Abschätzung der Belegung ist es nötig, dass der Anwender weiß, wie seine Applikation aussehen soll. Die Funktionsblöcke machen in vielen Applikationen den größten Teil der RAM-Belegung aus.

Nachfolgend werden die wichtigsten Größen aufgelistet, welche den RAM-Speicher belegen.

IO-Module

- Pro Eingang werden 48 Byte benötigt
- Pro SAFEDINT-Eingang werden 32 Byte benötigt
- Pro Ausgang werden 16 Byte benötigt
- Pro SAFEDINT-Ausgang werden 24 Byte benötigt
- ...

Variablen

- Pro Unsicherer Variablen werden 4 Byte benötigt
- ...

Funktionsblöcke (siehe Tabelle Speicherbelegung I/O-Module)

- Pro platzierten Funktionsblock werden 4 Byte benötigt
- Pro Eingang/Ausgang werden 4 Byte benötigt
- Pro SAFEDINT Eingang/Ausgang werden 8 Byte benötigt
- ...

Tab. 2: Speicherbelegung I/O-Module

Modul	RAM [Byte]
XS_322_10DI_PF	480
XS_322_8DIO_PF20	320
XS_322_4AI_02	512
XS_322_2ENC_INC	448
XS_322_2DO_RNOAC	32
XS_322_2DO_RNODC	32
XS_322_8DO_P20	128

4.23.2 Abschätzung der Zykluszeit des Anwenderprogramms

Je nach Komplexität und Größe des Anwenderprogramms kommt es zu einer höheren Zykluszeit bei der Abarbeitung des Anwenderprogramms in der Safety-CPU. Eine maximal errechnete Zykluszeit wird dem Anwender nach einem Kompiliervorgang auf Abruf zur Verfügung gestellt. Dabei wurde pro FB die maximale Durchlaufzeit bei der Entwicklung der FB ermittelt, diese Durchlaufzeiten der FB werden je nach Anwenderprogramm aufsummiert.

4.23.3 Gesamtreaktionszeit

Der Anwender stellt bei übertragenen Eingangs- und Ausgangssignalen eine maximale Übertragungszeit (maximales Alter) ein. Zusätzlich wird die maximale Programmlaufzeit (Zykluszeit) eingestellt, die von der Firmware überwacht wird. Mit Hilfe dieser eingestellten Werte kann eine maximale, überwachte Gesamtreaktionszeit von einem Modul (bzw. Eingang eines Moduls) zu einem Ausgang berechnet und dem Anwender dargestellt werden. Wird ein Ausgang von verschiedenen Module gesetzt, werden die Zeiten getrennt für jedes Modul angezeigt.



Der Anwender muss die vom XN Safety Designer vorgeschlagenen Werte im laufenden System auf Richtigkeit überprüfen (z.B. durch wiederholte Messungen der Reaktionszeiten im laufenden System)!

4.23.4 Länge der PDO-Frames

Der XN Safety Designer berechnet die Längen der PDO-Frames, welche von den Safety-Modulen erzeugt werden bzw. welche von den Safety-Modulen benötigt werden. Es wird eine Fehlermeldung generiert, wenn die Längen zu groß sind.

4.24 Debuggen

Um debuggen zu können, muss der Benutzer Online zur Steuerung sein und das Safety Projekt muss mit dem Programm auf den Safety-Modulen übereinstimmen. Die Anzeige von Werten kann ohne ein Login erfolgen. Sollen Werte geforct werden, muss man mind. im Debug-Modus angemeldet sein. Werte können nur im Temporären Operation Modus verändert werden.

Beim Debuggen werden die Ein-/ Ausgangszustände und Variablen angezeigt. Der Anwender kann die Werte von Konstanten und einzelnen Elementen der Eingangs- und Ausgangsspalte ändern (forcen) und zur Safety-CPU schicken.

TON	
TON0	
S_IN	S_Q
s_false	s_false
PT	ET
0	0

Die Elemente der Eingangs- bzw. Ausgangsspalte, sowie die temp. Variablen und unsichere Variablen (nur BOOL) werden farblich hinterlegt. Die Verbindungen, welche von Elementen bzw. von BOOL-Elementen weggehen, werden farblich dargestellt.

Farbliche Hinterlegung Eingangs-/Ausgangsspalte, farbliche Darstellung der Verbindungen:

- Safebool TRUE/TRUE ...grün
- Safebool FALSE/FALSE ...blau
- Safebool Error ... rot
- Bool TRUE ...grün
- Bool FALSE ... blau

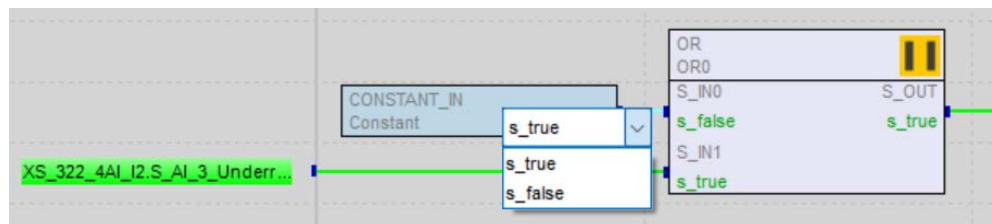
4.25 Forcen

Der Anwender kann die Werte von Konstanten, sicheren Eingängen (Eingangsspalte) und sicheren Ausgängen (Ausgangsspalte) ändern und zur Safety-CPU schicken. Um die Werte auf der Safety-CPU ändern zu können, muss mindestens das Passwort-Level **Debug** angemeldet sein. Außerdem muss die Safety-CPU im temporären Operation Modus laufen. Um Werte von Modulen im Slave-Modus forcen zu können, muss beim Modul im Master-Modus mindestens im Debug Level angemeldet sein.

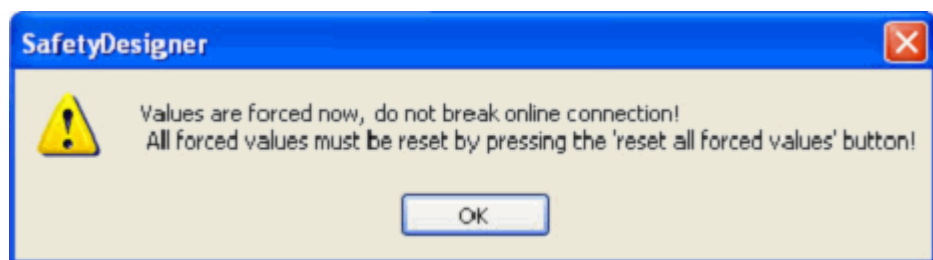
Bei den Konstanten kann mittels der **F2**-Taste bzw. mittels eines verzögerten Klicks ein Editor für die Eingabe des Wertes geöffnet werden, bei Bestätigung des Wertes wird dieser zur Safety-CPU gesendet und dort gesetzt.

Bei den sicheren Ein- und Ausgängen können die Werte mittels Kontextmenü geändert werden.

Geforcete Werte werden unterstrichen dargestellt.



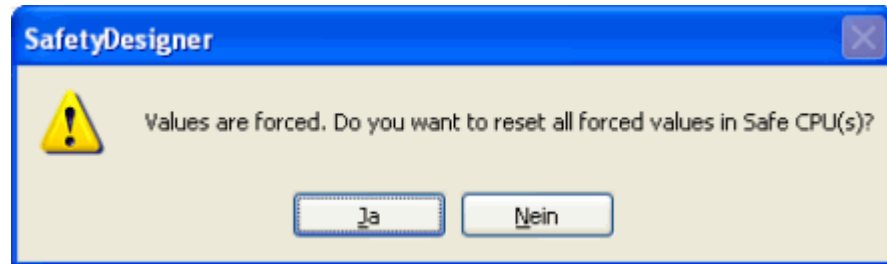
Der Anwender erhält einen Warnhinweis, dass die Werte manuell verändert wurden. Der Anwender muss diesen Hinweis bestätigen und wird außerdem darauf hingewiesen, dass die Online-Verbindung während des Forcens nicht getrennt werden darf.



Die Safety-CPU läuft nun mit geforcen Werten. Die Buttons zum Offline Gehen bzw. Schließen des Online State Views sind deaktiviert. Der Anwender hat jederzeit die Möglichkeit durch einen Rechtsklick auf einen leeren Bereich im Netzwerkeditor oder durch das Drücken des **Reset all forced values** Buttons, sämtliche geforceten Werte zurückzusetzen. Drückt der Anwender die Buttons **Login** oder **Service Mode** im Online State View, muss bestätigt werden, dass die Werte zurückgesetzt werden.

4. Grafik Editor

4.25 Forcen



Die Befehle zum Ändern des Logins oder zum Setzen des Service Modes werden nur ausgeführt, wenn das Rücksetzen der Werte explizit bestätigt wurde.

Wird die Verbindung zum XN Safety Designer unterbrochen und wurden die geforceten Werte nicht explizit durch den Anwender zurückgesetzt, geht die Safety-CPU in einen Fehlerzustand.

- ➔ Wenn die Werte geändert werden, so MUSS die Maschine im Blickfeld des Technikers sein, der die Werte ändert!
- ➔ Wenn die Werte geändert werden, darf die Online-Verbindung nicht getrennt werden!

4.26 Watch-Fenster

Im Watch-Fenster können im Online-Betrieb die Werte von Elementen im Netzwerk angezeigt werden. Die Anzeige von Werten kann ohne ein Login erfolgen.

Der Anwender kann im Online-Betrieb verschiedene Elemente aus Netzwerken oder aus den Trees mittels Drag & Drop in das Watch-Fenster ziehen. Es können Elemente der Eingangs- und Ausgangsspalte, sowie Instanzen von Variablen, Konstanten, Interfaces und ganze Funktionsblöcke in das Watch Fenster eingefügt werden. Werden Elemente aus einem Tree in das Watch Fenster gezogen, so wie z.B. ein Eingangskartenmodul, so wird ein Eintrag dieses Moduls generiert mit allen davon verwendeten Eingängen.

Die Elemente werden im Watch Fenster in der Reihenfolge des Einfügens angezeigt. Die Zeilen können vom Anwender durch einen Klick auf den Spaltenheader ‚Name‘ sortiert werden (alphabetisch nach Safety-CPU und Netzwerken). Die Daten des Watch Fensters werden beim Schließen des Projekt gespeichert und stehen beim nächsten Laden des Projekts wieder zur Verfügung.

Die Bezeichnung des Elements wird in der Spalte ‚Name‘ dargestellt (Zusammensetzung: Name der Safety-CPU, Name des Netzwerks, Name des Elements). Die unsicheren Variablen werden mit dem Symbolnamen (Name aus dem Tree) dargestellt. Die Online-Werte werden in der Spalte ‚Value‘ dargestellt. Der Datentyp des Elements wird in der Spalte ‚Type‘ angezeigt.

Watch		
Name	Value	Type
XS_102_C00_000NetworkFastInput(XS_102_C00_000) Bit1		BOOL
XS_102_C00_000NetworkXS_322_2DO_RNOAC Safe_Output1		SAFEBOOL
XS_102_C00_000NetworkBool_to_Safebool		BOOL
IN		BOOL
S_OUT		SAFEBOOL

Output Watch Search

Elemente können im Watch Fenster geforced werden. Alle nicht forcebaren Elemente sind grau hinterlegt dargestellt. Geforcete Werte werden in brauner Schrift dargestellt. Sollen Werte geforced werden, muss man mindestens im Debug-Modus angemeldet sein. Werte können nur im Temporären Operation Modus verändert werden.

Um Werte von Modulen im Slave-Modus forcen zu können, muss man beim Modul im Master-Modus mind. im Level Debug angemeldet sein. BOOL- und SAFEBOOL-Werte können über eine Combobox geändert werden, DINT- und SAFEDINT-Werte müssen über einen Editor eingegeben werden.

4. Grafik Editor

4.26 Watch-Fenster

Watch			
Name	Value	Type	
XS_102_C00_000/Network/XS_322_4AI_12.S_AI_1	5996	SAFEDINT	
XS_102_C00_000/Network/XS_322_2DO_RNOAC		SAFEBOOL	
Safe_Output1	s_true	SAFEBOOL	
Safe_Output2	s_true	SAFEBOOL	
XS_102_C00_000/Network/XS_322_4AI_12		SAFEBOOL	
S_AI_1	5996	SAFEDINT	
S_AI_1_Overage	s_true	SAFEBOOL	
S_AI_1_Underange	s_true	SAFEBOOL	
Insert new via Drag&Drop or context menu (only in Online Mode)			
	s_false		

Output Watch Search

Für die geforcten Werte gelten die selben Hinweise wie in → Abschnitt "Forcen", Seite 102 beschrieben.

Die Und-Verknüpfung der Eingangsspalte mittels Äquivalenz-Funktionsblock wird über einen Baum angezeigt. Es werden dazu die einzelnen Information des Funktionsblocks aufgelistet. Die Werte können nicht geforced werden.

Watch			
Name	Value	Type	
XS_102_C00_000/Network/XS_322_4AI_12.S_AI_2_Underange & S_AI_2_Overage (function block "SF_Equivalent")		SAFEBOOL	
Result	s_false	SAFEBOOL	
First Channel	s_true	SAFEBOOL	
Second Channel	s_false	SAFEBOOL	
Discrepancy Time	5	DINT	
Error	true	BOOL	
Diagnose code for error (details in help of SF_Equivalent)	0xc001	HDINT	
XS_102_C00_000/Network/XS_322_2DO_RNOAC Safe_Output1	s_false	SAFEBOOL	

Output Watch Search

Result: Ergebnis der Verknüpfung

First Channel: Wert des ersten sicheren Eingangs

Second Channel: Wert des zweiten sicheren Eingangs

Discrepancy Time: Eingestellte Zeitdifferenz

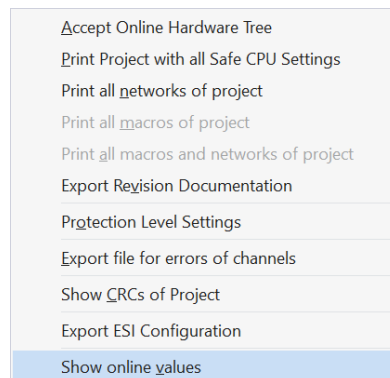
Error: Flag, ob der Funktionsblock im Fehlerzustand ist

Diagnose code for error: Diagnosecode für den Fehlerzustand (die Bedeutung des Codes kann beim SF-Äquivalenzblock nachgelesen werden)

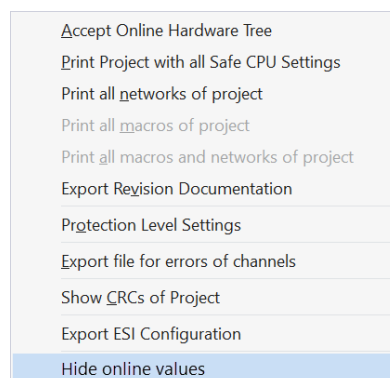
4.27 Anzeige der Online-Werte von Modulen im Hardware Tree

4.27 Anzeige der Online-Werte von Modulen im Hardware Tree

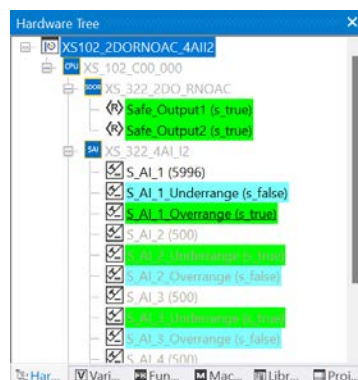
Die Anzeige von Online-Werten im Hardware Tree ist nur dann verfügbar, wenn sie aktiviert wurde. Dafür gibt es im Kontextmenü des Projekts den Punkt **Show online values**.



Um die Geschwindigkeit der Online-Anzeigen in Netzwerk und Watch-Fenster nicht unnötig zu verlangsamen, kann sie mit **Hide online values** auch wieder abgeschaltet werden. Die letzte Einstellung dazu wird im Projekt gespeichert.



Bei SafeBool Ein- und Ausgängen wird der Zustand des Ein- bzw. Ausgangs (s_true, s_false oder s_error) in Klammern nach dem Namen angezeigt, zusätzlich wird der Name farblich hinterlegt: grün für s_true, blau für s_false und rot für s_error.



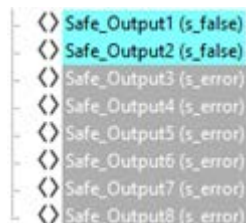
4. Grafik Editor

4.27 Anzeige der Online-Werte von Modulen im Hardware Tree

Bei SafeDint Ein- und Ausgängen wird der Wert nach dem Namen in Klammern angezeigt und nur im Fall eines Errors rot hinterlegt.



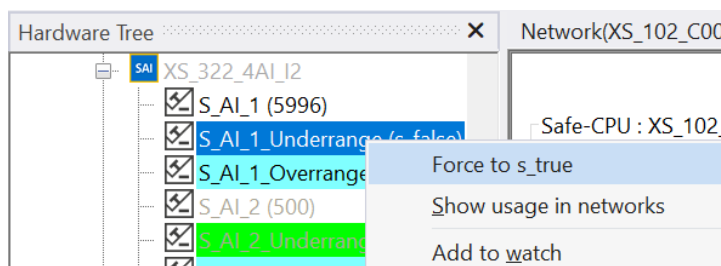
Alle nicht in Netzwerken verwendete Ausgänge sind mit „s_error“ belegt, diese werden allerdings nicht rot, sondern grau hinterlegt.



Werte werden für die Safety-CPU angezeigt, mit der sie auf der Steuerung verbunden sind.

Forcen von SafeBool-Werten im Hardware Tree

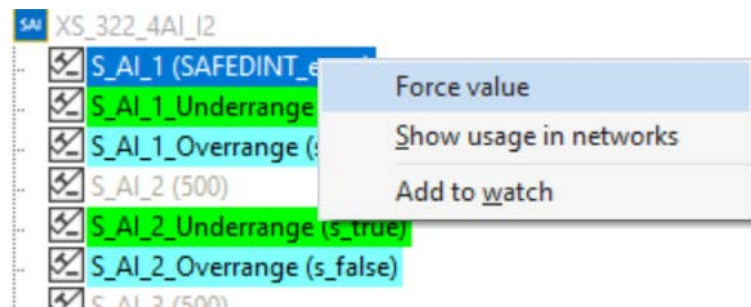
SafeBool-Werte können im Hardware Tree über das Kontextmenü verändert werden, ähnlich wie in den Ein-/Ausgangsspalten der Netzwerke. Im Falle eines Fehlers (s_error) kann sowohl zu s_true als auch zu s_false geändert werden.



Forcen von SafeDint-Werten im Hardware Tree

SafeDint-Werte können im Hardware Tree ebenfalls über das Kontextmenü verändert werden.

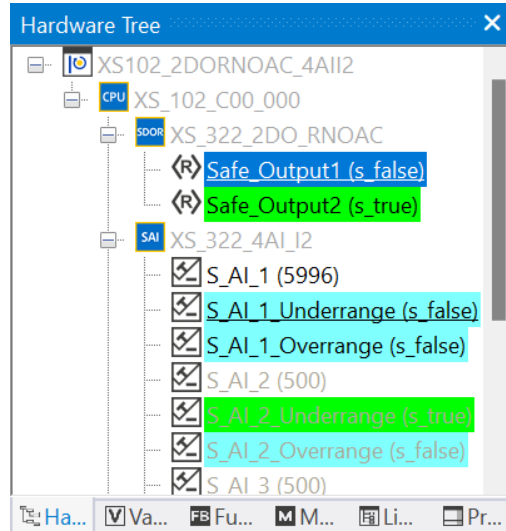
Dabei öffnet sich ein kleines Eingabefenster, in dem der Wert editiert werden kann.



4. Grafik Editor

4.27 Anzeige der Online-Werte von Modulen im Hardware Tree

Ein- und Ausgänge, deren Werte geforced wurden, werden ebenso wie in Ein- und Ausgangsspalten mit unterstrichenem Text angezeigt.



Werte können nur geforced werden, wenn der Ein-/Ausgang in einem Netzwerk der CPU verwendet wird, mit der das Modul verbunden ist.

4. Grafik Editor

4.28 Log-Dateien von Safety-Modulen auslesen und anzeigen

4.28 Log-Dateien von Safety-Modulen auslesen und anzeigen

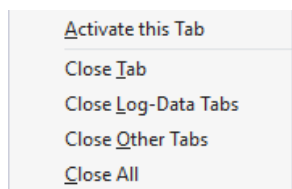
Über das Kontextmenü im Hardware Tree (jeweiliges Modul) werden die Daten des Log-Speichers eines Sicherheitsmoduls (z.B. Safety-CPU) ausgelesen und in zwei Binärdateien (je eine Datei für Controller 1 und Controller 2) gespeichert. Die Informationen werden im XN Safety Designer aufbereitet, entsprechend abgespeichert und angezeigt. Enthalten die Controller 1 und Controller 2 unterschiedliche Informationen, werden diese Einträge in getrennten Zeilen untereinander angezeigt, rot eingefärbt und die zweite Zeile mit dem Vermerk „Controller 2“ versehen. In den ersten und letzten Zeilen der Anzeige findet man grau hinterlegt Angaben zum Modulnamen, der Firmwareversion und gegebenenfalls zum Projekt, welches in der Safety-CPU geladen ist.

Log-Einträge an unterschiedlichen Tagen werden mit einer gelb eingefärbten Zeile, mit dem jeweiligen Datum zwischen den Einträgen angezeigt.

Einträge über einen (Neu-) Start des Systems sind violett hinterlegt.

Time	Message (press here to go to the latest message)
	Startup of system
3574257.720	Error 1150,0x00000000,0x00000008 - Error(1150): The input value hasn't been updated for an unacceptable amount of time. Involved input: 0x00000000 (Bit 0 = 1.inp...
3574257.774	Error 1043,0x0000015D,0x000000C4 - Error(1043): A local module is in error mode. Error of local module: Error 1093 at module XS_322_4AI_I2 (child module at positio...
Controller 2	Error 1196,0x27250104,0x03490357 - Error(10021): The 24V1 supply voltage is under the defined limits. Error identifier 0x00000104 (Byte 0 is index of ADC channel) in...
	Startup of system
3576161.608	Set Service-Mode
3576205.177	Change state of Configuration from 'configured and verified' to 'invalid'
3576205.177	Change state of Configuration from 'configured and verified' to 'not configured'
3576227.982	Change state of Configuration from 'not configured' to 'configured but not deployed and not verified'
3576233.397	Start Application
3576233.486	Change state of Configuration from 'configured but not deployed and not verified' to 'configured and deployed but not verified'
3576241.510	Set Service-Mode
3576248.015	Change state of Configuration from 'configured and deployed but not verified' to 'configured and verified'
3576254.827	Start Application
3576781.613	Set Service-Mode
3576894.320	Start Application
3576987.333	Set Service-Mode
3576991.263	Start Application
3577014.234	Set Service-Mode
3577280.408	Start Application
3577347.056	Set Service-Mode
3577404.928	Start Application

Um die Log-Daten Anzeige(n) wieder zu schließen, enthält das Kontextmenu der Register-Reiter den Punkt **Close Log-Data Tabs**.



Einmal übertragene Log-Daten können jederzeit über das Menu **File** und den Punkt **Log-Data** mit dem Namen wieder angezeigt werden.

4. Grafik Editor

4.28 Log-Dateien von Safety-Modulen auslesen und anzeigen

Log-Data	Open Logfile...
Print... Ctrl+P	Delete All Stored Logfiles
Print Setup...	000

Der Menüpunkt **Delete All Stored Logfiles** löscht alle zu dem geöffneten Projekt gehörigen Log-Daten.

Über **Open Logfile...** können auch Log-Daten eingesehen werden die zu anderen Projekten gehören. Nach dem Aufruf erscheint ein Dialog **Open Log-Data File**.

Projects > XS102_2DORNOAC_4AII2_V2		Search XS102_2DORNOAC_...	
Name	Date modified	Type	Size
ProjectInternal	27.01.2025 14:35	File folder	
XS102_2DORNOAC_4AII2_V2_XS_102_C00_000_L...	27.01.2025 13:35	xmlfile	8 KB

Projects > XS102_2DORNOAC_4AII2_V2 >		Search XS102_	
Name	Date modified	Type	
ProjectInternal	27.01.2025 14:35	File folder	
XS102_2DORNOAC_4AII2_V2_XS_102_C00_000_1.bin	27.01.2025 13:35	BIN File	
XS102_2DORNOAC_4AII2_V2_XS_102_C00_000_2.bin	27.01.2025 13:35	BIN File	

Die Dateien **<..._log.xml>** enthalten Informationen zum Auswerten der Daten. Sollten diese nicht zur Verfügung stehen und nur die Roh-Daten aus dem Speicher in Form von BIN-Dateien vorhanden sein, dann muss man in der Combobox anstatt „Log-Data info(*_log.xml)“ den Filter „Log-Data file(*.bin)“ auswählen. Dabei enthalten die Dateien „..._1.bin“ Daten von Controller 1 und „..._2.bin“ von Controller 2. Sind bei Dateien vorhanden und man öffnet die „..._1.bin“-Datei, wird automatisch auch die andere geöffnet und beide Dateien verglichen.

Bedeutung der Log-Daten Einträge

„Change State of Login from ‘old’ to ‘new’“

Anmeldung bzw. Abmeldung für Service und Datenübertragung zu einer Safety-CPU.

Mögliche Verbindungsarten:

not logged in: keine Verbindung

debug: für Fehlersuche

configure: um Konfiguration zu ändern und Projekte zu übertragen

general: Firmware Download (nur für berechtigte Personen)

4. Grafik Editor

4.28 Log-Dateien von Safety-Modulen auslesen und anzeigen

„Error Login from ‘old’ to ‘new’, returned message code: Errornumber”

Anmeldung war nicht erfolgreich.

Mögliche Ursachen: keine Berechtigung oder Tippfehler.

Mögliche Verbindungsarten: siehe „Change state of Login ...”

„Password changed for ‘level’”

Passwort für Verbindungsart ‘level’ wurde geändert.

Mögliche Verbindungsarten: siehe „Change state of Login ...”

„Password change failed for ‘level’ returned message code: Errornumber”

Eine Änderung des Passworts wurde abgelehnt.

Mögliche Ursachen: keine Berechtigung oder Tippfehler.

Mögliche Verbindungsarten: siehe „Change state of Login ...”

„Password loaded from SD-card”

Das Passwort wurde von einer Speicherkarte bezogen.

„Set Service-Mode”

Safety-CPU wurde angehalten, um Veränderungen vorzunehmen, z.B. ein anderes Projekt zu laden oder das Projekt zu verifizieren.

„Set temporary Service-Mode”

Safety-CPU wurde in eine vorübergehenden Service-Mode versetzt.

„Start Application”

Projekt in der Safety-CPU wurde gestartet.

„Change state of Configuration from ‘old’ to ‘new’”

Safety-CPU hat die Betriebsart von „old” auf „new” geändert.

Mögliche Betriebsarten:

unknown: Defaultwert

invalid: das geladene Projekt wurde verworfen und wird nicht mehr abgearbeitet

not configured: kein Projekt ist geladen

configured but not deployed and not verified: es ist ein Projekt geladen, aber es wurde noch nicht an die Slave-Module übertragen und noch nicht verifiziert

configured and deployed but not verified: Projekt ist geladen und verteilt aber noch nicht verifiziert

configured and verified: Projekt ist geladen und verifiziert.

„Configuration loaded from SD-card”

Konfiguration und Projektdaten wurden von einer Speicherkarte geladen.

„Got safety number from remote module ‘XYZ’: 0x000000yy”

4. Grafik Editor

4.28 Log-Dateien von Safety-Modulen auslesen und anzeigen

„Got safety number from remote module with index 123: 0x000000yy“

Sicherheitsnummer wurde vom Modul 'XYZ' bzw. Modul mit Index 123 abgefragt. Ergebnis: 0x000000yy.

„Module 'XYZ' changed“

„Module with index 123 changed“

Ein Modulwechsel von 'XYZ' bzw. Modul mit Index 123 wurde festgestellt

„Deployed configuration to module ,XYZ' with safety number 0x000000yy“

„Deployed configuration to module with index 123 with safety number 0x000000yy“

Das entfernte Modul mit Namen 'XYZ' bzw. Index 123 und Sicherheitsnummer 0x000000yy wurde konfiguriert.

„Error 123, ...“

Allgemeine Fehlermeldung mit einem erklärenden Text.

„Quit Error 123“

Fehler mit der Nummer 123 wurde zurückgesetzt.

„Too many messages, no more messages added“

Mindestens ein Log-Eintrag konnte nicht eingefügt werden.

„Date and Time: 12.01.2013 12:13:51 overrun of internal timestamp: 2“ (gelb hinterlegt)

Datum und Uhrzeit. Die Information über den internen Zeitüberlauf-Zähler ist nur für den Techniker interessant.

„Date: 12.01.2013“ (gelb hinterlegt)

„New Date: 12.01.2013“ (gelb hinterlegt)

Angabe über das Datum der folgenden Einträge. Darunter stehen dann am linken Rand die Zeitangaben für die Meldungen, die am gleichen Tag eingetragen wurden. Bei jedem Datumswechsel und nach jedem Systemneustart wird diese Nachricht eingefügt.

„Time passed: 13 days, 2 hours, 32 minutes, 12 seconds“ (gelb hinterlegt)

Sollte eine Auflösung der Uhrzeit nicht möglich sein, wird im Fall, dass der Sekundenzähler am linken Fensterrand wieder bei Null begonnen hat, angegeben wieviele Tage, Stunden, Minuten und Sekunden zwischen der oberen und der unteren Meldung vergangen sind.

„Startup of system“

„Startup of system at: 12.01.2013 12:13:51“

Die Safety-CPU wurde (neu) gestartet. Datum und Uhrzeit werden angegeben, falls sie ermittelt werden können.

„Shut down of system“

4. Grafik Editor

4.28 Log-Dateien von Safety-Modulen auslesen und anzeigen

Die Safety-CPU wurde ausgeschaltet oder die Stromzufuhr wurde unterbrochen.

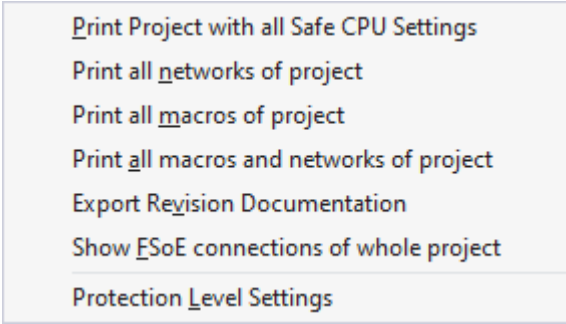
„Missing optional module(s): Safety-CPU1, Safety-CPU2, #15“

„Missing optional module(s) with indexes: 1, 5, 17“

Die optionalen Module mit Namen Safety-CPU1,... bzw. mit Index 15,... wurden nicht gefunden.

4.29 Drucken

Beim Ausdruck können das Projekt oder die einzelnen Netzwerke ausgedruckt werden. Die Darstellung des Ausdrucks der Netzwerke entspricht der grafischen Anzeige im Editor. Beim Projekt können der Hardware Tree, die Variablen, die Projekteinstellungen, die Safety-CPU- und Safety-Modul Einstellungen ausgedruckt werden. Der Ausdruck wird über das Kontextmenü im Projekt Tree oder im Hardware Tree ausgeführt. Das aktuell sichtbare Netzwerk kann auch direkt über das File-Menü ausgedruckt werden. Zusätzlich besteht die Möglichkeit alle Netzwerk mit einem Befehlsaufruf auszudrucken. Der Ausdruck aller Netzwerke wird über das Kontextmenü im Projekt Tree oder im Hardware Tree ausgeführt.



Print Project with all Safe CPU Settings
Print all networks of project
Print all macros of project
Print all macros and networks of project
Export Revision Documentation
Show ESoE connections of whole project
Protection Level Settings

Ein Ausdruck des gesamten Projekts mit allen Netzwerken kann über den Druckbefehl im File-Menü ausgeführt werden; es werden zuerst die Projekteinstellungen und dann die Netzwerke gedruckt.

Wird der Befehl zum Drucken aller Netzwerke aufgerufen, werden alle Netzwerke gleichzeitig im Vorschaudioalog angezeigt. Die Zoom-Stufe berechnet sich nach dem größten Netzwerk.

Es wird auf allen gedruckten Seiten eine Kopfzeile und Fußzeile mit Einstellungen des Projekts eingefügt:

Kopfzeile enthält:

- Projektnamen
- Firma des Projekts
- Autor des Projekts
- Safety-System mit Version des XN Safety Designers
- Revision des Projekts

Fußzeile enthält:

- Datum und Uhrzeit des Drucks
- Seite mit Gesamtseitenzahl

Bei den Safety-CPUs werden beim Drucken des Projekts die CRCs der jeweiligen Konfiguration angezeigt. Die CRCs können nur angezeigt werden, wenn das Projekt

4. Grafik Editor

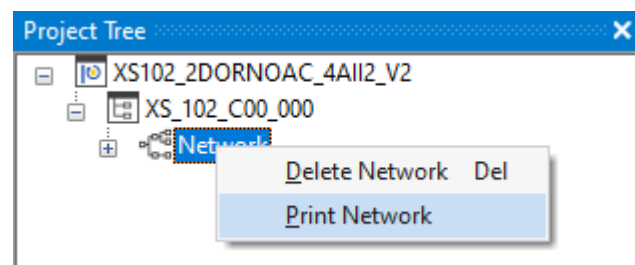
4.29 Drucken

gebildet erstellt ist. Die Anzeige der CRCs erfolgt im Hexadezimal-Format; ist keine CRC vorhanden wird der Text „no CRC“ angezeigt.

Im Online-Betrieb werden beim Drucken des Projekts zusätzliche Daten der Module gedruckt.

Online-Daten enthalten:

- Modulnamen
- Sicherheitsnummer des Moduls
- Firmwareversion des Moduls
- Konfigurationsstatus
- Projektname und Revision des Projekts
- Bei optionalen Modulen wird angezeigt, ob das Modul vorhanden ist



Im Druckvorschau-Dialog gibt es jeweils zwei Karteireiter.

Beim Ersten kann der Drucker ausgewählt und allgemeine Druckeinstellungen gesetzt werden.

Der zweite Reiter ist abhängig vom auszudruckenden Element.

4.29.1 Einstellungen beim Netzwerk drucken

Reiter „Network“:

Zoomfaktor: Vergrößerungs- bzw. Verkleinerungsfaktor kann in Prozent festgelegt werden.

Landscape = Querformat

Autosize of columns = Spaltenbreite fürs Drucken automatisch anpassen

Navigation durch die einzelnen Seiten

Mittels Kontextmenü kann ausgewählt werden:

- Print this Page (nur diese Seite drucken)
- Select this Page (diese Seite auswählen)
- Select All (alles auswählen)
- Deselect All (komplette Auswahl aufheben)

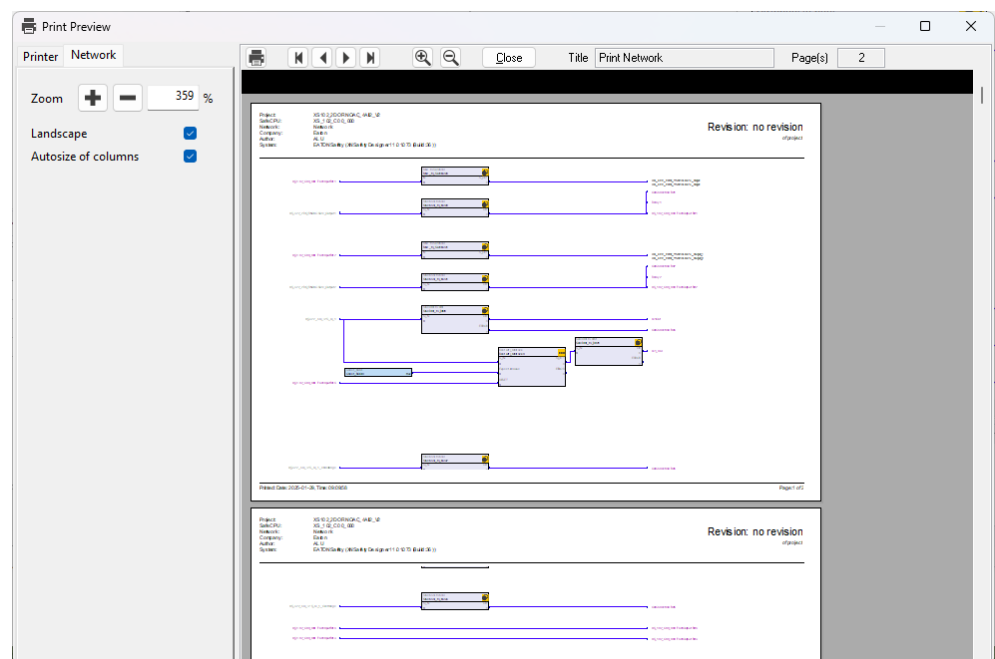


Abb. 1: Beispiel für einen Netzwerk Ausdruck

4. Grafik Editor

4.29 Drucken

4.29.2 Einstellungen beim Project Tree drucken

Reiter „Project“:

Hier kann man die einzelnen Module mitsamt Hardware Tree auswählen, welche ausgedruckt werden sollen.

Auswählbar sind:

- Safety-CPU
- Safe Modules
- Hardware Tree
- Unsafe
- Temp. Variables
- Revision documentation
- Memory-Variablen
- Konstanten
- Makros
- Parameterlisten

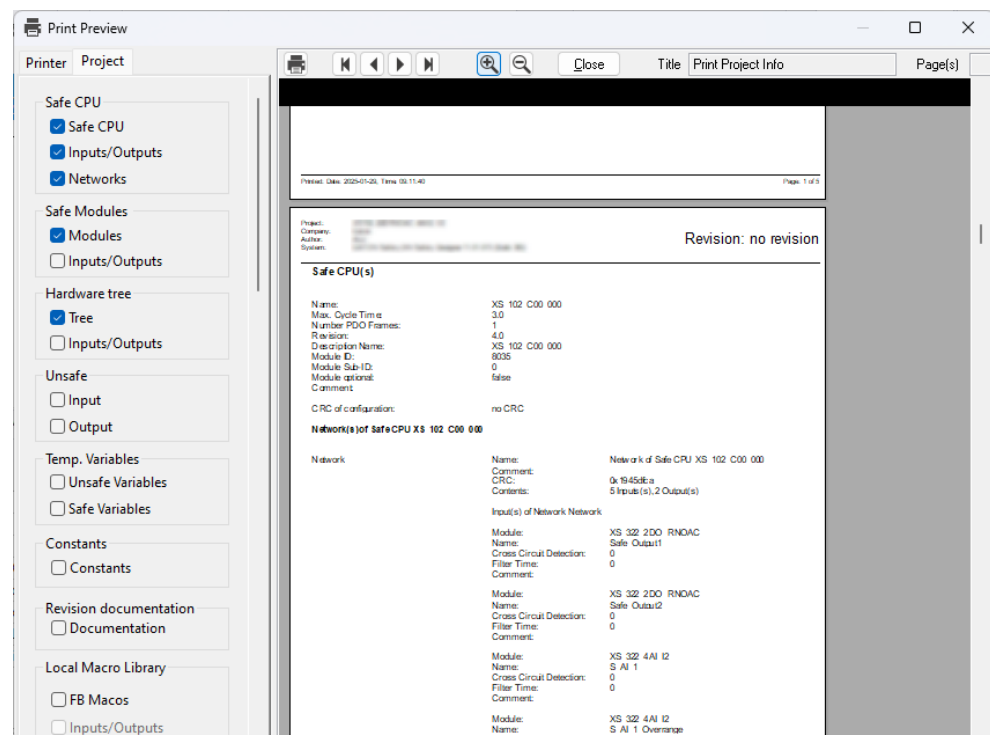
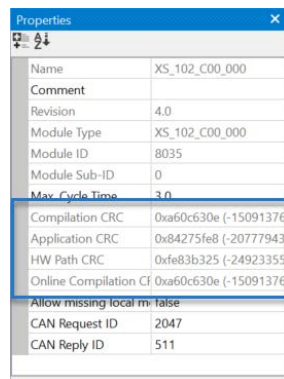


Abb. 2: Beispiel für Project Tree drucken

4.30 CRC-Vergleichsdialog

Für eine Safety-CPU werden 3 CRCs berechnet. Eine CRC repräsentiert den Hardware-Aufbau, eine die Applikation, und eine das komplette Kompilat. Die CRCs werden erst nach einem Build des Projekts berechnet. Das Projekt wird nach dem Build nochmals gespeichert, wenn sich die Werte der CRCs geändert haben. Die CRCs werden im Property Browser der Safety-CPU angezeigt. Die Online-Werte werden ebenfalls angezeigt.



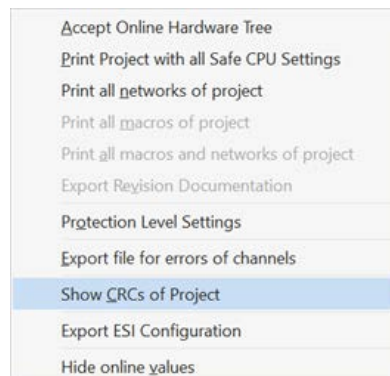
Name	XS_102_C00_000
Comment	
Revision	4.0
Module Type	XS_102_C00_000
Module ID	8035
Module Sub-ID	0
Max. Cycle Time	3.0
Compilation CRC	0xa60c630e (-15091376)
Application CRC	0xb4275fe8 (-20777943)
HW Path CRC	0xfe83b325 (-24923355)
Online Compilation CRC	0xa60c630e (-15091376)
Allow missing local m...	false
CAN Request ID	2047
CAN Reply ID	511



Es muss beachtet werden, dass sich durch das Verschieben von Modulen die Übertragungszeit am Hardware-Baum ändern kann (z.B. Verschieben hinter ein CIV 512).

Die Reaktionszeit einzelner Ausgänge kann sich dadurch ändern. Die CRC der Applikation ändert sich nicht, die CRC des Hardware-Baums wird angepasst. Es muss hier eine neue Betrachtung der Zeiten der Maschine durchgeführt werden.

Es gibt es die Möglichkeit die CRCs zu exportieren und auch zu vergleichen. Wenn das Projekt erfolgreich kompiliert ist, kann man über das Kontextmenü im HW-Tree oder Project Tree den CRC Dialog aufrufen.



Accept Online Hardware Tree
Print Project with all Safe CPU Settings
Print all networks of project
Print all macros of project
Print all macros and networks of project
Export Revision Documentation
Protection Level Settings
Export file for errors of channels
Show CRCs of Project
Export ESI Configuration
Hide online values

4. Grafik Editor

4.30 CRC-Vergleichsdialog

The screenshot shows the 'CRC Dialog' window. It contains a table with the following data:

Name	CRC_SCP	CRC_APPL	CRC_HW
XS_102_C00_000	0XA60C630E	0X84275FE8	0XFE83B325

At the bottom of the dialog are three buttons: 'Export', 'Compare', and 'OK'.

Der Dialog zeigt nun für jede Safety-CPU die berechneten CRCs an. Über den Button Export lässt sich eine XML-Datei exportieren, welche die Informationen beinhaltet. Hier stehen die 3 CRCs für jede Safety-CPU, sowie das Erstellungsdatum in der Datei. Die Datei wird über eine eigene CRC geschützt. Über den Button Compare lässt sich eine exportierte XML-Datei importieren und wird automatisch mit den aktuellen CRCs verglichen.

The screenshot shows the 'CRC Dialog' window after a comparison. The table now displays two CRC values for each component, separated by a slash. The first CRC value for each component is highlighted in red, indicating a difference.

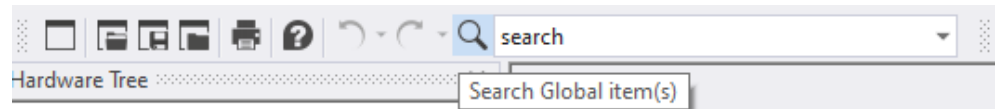
Name	CRC_SCP	CRC_APPL	CRC_HW
XS_102_C00_000	0XS42CE948 / 0XA60C630E	0X84275FE8 / 0X84275FE8	0XFE83B325 / 0XFE83B325

At the bottom of the dialog are three buttons: 'Export', 'Compare', and 'OK'.

Rote Bereiche zeigen einen Unterschied der CRCs an. Gelbe Bereiche sind nur in der importierten Datei vorhanden. Weiße Bereiche mit einer CRC sind nur in der aktuellen Konfiguration enthalten. Weiße Bereiche mit 2 CRCs haben identische CRCs in Konfiguration und importierter Datei.

4.31 Globale Suche

Im XN Safety Designer gibt es eine globale Suchfunktion.



Beim Editfeld handelt es sich um eine Schnellsuche, welche die zuletzt verwendete Einstellung der Suche verwendet. Die Auswahl wird mit Enter bestätigt, das Ergebnis wird im Search Window angezeigt.

Name	Found in	Type
SearchSCP111.SearchNetwork.WritelInterfaceSearch:WriteVarSearch1	Network	Instance
SearchSCP111.SearchNetwork.WritelInterfaceSearch:WriteVarSearch	Network	Instance
SearchSCP111.SearchNetwork.SearchmemoVar	Network	Instance
SearchSCP111.SearchNetwork.SearchmemoVar1	Network	Instance
SearchSCP111.SearchNetwork.SearchConst_SAFEBOOL0	Network	Instance
SearchSCP111.SearchNetwork.SearchMacro0	Network	Instance
SearchSCP111.SearchNetwork.SearchMacro0.SearchMacro.In0	Network	Input
SearchSCP111.SearchNetwork.SearchMacro0.SearchMacro.In1	Network	Input

Durch das Drücken der **Enter**-Taste bzw. einen Doppelklick werden die Elemente angesprungen.

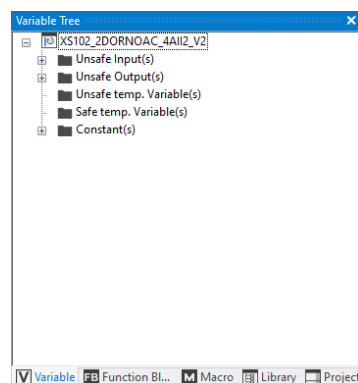
Über den Button in der Toolbar kann eine genauere Suche definiert werden. Hier kann unterschieden werden, in welchen Bereichen gesucht werden soll.

The screenshot shows the 'Global Search' dialog box. It has a title bar with a search icon and a close button. Inside, there is a 'Search Text:' field with the text 'search'. Below it are three checkboxes: 'Match Case', 'Whole Word', and 'Comments in:'. The 'Comments in:' section has two sub-sections: 'Tree' and 'Networks', both with checked checkboxes. To the right of the 'Search Text:' field is a 'Search in:' section with three checkboxes: 'Tree', 'Network', and 'Macros and Libraries', all of which are checked. At the bottom right is a 'Search' button.

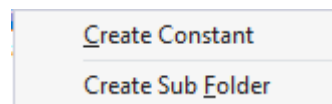
5. Konstanten

5. Konstanten

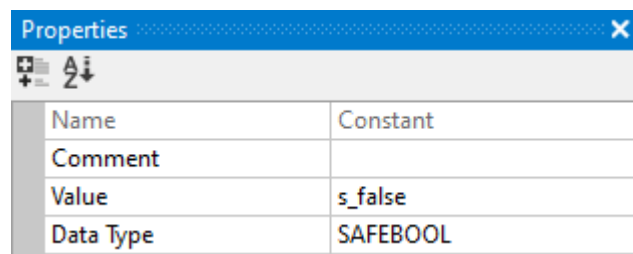
Bei Konstanten handelt es sich um Elemente mit einem vorgegebenen Wert. Der Wert kann vom Anwender eingestellt werden. Die Konstanten können mehrfach in Netzwerken platziert werden.



Die Konstante werden im Variable Tree über das Kontextmenü am Ordner Constant (s) angelegt.



Jede Konstante hat einen Datentyp und einen Wert. Die Daten können im Property Browser der Konstanten eingegeben werden.



Die Konstante können mittels Drag & Drop in ein Netzwerk gezogen werden.



Der Unterschied zu Funktionsblock-Konstanten besteht darin, dass

1. Konstanten mehrfach platziert werden können.
2. alle Instanzen einer Konstante denselben Wert haben.
3. Konstanten nicht in Function Block Macros verwendet werden können.
4. Konstanten in Parameterlisten verschoben werden können und umgekehrt.

6. Versionskontrolle und Archivierung

6.1 Versionierung

Der Anwender wird bei jedem Speichern des Projekts aufgefordert einen neuen Eintrag in der Revisionshistorie anzulegen, d.h. eine Versionsnummer zu vergeben und eine Beschreibung für die Version zu erstellen. Das Eintragen dieser Daten ist nicht zwingend und bleibt dem Anwender überlassen. Die Einträge können über den Button **Export** in eine XML-Datei gespeichert werden.

Revision Documentation of XS102_2DORNOAC_4All2_V2

Owner
Company: Eaton Author: Author

Revision	Date	Company	Author	Description
0.0	2025-01-29	Eaton	Author	First version

Insert Export OK Cancel

6.2 Archivierung

Der Anwender kann im File-Menü mit dem Befehl Archive Project das Projekt als nicht mehr veränderbar abspeichern. Dieses Projekt kann dann nur noch lesend geladen werden. Falls eine Änderung des Projekts nötig sein sollte, muss das Projekt unter einem anderen Namen gespeichert und mit dieser Kopie weitergearbeitet werden. Der Anwender kann beim Speichern einstellen, dass ein archiviertes Projekt auch nicht mehr als Kopie gespeichert werden kann.

6.3 Speichern als XML-Datei

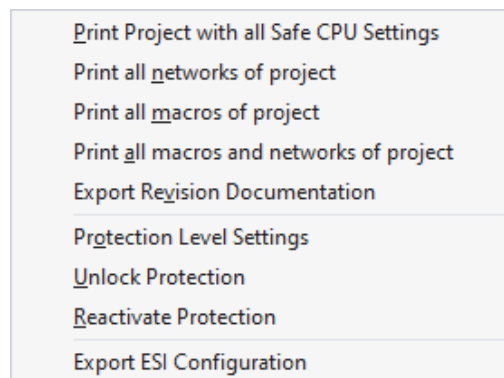
Der Anwender kann über die Einstellung ‚Save in XML-File‘ das Projekt unverschlüsselt als XML-Datei abspeichern. Die XML-Datei beinhaltet die zum Projekt dazugehörige CRC, somit ist die Datei vor Manipulationen abgesichert. Die XML-Datei kann auch zum Öffnen des Projekts verwendet werden.

7. Protection Level Settings Passwort-Mechanismus

7. Protection Level Settings Passwort-Mechanismus

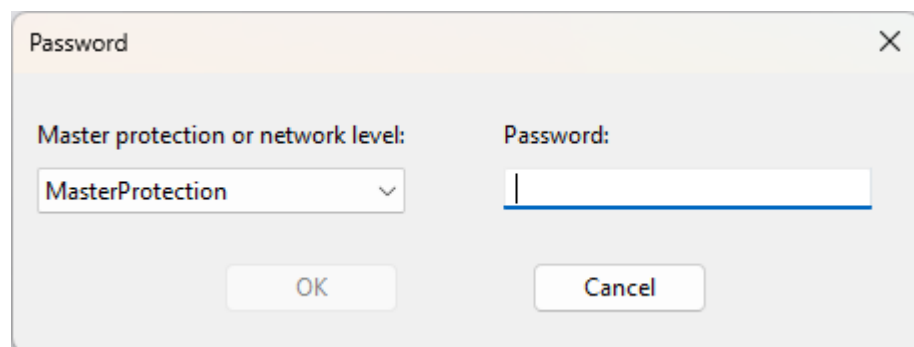
Über ein Passwort kann entweder das ganze Projekt oder einzelne Netzwerke vor einer Veränderung geschützt werden.

Über das Kontextmenü beim **Hardware Tree**, **Project Tree** und **Variablen Tree** kann der Dialog **Network Level** zum Erstellen bzw. Ändern der Level und Passwörter aufgerufen werden. Ebenfalls kann über das Kontextmenü ein **Unlock** bzw. **Reactivate** der Protection Settings aufgerufen werden.



Der Dialog für die Passwörter und Level kann nur aufgerufen werden, wenn noch kein Schreibschutz für das Projekt besteht bzw. bei einem bestehendem Schreibschutz muss das Einloggen mit dem **MasterProtection**-Passwort erfolgt sein.

Das Einloggen erfolgt über einen Dialog, bei welchem das Level ausgewählt und das entsprechende Passwort eingegeben wird.



Im Dialog **Network Level** können die Master Protection und die einzelnen Netzwerk Level bearbeitet werden.

7. Protection Level Settings Passwort-Mechanismus

Ind...	Level Name	Password	Confirm Password	Show Networks with lower Index	Comment
0	MasterProtection	*****	*****		Master protection of w...
1	Developer	*****	*****	Yes	
2	User	*****	*****	No	

☐ Show all networks without password

Insert new Level OK Cancel

Die Master Protection ist automatisch vorgegeben. Der Anwender muss nur das Passwort festlegen und bestätigen. Das Passwort muss zwischen 3 und 8 Zeichen lang sein. Die Master Protection kann nicht gelöscht werden. Durch das Entfernen des Passworts bei der Master Protection wird der komplette Passwortschutz des Projekts entfernt.

Bei eingestellter Master Protection können über den „Insert new Level“ Button einzelne Level für die Netzwerke festgelegt werden. Es wird ein Level mit einem Default-Index und einem Default-Namen angelegt. Der Index ist eindeutig und liegt zwischen 1 und 100. Der Index kann geändert werden. Die Sortierung der Level erfolgt über den Index. Der Levelname kann ebenfalls geändert werden und muss eindeutig sein. Es muss für jedes Level ein Passwort vergeben werden. Es kann zusätzlich zu jedem Level ein erklärender Kommentar vergeben werden.

Über die Checkbox „Show all networks without password“ kann festgelegt werden, ob alle Netzwerke ohne Passwort geöffnet werden können. Ist die Checkbox nicht gesetzt, dann kann bei den einzelnen Levels eingestellt werden, ob Netzwerke mit einem niedrigeren Index lesend geöffnet werden dürfen oder nicht.

Die eingestellten Daten im Dialog werden erst mit OK in das Projekt übernommen. Der OK-Button ist erst freigeschaltet, wenn alle benötigten Daten eingetragen sind (z.B. müssen alle Netzwerk-Level ein Passwort haben).

Bei den einzelnen Netzwerken kann im Properties Pane das entsprechende Level eingestellt werden. Das Netzwerk ist nur bearbeitbar, wenn das Einloggen im Projekt mit dem entsprechenden Level erfolgt ist. Das Einloggen mit einem übergeordneten Level (Index ist niedriger als festgelegter Level) erlaubt ein Bearbeiten des Netzwerkes. Ist bei einem Netzwerk kein Level festgelegt, kann das Netzwerk nur mit dem Master Protection bearbeitet werden.

Im Tree für die Netzwerke werden Netzwerke, welche nur mit einem bestimmten Schutz geöffnet werden können, farblich (blau) dargestellt.

8. Verwendung unsicherer Variablen über den CAN-Bus

8.1 Voraussetzung zur Verwendung

8. Verwendung unsicherer Variablen über den CAN-Bus

8.1 Voraussetzung zur Verwendung

Die Verwendung von unsicherer Variablen über den CAN-Bus ist nur im Stand-Alone-Betrieb der Safety-CPU's möglich.

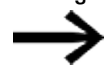
Die Safety-CPU muss dazu über die CAN-Schnittstelle mit einer entsprechenden Gegenstelle verbunden sein. Die unsicheren Variablen können dann über diese Schnittstelle von der Safety-CPU gesendet bzw. empfangen werden.

8.2 Aufbau der unsicheren Variablen

Die unsicheren Variablen über den CAN-Bus haben eine Größe von 8 Byte. Diese 8 Byte können im Safety Designer verschieden aufgeteilt werden.

- Zwei 4 Byte Werte
- Ein 4 Byte Wert und 32 Bitwerte
- 64 Bitwerte

Je nach Verwendung dieser Subvariablen ändert sich der Aufbau und die Größe der Übertragungsdaten. Die CAN-Daten sind immer so aufgebaut, dass erst die DINT-Werte und dann die BIT-Werte stehen. Die Größe der Daten hängt davon ab, welche DINT-Werte bzw. BIT-Werte verwendet werden. Bei der Verwendung einer Variablen mit 64 BIT-Werten bestimmen die im Netzwerk benutzten Bits die Größe des CAN-Pakets. Bei der Platzierung der Bits 1 bis 8 wird eine CAN-Länge von einem Byte belegt; bei der Verwendung der Bits 57 bis 64 beträgt die Größe der Daten die vollen 8 Bytes. Wird ein 4 Byte Datentyp (DINT) und die Bits 1 bis 8 verwendet, beträgt die Länge des Pakets 5 Byte.



Es wird darauf hingewiesen, die CAN Subvariablen von Bit 1 nach Bit 64 zu belegen, sowie DINT 1 vor DINT 2 zu verwenden. Die Größe der CAN-Pakete kann so minimiert werden.

Werden nur die Bits 57-64 verwendet, beträgt die Länge der CAN Daten trotzdem 8 Byte.



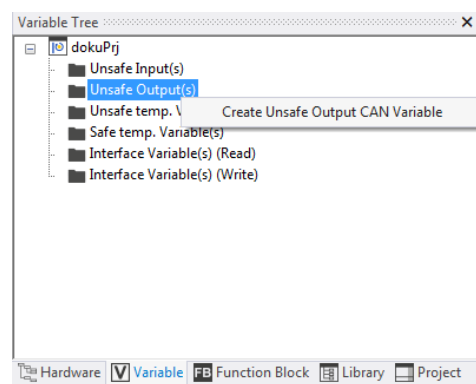
Die Baudrate der CAN-Pakete beträgt 500.000.

8. Verwendung unsicherer Variablen über den CAN-Bus

8.3 Erstellen eines Projekts mit der Verwendung der unsicheren Variablen über den CAN-Bus

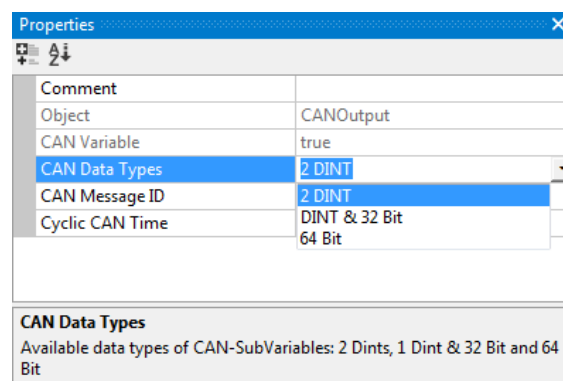
8.3 Erstellen eines Projekts mit der Verwendung der unsicheren Variablen über den CAN-Bus

Im Hardware Tree muss eine Stand-Alone-CPU angelegt werden. Das Anlegen der Safety-CPU schaltet das Erstellen von unsicheren Variablen über den CAN-Bus frei. Mittels Kontextmenü können die unsicheren Eingangs- und Ausgangsvariablen im Variable Tree bei den unsicheren Variablen angelegt werden. Das Kontextmenü ist nur bis zur maximalen Anzahl von Variablen freigeschaltet (max. Anzahl sind 5 Eingangs- und 5 Ausgangsvariablen).



Die angelegten Variablen sind in deren Subvariablen unterteilt. Die Unterteilung erfolgt in DINT-Werte und BIT-Werte. Die entsprechende Unterteilung wird im Variable Tree und im Property Browser angezeigt. Bei der Erstellung einer Variablen werden 2 DINT-Subvariablen angelegt.

Die Aufteilung der Variablen kann im Property Browser geändert werden. Die Änderung ist nur möglich, solange keine Subvariable in einem Netzwerk platziert ist.



Auswahlmöglichkeiten:

- 2 DINT
- 1 DINT und 32 BIT
- 64 BIT

Die entsprechende Auswahl wird automatisch im Variable Tree angezeigt.

8. Verwendung unsicherer Variablen über den CAN-Bus

8.3 Erstellen eines Projekts mit der Verwendung der unsicheren Variablen über den CAN-Bus

Der Anwender hat die Möglichkeit den Namen der Variablen und Subvariablen im Variable Tree zu ändern. Die Reihenfolge kann nicht geändert werden.

Im Property Browser muss für jede Variable eine Message ID vergeben werden, welche die CAN Kommunikation eindeutig festlegt. Die Message IDs beginnen bei 513 und werden vom XN Safety Designer automatisch vergeben. Es können keine Message IDs doppelt vergeben werden. Der Vergabebereich der IDs liegt zwischen 513 und 2047. Der Anwender kann die Message ID im vorgegebenen Bereich im Property Browser selbst ein-stellen.



Die Message IDs müssen im gesamten System eindeutig sein. Der XN Safety Designer kann die Eindeutigkeit nur innerhalb eines Projekts prüfen. Bei der Verwendung mehrerer Safety-Projekte in einem System muss der Anwender die Message IDs auf Eindeutigkeit prüfen.

Properties	
Comment	
Object	CANOutput
CAN Variable	true
CAN Data Types	2 DINT
CAN Message ID	1234
Cyclic CAN Time	50

CAN Message ID
Message ID of CAN Variable. Range: 513 - 2047

Bei den Ausgangsvariablen kann im Property Browser zusätzlich eine Zykluszeit eingestellt werden. Die Safety-CPU versendet die unsichere Variable im Abstand der eingestellten Zeit. Die angegebene kann sich etwas erhöhen, da die Daten immer nur am Ende eines Safety-CPU Zyklus geschrieben werden können; dies bedeutet, dass sich die wirkliche Übertragungszeit aus Zykluszeit und eingestellter Übertragungsrate zusammensetzt. Als Richtwert kann 5 bis 5000 [ms] eingestellt werden. Der Defaultwert beträgt 50 [ms].

Properties	
Comment	
Object	CANOutput
CAN Variable	true
CAN Data Types	2 DINT
CAN Message ID	513
Cyclic CAN Time	1000

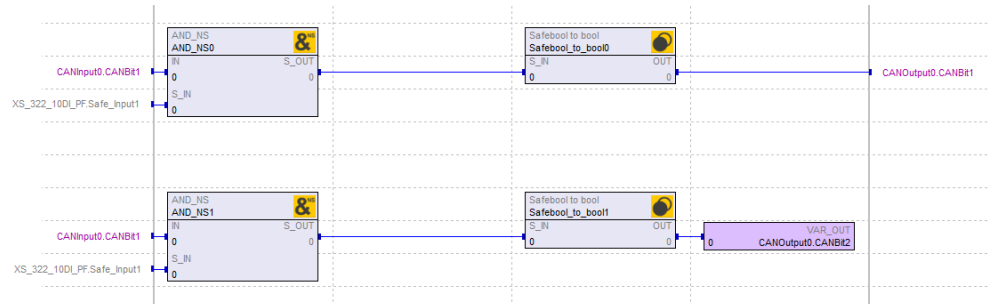
Cyclic CAN Time
Cyclic Transmission Time of CAN Variable. Range: 5-5000[ms]

Die Subvariablen können per Drag & Drop ins Netzwerk gezogen und platziert werden. Die Ausgänge können nur einmal verwendet werden; die Eingänge können

8. Verwendung unsicherer Variablen über den CAN-Bus

8.4 Anzeigen der Variablen im Online-Betrieb

mehrmals platziert werden. Das Cursor-Symbol zeigt an, ob eine Subvariable platziert werden kann.



Das Bild (s.o.) zeigt die Verwendung von CAN Variablen im Netzwerk.



Zu beachten ist, dass es sich bei den Variablen über den CAN-Bus um unsichere Daten handelt.

Die Verwendung der unsicheren Variablen hat Einfluss auf die maximale Zykluszeit der Safety-CPU. Die CAN-Pakete müssen zusammengestellt und versendet/empfangen werden. Die Berechnung der Safety-Zykluszeit durch den XN Safety Designer berücksichtigt dieses Verhalten. Die CAN-Datenpakete werden am Ende eines CPU-Zyklus versendet.

8.4 Anzeigen der Variablen im Online-Betrieb

Die Werte der Variablen werden im Onlinebetrieb im Netzwerk und im Watch angezeigt. Zur Anzeige im Watch müssen die Elemente mittels Drag & Drop aus dem Netzwerk in das Watch-Fenster gezogen werden.

Watch			
Name	Value	Type	
XS_102_C00_000~XS_322_200_RNOAC			
XS_102_C00_000~XS_322_4AI_12			
XS_102_C00_000NetworkXS_322_4AI_12.S_AI_1	5996	SAFEDINT	
XS_102_C00_000NetworkCANInput.CANDint1	???	DINT	
XS_102_C00_000NetworkCANInput.CANDint2	???	DINT	
Insert new via Drag&Drop or context menu (only in Online Mode)			
Output Watch Search			

Bit-Variablen können den Wert true und false annehmen. DINT-Variable besitzen einen numerischen Wert. Die Werte können nicht geforced werden.

8. Verwendung unsicherer Variablen über den CAN-Bus

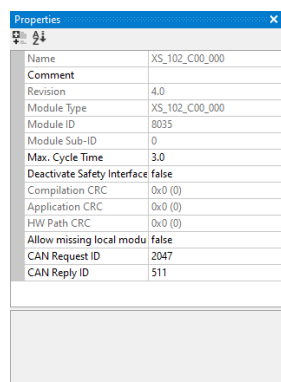
8.5 Austausch von Statusinformationen über CAN-Bus zur Safety-CPU

8.5 Austausch von Statusinformationen über CAN-Bus zur Safety-CPU

Zum Auslesen von Status-Informationen einer Stand-Alone-Safety-CPU über den CAN-Bus wird standardmäßig die CAN-Massage-ID 2047 verwendet.

Zum Senden der Antwort wird die CAN-Massage-ID 511 verwendet.

Zum Ändern der Werte können im Property Browser der CPU IDs aus dem Bereich der CAN-Variablen (513-2046) eingetragen werden. Die Punkte dazu sind **CAN Request ID** und **CAN Reply ID**. Diese Nummern sind dann für die Verwendung in den Variablen gesperrt.



Name	XS_102_C00_000
Comment	
Revision	4.0
Module Type	XS_102_C00_000
Module ID	8035
Module Sub-ID	0
Max. Cycle Time	3.0
Deactivate Safety Interface	false
Compilation CRC	0x0 (0)
Application CRC	0x0 (0)
HW Path CRC	0x0 (0)
Allow missing local modu	false
CAN Request ID	2047
CAN Reply ID	511

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

Bei Stand-alone-Betrieb muss das Schreiben einer unsicheren Variablen auf ein Safety-CPU bzw. das Lesen einer unsicheren Variablen vom Anwender übernommen werden. Die Kommunikation erfolgt über die sogenannte SDO-Kommunikation. Diese Kommunikation muss vom Anwender entsprechend aufgebaut werden, um die Daten mit der Safety-CPU austauschen zu können.

9.1 SDO-Kommunikation

Über die SDO-Kommunikation können verschiedene Befehle an die Safety-CPU geschickt werden. Diese Befehle werden von der Safety-CPU verarbeitet und die Antwort zurückgesendet.

Beispiele für SDO-Befehle:

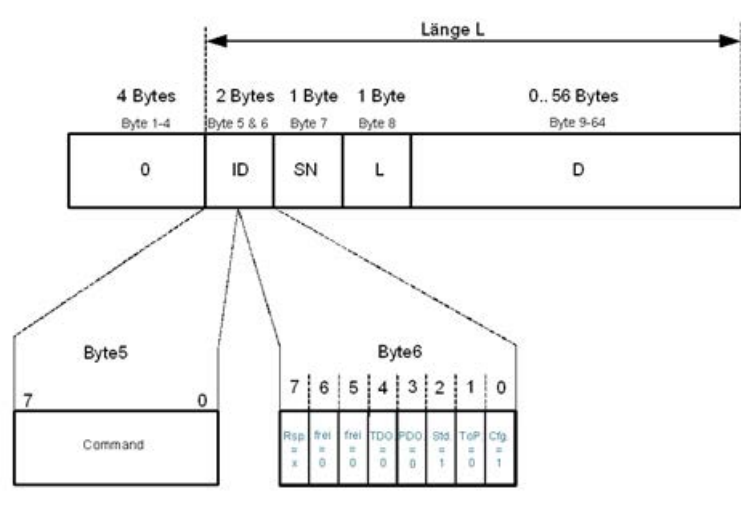
- Auslesen von unsicheren Variablen
- Schreiben von unsicheren Variablen

Zum Schreiben und Lesen eines SDO-Frames gibt es bei der Safety-CPU einen SDO-Write Buffer mit einem SDO-Write Status Register und eine SDO-Read Buffer mit einem SDO-Read Status Register.

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.2 Aufbau SDO-Frame

9.2 Aufbau SDO-Frame



Beim Feld Command wird das jeweils verwendete SDO-Kommando eingetragen (siehe Übersicht).

In Byte 6 wird der Telegrammtyp angegeben, welcher für die Anfrage unsicherer SDOs immer $00000101_2 (=5_{10})$ für die Anfrage bzw. $10000101_2 (=133_{10})$ für die Antwort ist.

Danach wird im Feld SN eine Sequenznummer eingetragen, die bei jeder neuen Anfrage erhöht wird.

Im Feld L befindet sich die Länge die sich über den gesamten Frame ohne dem fix mit 0 belegten CRC-Feld (die ersten 4 Byte) erstreckt.

Nach dem Längenfeld folgen die Daten, die je nach Kommando unterschiedlich sind.

Die Firmware der Safety-CPU spiegelt diesen Frame-Aufbau für die Antwort (gleiches Kommando, gleiche Sequenznummer und Response-Bit im Typenfeld gesetzt),

fügt die kommandospezifischen Antwortdaten in das Daten-Feld ein und setzt dann die Länge entsprechend.

9.3 Schreiben eines SDO-Frames

Es wird das SDO-Write Status Register (Offset: 0x01D1 bei XS-102-C00-000) gelesen. Wenn das Bit 7 in diesem Register 0 ist, dann kann der SDO-Write Puffer (Offset: 0x00C0 bei XS-102-C00-000) beschrieben werden.

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.4 Lesen eines SDO-Frames

9.4 Lesen eines SDO-Frames

Es wird das SDO-Read Status Register (Offset: 0x01D0 bei XS-102-C00-000) gelesen. Wenn das Bit 7 in diesem Register 1 ist, dann sind im SDO-Read Puffer (Offset: 0x0188 bei XS-102-C00-000) Daten enthalten. Die Anzahl der Byte im Puffer steht im SDO-Read Status auf den Bits 0 bis 6.

9.5 Übersicht über SDO-Kommandos

Cmd	Bezeichnung	Beschreibung
0	READ	Wird zum Lesen eines Speicherbereichs verwendet (Auslesen der Konfiguration, zyklische Prüfung der Config-CRC) → "Detailbeschreibung SDO-Kommando READ", Seite 133
1	WRITE	Wird zum Schreiben eines Speicherbereichs verwendet → "Detailbeschreibung SDO-Kommando WRITE", Seite 133
4	GET_STATE	Liefert den aktuellen Laufzeit-Zustand, Config-Zustand, Login-Level und gespeicherte CRCs aus der Konfiguration → "Detailbeschreibung SDO-Kommando GET_STATE", Seite 134
6	GET_DIAG_INFO	Liefert Diagnose-Informationen (Fehler-Code, wenn ein Fehler anliegt) → "Detailbeschreibung SDO-Kommando GET_DIAG_INFO", Seite 135
15	GET_SAFENBR	Liefert die Sicherheitsnummer eines Moduls → "Detailbeschreibung SDO-Kommando GET_SAFENBR", Seite 136
17	GET_FIRMWARE_VER	Liefert die Firmware-Version → "Detailbeschreibung SDO-Kommando GET_FIRMWARE_VER", Seite 136
19	NOP	Dummy-Befehl zum Leeren des Out-Buffers in einem Safe-Modul → "Detailbeschreibung SDO-Kommando GET_NOP", Seite 137
28	READ_VALUES	Allgemeiner Lesebefehl zum Auslesen eines oder mehrerer Speicherbereiche in einem virtuellen Speicherbereich → "Detailbeschreibung SDO-Kommando READ_VALUES", Seite 138
30	WRITE_VALUES	Allgemeiner Schreibbefehl zum Beschreiben eines oder mehrerer Speicherbereiche in einem virtuellen Speicherbereich → "Detailbeschreibung SDO-Kommando WRITE_VALUES", Seite 138

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.6 Detailbeschreibung SDO-Kommando READ

9.6 Detailbeschreibung SDO-Kommando READ

Allgemeiner Lesebefehl zum Auslesen eines Speicherbereichs. Mit diesem Befehl können die Konfiguration und nicht sichere Variablen ausgelesen werden. Mit der Adresse wird angegeben, welche Daten gelesen werden. Um eine nicht sichere Variable auszulesen, benötigt man den Offset dieser Variable im Applikationsspeicher. Dieser wird dann zur Adresse des Applikationsspeichers addiert. Der Offset der Variable wird durch das Parsen der Konfigurationsdaten ermittelt.

Command: 0

Request-Daten	
4 Byte	virtuelle Adresse des Speicherbereichs
1 Byte	Anzahl der zu lesenden Byte
Response-Daten	
1 Byte	Return-Code
1 Byte	Anzahl der gelesenen Byte. Der Wert 0 sagt aus, dass keine weiteren Daten vorhanden sind.
n Byte	gelesene Daten

9.7 Detailbeschreibung SDO-Kommando WRITE

Mit diesem Befehl können nicht sichere Variablen beschrieben werden. Mit der Adresse wird angegeben, welche Daten gelesen werden. Um eine nicht sichere Variable zu beschreiben, benötigt man den Offset dieser Variable im Applikationsspeicher. Dieser wird dann zur Adresse des Applikationsspeichers addiert. Der Offset der Variable wird durch das Parsen der Konfigurationsdaten ermittelt.

Command: 1

Request-Daten	
4 Byte	virtuelle Adresse des Speicherbereichs
1 Byte	Anzahl der zu schreibenden Byte
n Byte	zu schreibende Daten
Response-Daten	
1 Byte	Return-Code

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.8 Detailbeschreibung SDO-Kommando GET_STATE

9.8 Detailbeschreibung SDO-Kommando GET_STATE

Liefert den aktuellen Laufzeit-Zustand, Config-Zustand, Login-Level und gespeicherte CRCs aus der Konfiguration. Die CRCs aus der Konfiguration sind nur gültig, wenn der Config-Zustand einen Wert > NOT_CONFIGURED aufweist.

Beschreibung der Config-Zustände:

Bezeichnung	Wert	Beschreibung
UNKNOWN	0	unbekannt (z.B. während die Firmware heruntergeladen wird)
INVALID	1	ungültig (wenn z.B. ein Fehler in der Konfiguration festgestellt wurde)
NOT_CONFIGURED	2	unkonfiguriert, d.h. die Konfiguration wurde explizit gelöscht.
CONFIGURED_NOT_DEPLOYED_NOT_VERIFIED	4	konfiguriert, aber nicht verteilt und auch nicht verifiziert
CONFIGURED_AND_VERIFIED	8	konfiguriert, verteilt und verifiziert
CONFIGURED_DEPLOYED_NOT_VERIFIED	16	konfiguriert, verteilt aber noch nicht verifiziert
CONFIGURED_NOT_DEPLOYED_NOT_VERIFIED_DEV	36	konfiguriert, aber nicht verteilt und auch nicht verifiziert + Developer-Mode
CONFIGURED_DEPLOYED_NOT_VERIFIED_DEV	48	konfiguriert, verteilt aber noch nicht verifiziert + Developer-Mode

Command: 4

Request-Daten	
1 Byte	optional: Flags

Wenn das Bit 0 in Flags gesetzt ist, dann wird der Zähler für die Änderung an den Geräteadressen mitgeschickt, sonst nicht.

Response-Daten	
1 Byte	Returncode
1 Byte	Laufzeit-Zustand
	1 POST
	2 SERVICE
	4 ERROR
	8 IDLE
	16 CHK_CFG
	32 OP_TEMP
	64 OP

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.9 Detailbeschreibung SDO-Kommando GET_DIAG_INFO

Response-Daten	
1 Byte	Config-Zustand 1 INVALID 2 NOT_CONFIGURED 4 CONFIGURED_NOT_DEPLOYED_NOT_VERIFIED 8 CONFIGURED_AND_VERIFIED 16 CONFIGURED_DEPLOYED_NOT_VERIFIED 36 CONFIGURED_NOT_DEPLOYED_NOT_VERIFIED_DEV 48 CONFIGURED_DEPLOYED_NOT_VERIFIED_DEV
1 Byte	Login-Level
4 Byte	CRC des Konfigurationscontainers. Der Konfigurationscontainer enthält bei einer Safety-CPU die eigene und die an die entfernten Module zu sendenden Konfigurationen.
4 Byte	CRC des Listenheaders der Konfiguration
2 Byte	Fehlerzähler I/O-Status. Wird bei jeder Änderung des Fehlerstatus eines lokalen Eingangs oder Ausgangs inkrementiert
optional	
4 Byte	CRC des Interfaceframes, den das Modul erstellt. Der Wert 0 bedeutet, dass kein Interface-Frame erstellt wird.
	Zähler für die Änderung an den Geräteadressen im Adressvergabeservice (können an der Adresse V_ADDR_FSOE_DEVADDR abgefragt werden). Dieser Wert wird nur geschickt, wenn im Request im Element Flags das Bit 0 gesetzt ist.

9.9 Detailbeschreibung SDO-Kommando GET_DIAG_INFO

Liefert Diagnose-Informationen. Die von der Funktion zurückgegebenen Diagnose-Informationen können mit der Hilfe der Errorcode-Liste ausgewertet werden.

Command: 6

Request-Daten	
1 Byte	Auswahl µC1 oder µC2 0 µC1 ≠0 µC2
Response-Daten	
1 Byte	Returncode
1 Byte	Auswahl µC1 oder µC2 0 µC1 ≠0 µC2
1 Byte	aktueller Fehler-Code
1 Byte	der erste Fehler, der den Fehlerzustand ausgelöst hat
4 Byte	zum Fehler-Code (erster Fehler) zugehöriger Reason-Code 0 (Bedeutung je nach Fehler-Code unterschiedlich)
4 Byte	zum Fehler-Code (erster Fehler) zugehöriger Reason-Code 1 (Bedeutung je nach Fehler-Code unterschiedlich)

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.10 Detailbeschreibung SDO-Kommando GET_SAFENBR

9.10 Detailbeschreibung SDO-Kommando GET_SAFENBR

Liefert die Sicherheitsnummer eines Moduls.

Bei diesem Befehl gibt es die Besonderheit, dass beim Request ein eigener SSDO-Typ (SSDO-Cfg oder SSDO-Mod Frame) verwendet wird, bei dem als Ziel-Adresse nicht die Modul-Sicherheitsnummer, sondern der Topology-Pfad bzw. eine komprimierte Version des Topolgy-Pfades angegeben wird.

Ein Safe-Modul hat keine Kenntnis über die Bedeutung und den Aufbau des Topology-Pfads. Wenn ein C_GET_SAFENBR Request mit einem Topology-Pfad als Zieladresse bei einem Safe-Modul eintrifft, dann wird der Topology-Pfad nicht interpretiert und es wird eine Response mit der eigenen Modul-Sicherheitsnummer als Absenderadresse zurückgeschickt.

Command: 15

Request-Daten	
	keine
Response-Daten	
1 Byte	Return-Code
4 Byte	Modul-Sicherheitsnummer

9.11 Detailbeschreibung SDO-Kommando GET_FIRMWARE_VER

Liefert die Firmware-Version

Command: 17

Request-Daten	
	keine
Response-Daten	
1 Byte	Return-Code
8 Byte	Firmware-Version
	4 Byte 2 Byte 2 Byte
	Minor-Version Major-Version Modul-Art
	Wenn sich das Modul im Laufzeit-Zustand 'Bootloader' befindet, dann ist in der Minor-Version die Bootloader-Version enthalten.
4 Byte	Bootloader-Version

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.12 Detailbeschreibung SDO-Kommando GET_NOP

9.12 Detailbeschreibung SDO-Kommando GET_NOP

Dummy-Befehl zum Leeren des Out-Buffers in einem Safe-Modul.

Dieser Befehl wird von der Standard-SPS benötigt, um nach einem Timeout den Out-Puffer in einem Safe-Modul zu leeren. Nachdem die Response auf diesen Befehl empfangen worden ist, kann die SPS davon ausgehen, dass keine Response auf einen früheren Befehl im Out-Buffer des Safe-Moduls gespeichert ist.

Erforderlicher Login-Level: 0

Command: 19

Request-Daten	
	keine
Response-Daten	
1 Byte	Return-Code

9.13 Detailbeschreibung SDO-Kommando READ_VALUES

Allgemeiner Lesebefehl zum Auslesen eines oder mehrerer Speicherbereiche in einem virtuellen Speicherbereich.

Mit diesem Befehl können im Unterschied zu C_READ gleiche mehrere Adressen angegeben werden. Die Größe eines mit einer Adresse angegebenen Speicherbereichs ist durch die Art des virtuellen Speichers implizit festgelegt. Beim Applikationsspeicher ist diese Größe 4 Byte (= Größe einer Zelle im Applikationsspeicher).

Command: 28

Request-Daten	
1 Byte	Anzahl der folgenden Adressen (n)
n * 4 Byte	virtuelle Adressen der Speicherbereiche
Response-Daten	
1 Byte	Return-Code
1 Byte	Anzahl der gelesenen Werte
n * x Byte	Gelesenen Speicherbereiche mit jeweils x Byte. Für den Applikationsspeicher hat x den Wert 4.

9.14 Detailbeschreibung SDO-Kommando WRITE_VALUES

Allgemeiner Schreibbefehl zum Beschreiben eines oder mehrerer Speicherbereiche in einem virtuellen Speicherbereich.

Die Adresse des Speicherbereichs ist keine physikalische, sondern eine virtuelle Adresse. Die Werte der virtuellen Adressen sind nach der Beschreibung der Service-Befehle zu finden.

Mit diesem Befehl können im Unterschied zu C_WRITE gleiche mehrere Adressen angegeben werden. Die Größe eines mit einer Adresse angegebenen Speicherbereichs ist durch die Art des virtuellen Speichers implizit festgelegt. Beim Applikationsspeicher ist diese Größe 4 Bytes (= Größe einer Zelle im Applikationsspeicher).

Command: 30

Request-Daten	
1 Byte	Anzahl der folgenden Adressen (n)
n * (4+4) Byte	virtuelle Adressen der Speicherbereiche und deren Werte
Response-Daten	
1 Byte	Return-Code

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.15 Wichtige virtuelle Adressen der Safety-CPU

9.15 Wichtige virtuelle Adressen der Safety-CPU

Adressoffset	Bezeichnung	Beschreibung
0x2000 0000	CONFIG_DATA	Basisadresse der Konfigurationsdaten
0x2010 0000	SER_NR_DATA	Basisadresse der Serien- und Safety-Nummern
0x2070 0000	OPTIONAL_CFG	Hier können Optional-Flags und die IDs der erwarteten Module ausgelesen werden. Diese sind wie folgt aufgebaut: 2 Byte Optionalflags (jedes Bit gibt an ob das Modul an diesem Steckplatz hinter der Safety-CPU als optional konfiguriert wurde) 1 Byte Optionalflag HBG (das Bit0 gibt an, ob das Handbediengerät als optional konfiguriert wurde) 16 Byte Array mit 16 x 1 Byte für die IDs der am jeweiligen Steckplatz erwarteten Module 1 Byte Erwartete ID des HBG
0x3000 0000	APP_DATA	Applikationsspeicher Der Applikationsspeicher ist der RAM-Speicher, der bei der Verarbeitung der Sicherheitsapplikation benötigt wird. Durch Lesen aus diesem Bereich können die Werte von Variablen ermittelt werden, durch Beschreiben dieses Bereichs werden Werte geforced.
0x3400 0000	UNSAFE_BDINT	Adressbereich für unsichere BDINT-Variablen (32 Bit, bei denen jedes einzelne Bit im Safety Designer getrennt verknüpft werden kann).

9.16 Blocktypen der Konfiguration

Nummer	Bezeichnung	Beschreibung
19	CFG_CONTAINER_V3	Blocktyp der Gesamtkonfiguration (enthält Modulkonfigurationen)
21	CFG_REV	Safety Designer Projektname und Revisionsnummernstring
40	CFG_MODULE_V3	Blocktyp einer Modulkonfiguration (inkl. SafeDINT)

9.17 Auslesen der Konfiguration

Die Konfiguration besteht aus 4 Byte Werten, wobei Strings eine Ausnahme darstellen und dort die angegebene Länge zu verwenden ist.

Zu Beginn werden von der Adresse CONFIG_DATA die ersten 12 Byte der Konfiguration ausgelesen: CRC, Länge der gesamten Konfiguration und Blocktyp der Konfiguration (CFG_CONTAINER..).

Wenn der Blocktyp der Konfiguration CFG_CONTAINER_V3 ist gibt es einen dynamischen Header gleich hinter den bisher ausgelesenen Daten (die Länge der Daten ist in den ersten 4 Byte des Headers vermerkt = Offset 0x0C der Konfiguration).

Zur Adresse des dynamischen Headers wird dessen Länge addiert, um zum ersten Listeneintrag zu gelangen.

Wenn der Blocktyp der Konfiguration nicht CFG_CONTAINER_V3 ist, beginnt der erste Listeneintrag 4 Byte hinter den bisher gelesenen Daten (Offset 0x10 der Konfiguration).

Mit Hilfe der Länge der Gesamten Konfiguration als Obergrenze für den Offset wird nun eine Schleife gestartet, um die einzelnen Listen durchzugehen.

Genereller Aufbau eines Listeneintrags (unabhängig vom Blocktyp)

4 Byte	Länge des Modulpfads gefolgt vom Pfad-String mit der entsprechenden Länge.
4 Byte	Länge des komprimierten Modulpfads gefolgt vom Pfad-String mit der entsprechenden Länge.

Danach liegen im Speicher CRC, Gesamtlänge, Blocktyp und ModuleID der aktuellen Liste. Um nun zur nächsten Liste zu gelangen addiert man zum Offset der Gesamtlänge den Wert der Gesamtlänge hinzu.

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.18 Aufbau der einzelnen Blocktypen der Listeneinträge

9.18 Aufbau der einzelnen Blocktypen der Listeneinträge

9.18.1 CFG_MODULE_V3

Dieser Blocktyp beschreibt eine Modulkonfiguration inklusive sicherer und unsicherer Ein- und Ausgänge.

Berechnung der Adressen im Applikationsspeicher für den späteren Zugriff auf die unsicheren Variablen (Speicheraufbau ergibt sich aus dem hier geparsten Konfigurationsspeicher).

Die Adresse beginnt mit 4 (Bereich 0-3 durch time belegt)

Es werden pro safebool 4 Byte und pro safedint 8 Byte belegt.

Die Bitvariablen (unsafe BDINT) belegen pro verwendetem Bit 4 Byte im Applikationsspeicher.

Die Adressen müssen für die einzelnen Elemente entsprechend der Verwendungen aufaddiert werden. Für den späteren Zugriff auf die unsicheren Variablen muss zum jeweiligen Offset noch die Basisadresse (APP_DATA bzw. UNSAFE_BDINT für unsichere BDINT Variablen) für den Zugriff auf den Applikationsspeicher addiert werden.

Aufbau des Blocks:

Bereich Interface-Frames (beim Stand-Alone gibt es keine Frames).

Aufbau: 4 Byte CRC

4 Byte Pfadlänge

Pfad

4 Byte Länge komp. Pfad

Komp. Pfad

Bereich FSoE Verbindungen (beim Stand-Alone gibt es keine Verbindungen), hier muss nur die Länge ausgelesen werden. Die Länge beträgt 4Byte.

Bereich mit den COPY-Verbindungen und den CALL-Aufrufen der Funktionsblöcke.

Der Bereich wird mit dem END-Befehl abgeschlossen.

Der Endbefehl wird durch die Zahl 2 repräsentiert.

Pro Aufruf werden 16 Byte benötigt.

4 Byte auslesen und Wert prüfen; wenn dieser Wert 2 ist der Bereich zu Ende.

Adresse immer um 16 Byte erhöhen.

Bereich lokale Eingänge (SafeBool).

Am Beginn steht die Anzahl der Eingänge (4 Byte-Wert).

Pro Anzahl werden 16 Byte benötigt.

Bereich Ausgang als Eingang (SafeBool).

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Anzahl werden 4 Byte benötigt.

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.18 Aufbau der einzelnen Blocktypen der Listeneinträge

Bereich sicherer Ausgang (SafeBool).

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Anzahl werden 4 Byte benötigt.

Bereich lokale Eingänge (SafeDINT).

Am Beginn steht die Anzahl der Eingänge (4 Byte-Wert).

Pro Anzahl werden 16 Byte benötigt.

Bereich Ausgang als Eingang (SafeDINT).

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Anzahl werden 4 Byte benötigt.

Bereich sicherer Ausgang (SafeDINT).

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Anzahl werden 4 Byte benötigt.

Bereich Fast unsafe Outputs.

Am Beginn steht die Anzahl (4 Byte-Wert).

Bereich unsafe Outputs.

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Ausgang: 4 Byte Länge des Namens

Namen der Variable

Bereich unsafe Outputs (BDINT).

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Ausgang eine 4 Byte Bitmaske.

Die Bitmasken aller Ausgänge stehen vor den Ausgängen (z.B. bei 3 unsicheren Ausgängen stehen zuerst die 3*4 Byte für die Bitmasken und dann erst die Namenslänge + Namen).

Pro Ausgang: 4 Byte Länge des Namens

Namen der Variable

Bereich sichere Eingänge (SafeBool).

Am Beginn steht die Anzahl der Eingänge (4 Byte-Wert).

Pro Eingang 4 Byte Pfadlänge

Pfad

4 Byte Länge komp. Pfad

Komp. Pfad

4 Byte Index

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.18 Aufbau der einzelnen Blocktypen der Listeneinträge

Bereich sichere Eingänge (SafeDINT).

Am Beginn steht die Anzahl der Eingänge (4 Byte-Wert).

Pro Eingang 4 Byte Pfadlänge

Pfad

4 Byte Länge komp. Pfad

Komp. Pfad

4 Byte Index

Bereich sichere Interface.Eingänge (SafeBool).

Am Beginn steht die Anzahl der Eingänge (4 Byte-Wert).

Pro Eingang 4 Byte Pfadlänge

Pfad

4 Byte Länge komp. Pfad

Komp. Pfad

4 Byte Index

Bereich sichere Interface.Eingänge (SafeDINT).

Am Beginn steht die Anzahl der Eingänge (4 Byte-Wert).

Pro Eingang 4 Byte Pfadlänge

Pfad

4 Byte Länge komp. Pfad

Komp. Pfad

4 Byte Index

Bereich Fast unsafe Inputs.

Am Beginn steht die Anzahl (4 Byte-Wert).

Bereich unsafe Inputs.

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Eingang: 4 Byte Länge des Namens

Namen der Variable

Bereich unsafe Inputs (BDINT).

Am Beginn steht die Anzahl (4 Byte-Wert).

Pro Eingang eine 4 Byte Bitmaske.

Die Bitmasken aller Eingänge stehen vor den Eingängen.

Pro Eingang: 4 Byte Länge des Namens

Namen der Variable

...

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.19 Status der I/Os der Erweiterungsmodule

9.18.2 CFG_REV

Dieser Blocktyp beinhaltet einen String (Länge des Blocks) der den Projektnamen und die Projektrevision beinhaltet (die beiden Werte sind durch ein Tabulatorzeichen getrennt).

Die Länge des Strings ist die Gesamtlänge der Liste minus 12 Byte (= Header ohne CRC, da CRC nicht in die Länge miteinfließt).

9.19 Status der I/Os der Erweiterungsmodule

Der Status der Input/Outputs (Ein- und Ausgänge) kann aus dem Adressbereich der Safety-CPU (0x008A) direkt (d.h. nicht über SDO-Befehle) ausgelesen werden.

Die Belegung des I/O-Status-Register ist nicht fix, sondern richtet sich danach wie die Module angesteckt sind. Zuerst kommen die I/Os des CAN-HBT Moduls, dann die I/Os der angesteckten lokalen Eingänge, dann die der angesteckten lokalen Ausgänge:

Byte 0:

Bit 1...0 Validation Button State

Bit 7...2 free

Byte 1:

Bit 15...8 Status LEDs (Can, Run, Service, Error)

Bits LED

0,1 Can

2,3 Run

4,5 Service

6,7 Error

Byte 2-3:

Bit 17 ... 16 I/O Status SafetyCan (HBG) Input 1

...

Bit 31 ... 30 I/O Status SafetyCan (HBG) Input 8

Byte 4..67

9. Verwendung unsicherer Variablen im Stand-alone-Betrieb

9.20 Erkannte Erweiterungsmodule

Bit 32 Status der angesteckten I/Os

...543

1. lokale Eingänge

2. lokale Ausgänge (Digitalausgänge und Relaisausgänge können gemischt vorkommen)

Pro Input, Output oder Status sind 2 Bits vorgesehen mit folgender Bedeutung:

Input, Output:

b00 nicht vorhanden

b01 Fehlerzustand

b10 Low

b11 High

Button-State, Status LED:

b00 langsam blinkend

b01 schnell blinkend

b10 aus

b11 ein

9.20 Erkannte Erweiterungsmodule

Welche Erweiterungsmodule angesteckt sind, kann durch das I/O Modules Register (0x01D4) abgerufen werden.

9.21 Schnelle nicht sichere Ein- und Ausgangsvariablen

Bei der Safety-CPU XS-102 können bis zu 32 Bit nicht sichere Ein- und Ausgangsvariablen direkt (d.h. nicht über SDO-Befehle) gelesen und geschrieben werden (0x0080 des Lese- bzw. Schreibbereichs).

9.22 Unsichere Freigabesignale für lokale Ausgänge

Lokale sichere Ausgänge können mit einem unsicheren Freigabesignal UND-verknüpft werden.

Die Freigabesignale werden über Offset 0x0088 des Schreibbereichs am Modul bedient.

10. Ermittlung der Reihenfolge für das Kompilat einer Safety-CPU

- "Reihenfolge der Netzwerke", Seite 146
- "Reihenfolge der Instanzen innerhalb der Netzwerke", Seite 147
- "Anzeige der Abarbeitungsreihenfolge in Netzwerken", Seite 147

10.1 Reihenfolge der Netzwerke

Es müssen hier mehrere Fälle unterschieden werden

Fall 1: Es werden keine temporären Variablen in den Netzwerken verwendet

Die Reihenfolge wird über den Namen der Netzwerke festgelegt (alphabetisch aufsteigend).

Fall 2: Es werden schreibende temporäre Variablen verwendet

Die Reihenfolge wird über den Namen der temporären Variablen festgelegt (alphabetisch aufsteigend). Es wird das Netzwerk als erstes bearbeitet, welches die temporäre Variable mit den ersten Buchstaben (alphabetisch aufsteigend) enthält.

Fall 3: Mischung von Netzwerken mit schreibenden temporären Variablen und ohne temporäre Variablen

Es werden zuerst die Netzwerke mit schreibenden temporären Variablen abgearbeitet (alphabetisch aufsteigend nach temporären Variablen) und danach die anderen Netzwerke (alphabetisch aufsteigend nach Netzwerken).

Fall 4: Mischung von Netzwerken mit temporären Variablen (lesend und schreibend) und ohne temporären Variablen

Es werden zuerst die Netzwerke mit schreibenden temporären Variablen abgearbeitet. Die Verwendungen der temporären Variablen werden gegeneinander untersucht, d.h. die Reihenfolge muss nicht mehr alphabetisch sortiert sein. Es muss immer sichergestellt sein, dass das Netzwerk mit der schreibenden temporären Variablen vor den Netzwerken mit der lesenden temporären Variablen kommt. Nach diesen Netzwerken kommen wieder die Netzwerke ohne temporäre Variable bzw. nur mit lesenden temporären Variablen (alphabetisch aufsteigend nach Netzwerken).

10. Ermittlung der Reihenfolge für das Kompilat einer Safety-CPU

10.2 Reihenfolge der Instanzen innerhalb der Netzwerke

10.2 Reihenfolge der Instanzen innerhalb der Netzwerke

Innerhalb der Netzwerke wird im Prinzip von links oben nach rechts unten gearbeitet.

Vom obersten linken Funktionsblock (erst Zeile, dann Spalte) wird nach rechts gesucht, bis ein Ende gefunden ist (z.B. Ausgang, temporäre schreibende Variable, ...). Von dort werden dann alle benötigten Funktionsblöcke nach links zurückgesucht. Man hat damit eine umgekehrte Abarbeitungsreihenfolge. Diese Liste wird einfach umgekehrt und man erhält somit eine Reihenfolge, in welcher die Funktionsblöcke abgearbeitet werden müssen.

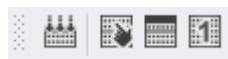
Alle bereits bearbeiteten Funktionsblöcke werden gemerkt und der Vorgang beginnt mit dem nächsten noch nicht bearbeiteten Funktionsblock links oben.

Der Vorgang wird so lange wiederholt bis alle Funktionsblöcke abgearbeitet sind.

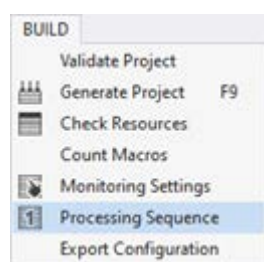
10.3 Anzeige der Abarbeitungsreihenfolge in Netzwerken

Die Anzeige der Abarbeitungsreihenfolge steht erst dann zur Verfügung, wenn das Projekt erfolgreich kompiliert wurde. Werden danach Änderungen vorgenommen, muss das Projekt erneut kompiliert werden.

Zum Aktivieren der Anzeige den Button  in der Build-Toolbar drücken.



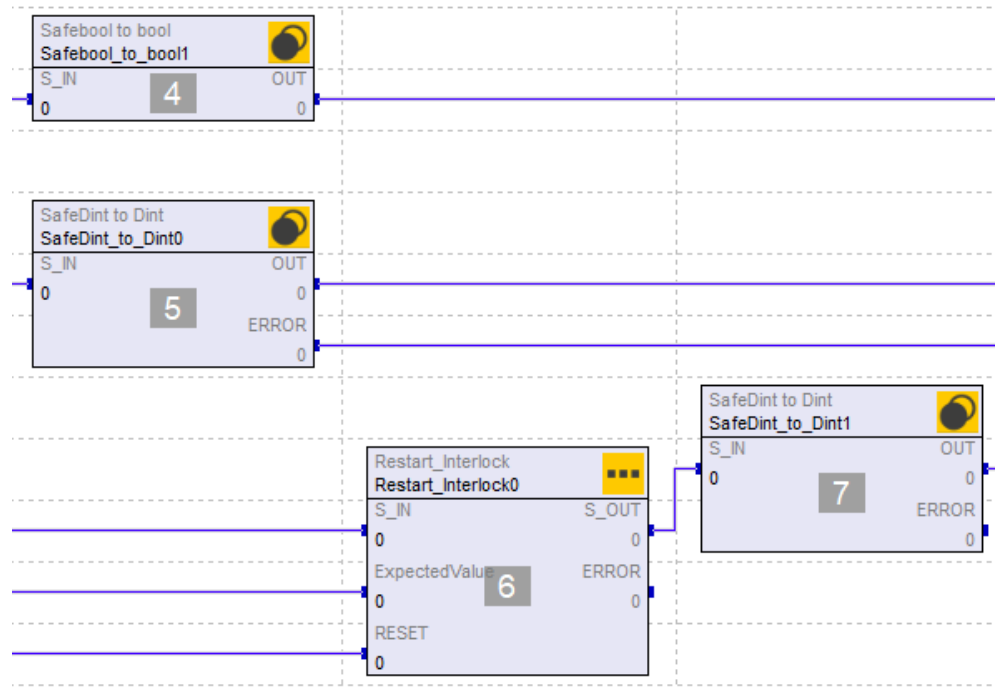
Alternativ kann man auch den Punkt **Processing Sequence** aus dem Build-Menü aktivieren.



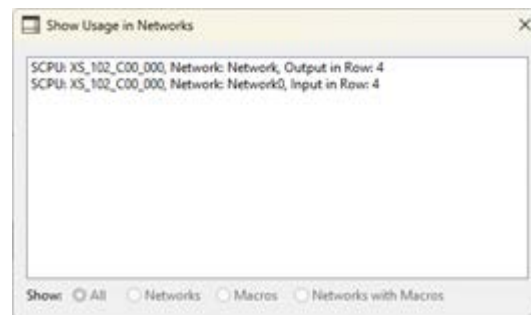
Im Netzwerk wird die Abarbeitungsreihenfolge mit grau hinterlegten Zahlen angezeigt.

10. Ermittlung der Reihenfolge für das Kompilat einer Safety-CPU

10.3 Anzeige der Abarbeitungsreihenfolge in Netzwerken



Mit dem Kontext-Menu-Punkt *Go to Previous Output(s)* kann zu der Stelle in einem anderen Netzwerk gesprungen werden, wo die Werte gesetzt wurden. Bei einer Angabe in eckigen Klammern können das durchaus mehrere sein. In solch einem Fall öffnet sich ein Fenster mit allen in Frage kommenden Stellen.



Mit Doppelklick auf einen Eintrag kann zu der Stelle gesprungen werden.

11. Online-Aktionen und Download

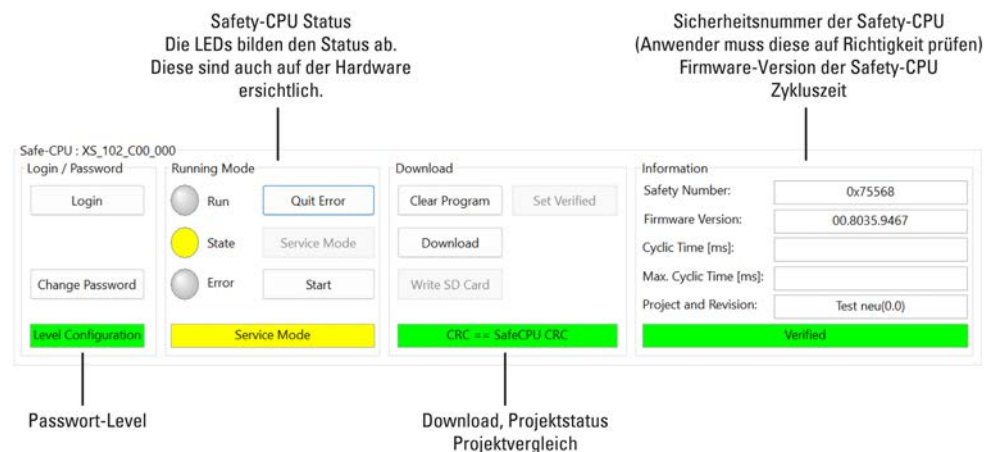
11. Online-Aktionen und Download

Nach dem Aufbau einer Online-Verbindung kann das Fenster „Online State“ geöffnet werden.

In diesem Fenster sind alle Safety-CPUs mit ihrem Status ersichtlich. Der Status wird ständig aktualisiert.

Dies betrifft folgende Informationen:

- Passwort-Level
- Service Mode-Status
- Status der Konfiguration
- Allgemeine Informationen (z.B. Firmware Version, Sicherheitsnummer der Safety-CPU, ...)

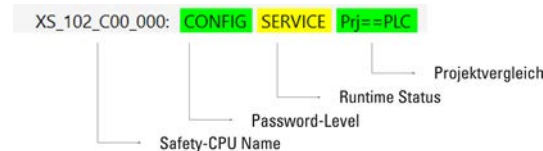


Folgende Informationen sind hier ersichtlich:

- Safety-CPUs ohne Module im Slave-Modus
- Alle Safety-CPUs
- Safety-CPU Bezeichnung
- Safety-CPU Status
- Download, Projekt Status
- Zykluszeit
- Projektvergleich
- Die LEDs bilden den Status ab. Diese LEDs sind auch auf der Hardware ersichtlich.
- Passwort level
- Firmware Version der Safety-CPU
- Sicherheitsnummer der Safety-CPU (Anwender muss diese auf Richtigkeit prüfen)
- Projektname und Projektversion

11. Online-Aktionen und Download

Nach dem Aufbau einer Online-Verbindung wird in der Statuszeile am unteren Fensterrand der Zustand der Safety-CPUs angezeigt. In der Statuszeile wird der Name der Safety-CPU, das Passwort-Level, der Laufzeit (Runtime)-Status und der Projektvergleich zur Safety-CPU angezeigt.



Im Online Editor wird ein Dialog mit den Fehlern der betroffenen Modulkonäle angezeigt. Mittels Doppelklick auf die Zeile kann das Element im Netzwerk selektiert werden. Die Anzeige der Fehler wird benötigt, da ein Modul nicht in einen generellen Fehlerzustand geht, wenn einzelne Kanäle einen Fehler aufweisen. Jeder Kanal und dessen Fehler wird aufgelistet. Die Fehler können mit dem Quit Error Befehl quittiert werden, das Safety-Modul versucht daraufhin die Abarbeitung wieder aufzunehmen. Zusätzlich werden im Dialog Fehler/Zustände der Slave-Module angezeigt. Die Anzeige ist nötig, um Fehler/Zustände bei den Slave-Modulen schneller zu sehen. Fehlerhafte Slave-Module werden im Dialog rot angezeigt. Es werden zusätzlich die Laufzeitzustände IDLE, Check configuration und Service angezeigt. Mittels Doppelklick kann das Element im Hardware-Tree selektiert werden um genauere Informationen zu ermitteln.

Errors of Channels	
Channel	Error
☒ XS_322_4AI_12	
└ XS_322_4AI_12.5_AI_1 (XS_102_C00_000, Network, Input Row 15)	Error(1189): The input has fallen below safe measuring range. The error ist automatically ack...

11. Online-Aktionen und Download

11.1 Soll-/ Istvergleich der Hardware

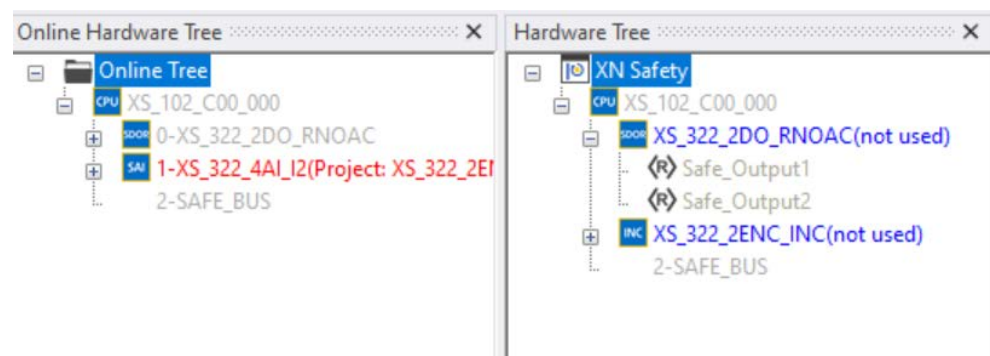
11.1 Soll-/ Istvergleich der Hardware

Der XN Safety Designer stellt eine Funktion zur Verfügung, die den Hardwarepfad (Topologiepfad) der Safety-Module im Safety-Projekt mit den physikalisch an der Steuerung vorhandenen Safety-Modulen vergleicht. Eine Konfiguration der Safety-CPU ist nur möglich, wenn das Ergebnis des Vergleichs positiv ist, d.h. es müssen alle im Projekt konfigurierten Safety-CPU's physikalisch an der entsprechenden Stelle vorhanden sein.

Dieser Vergleich erfolgt automatisch nach Aufbau einer Online-Verbindung. Schlägt der Vergleich fehl, wird ein Hardware Tree geöffnet, der die physikalisch vorhandenen Module aufzeigt. Der Anwender kann diesen Tree mit seinem Aufbau im Safety-Projekt vergleichen und die Unterschiede so korrigieren (der Fehler kann an einem falsch gesteckten Hardwaremodul an der Maschine oder an einer fehlerhaften Konfiguration im XN Safety Designer liegen).

Die Darstellung der Unterschiede im Hardware Tree erfolgt farblich (rot für unterschiedliche Module an derselben Stelle, bräunlich für Module, welche nur auf der SPS vorhanden sind und blau für Module, welche nur im Projekt vorhanden sind). Zusätzlich werden Texte angefügt, um die farbliche Darstellung zu unterstützen. Es wird zusätzlich vor allen Modulen die Position (Steckplatznummer) angezeigt, um bei z.B. Erweiterungsmodulen einen leichteren Vergleich zu ermöglichen.

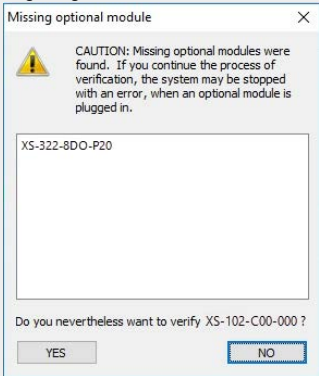
Es ist möglich mittels Kontextmenü einzelne Module aus dem Tree ins Projekt zu übernehmen. Das Kontextmenü wird nur angezeigt, wenn eine Übernahme ins Projekt möglich ist. Es wird entweder ein Insert angeboten oder, bei unterschiedlichen Modulen, ein Replace. Die Übernahme ins Projekt erfolgt im Online-Betrieb. Der Online-Dialog wird bei der Übernahme geschlossen, da das Projekt verändert wird.



Außerdem wird im Hardware Tree des XN Safety Designer zu einem bestimmten Safety-Modul die eindeutige Modulsicherheitsnummer im Property Browser angezeigt. Diese kann ebenfalls vom Anwender mit der physikalisch vorhandenen Hardware überprüft werden.

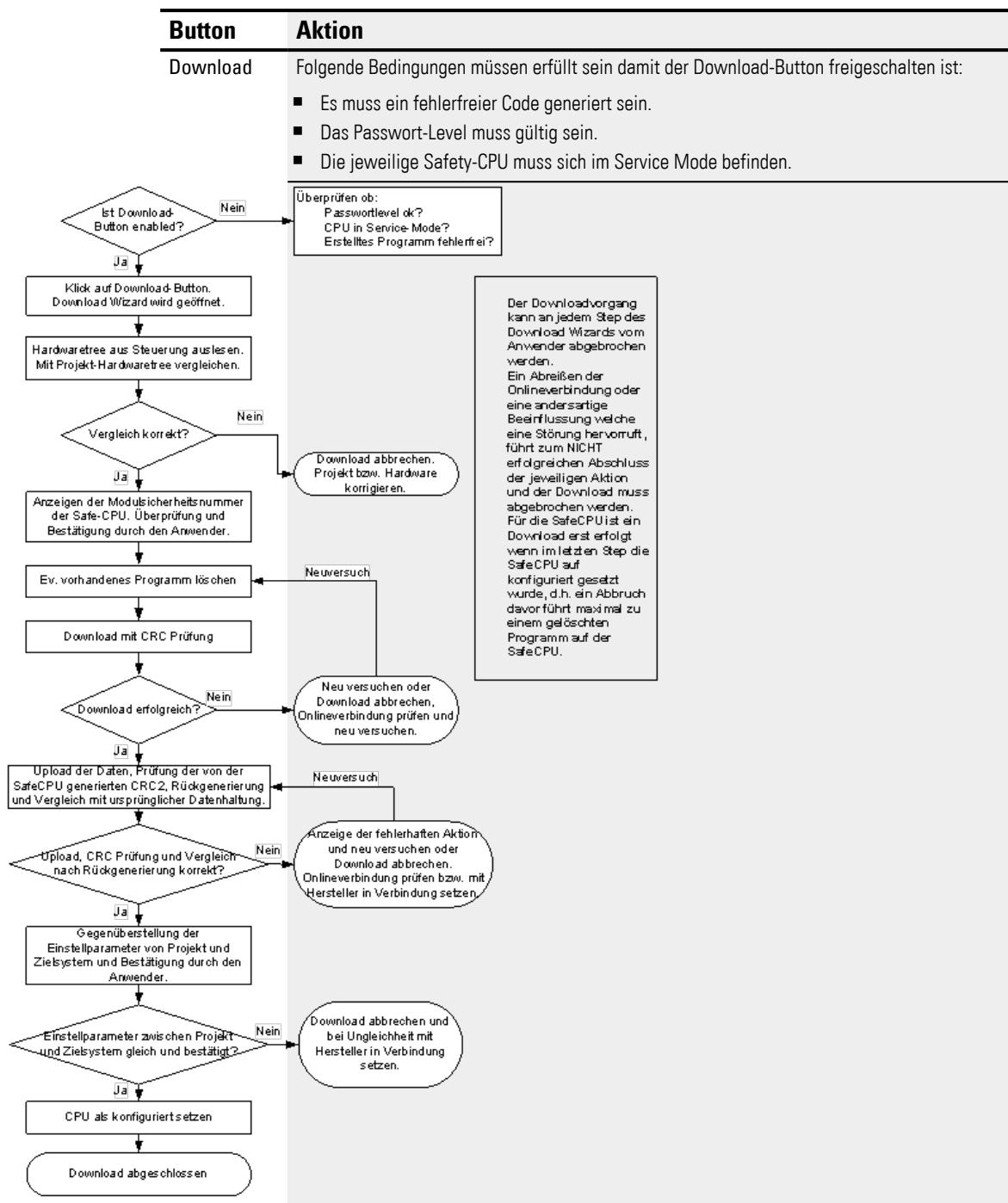
11.2 Online Aktionen

Die nachfolgend beschriebenen Online-Aktionen benötigen, mit Ausnahme von „Quit Error“, „Service Mode“ und „Start“, alle das Password level „Configuration“. „Quit Error“ ist auch ohne Password level verfügbar, „Service Mode“ und „Start“ ist bereits ab Level "Debug" verfügbar. Für eine Übersicht der möglichen Aktionen für den jeweiligen Password level siehe → " Passwörter", Seite 157.

Button	Aktion
Reset/Start PLC	Der Anwender kann über einen Button in der Debug-Toolbar auswählen, ob ein Reset/Start der PLC ausgeführt werden soll. Bei gedrücktem Button wird vor einem Download der PLC in den Reset gestellt. Ebenso wird vor einem Start der Safety-Applikation der PLC in den Reset gestellt und wieder gestartet. Der Status des Buttons wird in der Registry gespeichert.
Quit Error	Ein anliegender Fehler auf der Safety-CPU oder auf Modulen, welche von der Safety-CPU verwendet werden, wird bestätigt. Anliegende Fehler in Slave Safety-CPU's müssen über einen Quit Error bei der Master Safety-CPU bestätigt werden. Nach der Ausführung des Quit-Error-Befehls wird man wieder automatisch auf der Safety-CPU angemeldet, auf welcher man vor dem Befehl angemeldet war.
Service Mode	Die Safety-CPU wechselt in den Zustand Service Mode. Nur in diesem Zustand kann ein Download, Clear, Set Verified, Firmware Update, ... durchgeführt werden.
Start	Der Service Mode auf der Safety-CPU wird verlassen. Die Safety-CPU geht je nach Systemzustand in den Running Mode, temporären Running Mode, Error Mode oder in den Idle-Zustand.
Clear Program	Das Programm auf der Safety-CPU wird gelöscht. Ein Löschen der Konfiguration bei Slave-Modulen ist über den XN Safety Designer nicht möglich. Zum Löschen der Konfiguration muss das Modul entweder in den Master-Modus gesetzt oder als Stand-Alone-CPU betrieben werden.
Set Verified	Die Überprüfung des Programms in der Safety-CPU durch den Anwender wird damit positiv abgeschlossen. Die Bestätigung muss vom Anwender doppelt durchgeführt werden. Die Safety-CPU wechselt mit dieser Aktion von der zeitlich begrenzten Abarbeitung in die zeitlich unbegrenzte Abarbeitung. Bei Safety-CPU's wird bei diesem Prozess auch überprüft, welche optionalen Module angeschlossen sind. Bei fehlenden optionalen Modulen wird eine Warnung angezeigt.  Der Dialog informiert den Benutzer, dass ein nachträgliches Anschließen optionaler Module während des Betriebs einen Fehler auslösen kann.

11. Online-Aktionen und Download

11.2 Online Aktionen



Button	Aktion
	Der Anwender muss alle Parameter des Programms (z.B. Filterzeit,...), die nicht einfach über die Netzwerkansicht ersichtlich sind, nach dem Upload bestätigen. Dem Anwender werden die Parameter in einem gesonderten Dialog angezeigt. Es erfolgt eine Gegenüberstellung der hochgeladenen Werte mit den Werten des Projekts. Der Anwender muss durch das Setzen einer Checkbox die Richtigkeit der Werte bestätigen. Nur, wenn die Richtigkeit der Werte vom Anwender bestätigt wurde, kann die Safety-CPU in den konfigurierten Zustand gesetzt werden. Die Bestätigung kann entfallen, wenn der Anwender im „Developer Mode“ (siehe Kapitel Button "Login") eingeloggt ist. Im „Developer Mode“ wird ein Download ohne den Wizard und somit ohne die Einzelbestätigungen durch den Anwender durchgeführt. Die Steuerung kann bei eingestelltem Entwicklermodus nur im temporären Operation Mode laufen. Um über den „Set Verified“ Button in den zeitlich unbegrenzten Operation Mode zu kommen ist ein Download mit Bestätigung der einzelnen Parameter notwendig.
Login	Siehe auch Kapitel.→ " Passwörter", Seite 157 Anmeldung mittels Passwort an einer Safety-CPU. Wenn noch kein Passwort vergeben ist, muss erst ein Passwort vergeben werden (über Passwort ändern). Wenn ein entsprechendes Passwort auf der Safety-CPU vorhanden ist, wird die Gültigkeit des Login überprüft. Der Anwender kann beim Einloggen im Level „Configuration“ die Option "Developer Mode" setzen, um während der Entwicklung einen schnelleren Download der Daten zu ermöglichen. Bei aktiviertem Entwicklermodus müssen die vorhandenen Download-Schritte nicht einzeln bestätigt werden, der Betriebsmodus der Safety-CPU bleibt aber im zeitlich begrenzten Operation Mode. Das heißt der Button „Set Verified“ ist nicht verfügbar, wenn der Anwender einen Download mit aktiviertem Entwicklermode durchgeführt hat, ein Wechsel in den zeitlich unbegrenzten Operation Mode wird verhindert. Das Einloggen im Level „Configuration“, um eine Programmänderung vorzunehmen, ist jeweils immer nur für eine Safety-CPU möglich, d.h. es wird vom XN Safety Designer verhindert, sich bei mehreren Safety-CPU gleichzeitig im Level „Configuration“ einzuloggen. Der Anmeldezustand auf einer Safety-CPU wird für die Dauer einer Online-Session zwischengemerkt; dies bedeutet, dass man sich nur einmalig Anmelden muss; danach werden Anmeldungen automatisch durch den XN Safety Designer durchgeführt (bei der Verwendung der General Functions).
Change Password	Bei einer Passwortänderung muss das alte Passwort eingegeben werden. Jedes neue Passwort muss durch doppelte Eingabe bestätigt werden. Bei der ersten Inbetriebnahme ist für die Safety-CPU noch kein Passwort vergeben (Auslieferungszustand der Safety-CPU ohne Passwort). Der Anwender muss ein Passwort auf der Safety-CPU setzen. Das Eingabefeld des alten Passwortes bleibt leer, die Eingabefelder für das neue Passwort und für die Bestätigung des neuen Passwortes müssen ausgefüllt werden.

11.3 Download Wizard

Der Download Wizard erscheint, sobald ein Projekt auf die CPU gespielt werden soll. Dieser ist in 2 Schritte unterteilt.

11.3.1 Schritt 1: Hardware Tree vergleichen und Kontrolle der Safety-Nummer des Moduls.

Hier vergleicht der XN Safety Designer die angegebenen Module in Ihrem Projekt mit den tatsächlich an der Steuerung vorhandenen Modulen. Falls ein von der Safety-CPU benötigtes Modul nicht vorhanden ist, wird eine Warnung generiert; ein Download auf die Safety-CPU ist möglich, da hierfür nur die Safety-CPU benötigt wird. Mit

11. Online-Aktionen und Download

11.3 Download Wizard

dem Button **Confirm Safety Numbers** muss der Anwender bestätigen, dass die angezeigten Sicherheitsnummern korrekt sind. Die Sicherheitsnummern der Safety-CPU's werden in einer Liste angezeigt. Die angezeigten Nummern müssen mit den Sicherheitsnummern auf den Modulen verglichen werden.

SCPU	Safety Number
XS_102_C00_000	0x75568

Hardware tree comparison successful.

CPU Progress

☐ Write file with download informations for every Safe CPU

Confirm Safety Numbers Download

Set Configured and Finish Cancel Help

11.3.2 Schritt 2: Programm auf Safety-CPU löschen, neues Projekt downloaden, Code von der Safety-CPU hochladen und mit dem Projekt vergleichen.

Sobald die Sicherheitsnummern vom Anwender bestätigt wurden, wird der Button **Download All** freigeschaltet.

Mit diesem Button **Download All** wird das alte Projekt gelöscht und das neue Projekt auf die Safety-CPU gespielt.

Nach einem erfolgreichen Download, wird automatisch der Code hochgeladen (Upload) und mit dem Projekt verglichen.

Nach einem erfolgreichen Upload wird der Dialog **Check Parameter** geöffnet. Es müssen die Parameter geprüft und deren Richtigkeit bestätigt werden.

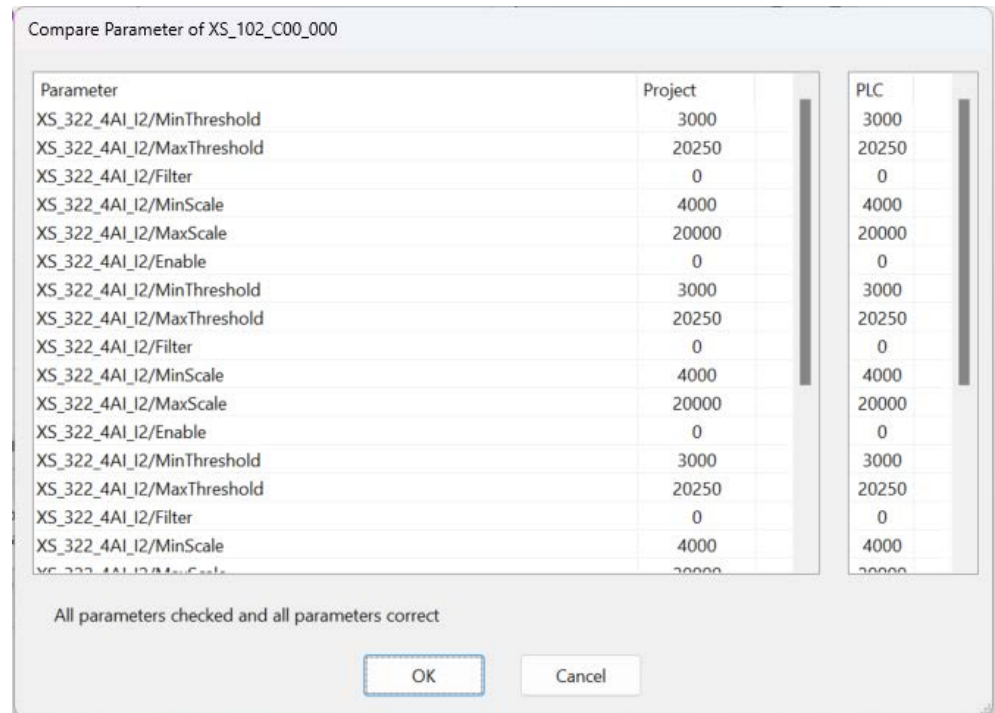
11. Online-Aktionen und Download

11.3 Download Wizard

Diese Arbeitsabfolge wird automatisch für alle aufgelistete Safety-CPU wiederholt.

Die Fortschrittsbalken zeigen zum einem den Ablauf der jeweiligen CPU und zum anderem den Gesamttablauf an.

Mit dem Button **Set Configured and Finish** wird die Safety-CPU auf "konfiguriert" gesetzt und der Downloadvorgang abgeschlossen.



11. Online-Aktionen und Download

11.4 Passwörter

11.4 Passwörter

- **Level 0 (kein Passwort)**
Ohne Passwort können nur lesende Zugriffe durchgeführt werden.
- **Level 1:**
Mit PW1 kann im Debug Modus (Force Modus) gearbeitet werden.
- **Level 2:**
Mit PW2 kann die Konfiguration der Safety Module geändert werden.
- **Level 3 (Herstellerepasswort):**
Mit PW3 kann ein Firmwareupdate durchgeführt werden.

Die Passwörter werden von der Safety-CPU verglichen, ausgewertet und gespeichert. Dadurch wird verhindert, dass mit einem Passwort eine Änderung an jeder Maschine möglich ist.

Das Passwort kann nur durch vorherige Eingabe des alten Passworts geändert werden. Ein neues Passwort muss zweimal eingegeben werden (um Tippfehler zu vermeiden) und muss zwischen 6 und 8 Zeichen lang sein. Dabei ist es unerheblich, in welchem Level man eingeloggt ist, die Änderung ist durch die Eingabe des alten Passworts gesichert.

Nach einer Stunde ohne Aktion schaltet der XN Safety Designer automatisch auf Ebene 0 zurück.

11.4.1 Übersicht über Aktionen abhängig vom Password level

grün..... Aktion möglich

rot..... Aktion nicht möglich

grau..... Kombination macht keinen Sinn

Aktion	Offline	Online				
	Offline	Kein PW Level 0	Level 1 Debuggen	Level 2 Configuration	Level 2 Configuration (Dev.-Mode)	Level 3 Herstellerepasswort
Programmänderung						
Debuggen						
Debuggen (nur Anzeige)						
Debuggen (Force-Modus)			nur in temp. OpMode	nur in temp. OpMode	nur in temp. OpMode	nur in temp. OpMode
Aktionen im Online Dialog						
Login						
Change Password						
Quit Error						
Service Mode			wenn in Run Mode	wenn in Run Mode	wenn in Run Mode	wenn in Run Mode
Start			wenn in Service Mode	wenn in Service Mode	wenn in Service Mode	wenn in Service Mode
Clear Program				wenn in Service Mode	wenn in Service Mode	wenn in Service Mode
Download				wenn in Service Mode	wenn in Service Mode	wenn in Service Mode
Write SD Card				wenn in Service Mode		wenn in Service Mode
Set Verified				wenn in Service Mode		wenn in Service Mode

11.5 Upload Log-Data

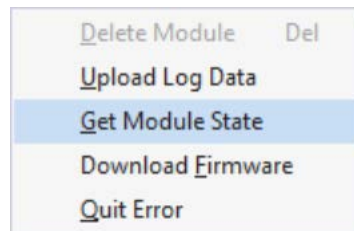
Über das Kontextmenü im Hardware Tree (jeweiliges Modul) werden die Daten des Log-Speichers des Moduls ausgelesen, in eine Binärdatei gespeichert und in einem Tab angezeigt. Diese Daten dienen zu Servicezwecken. Siehe dazu auch Kapitel Log-Dateien anzeigen.

11. Online-Aktionen und Download

11.6 Get Module State

11.6 Get Module State

Über das Kontextmenü im Hardware Tree (jeweiliges Modul) werden die Status-Daten des Moduls ausgelesen. Die Daten werden im Ausgabefenster dargestellt und können auch gespeichert werden.



Ausgegeben wird:

- Laufzeitstatus (z.B. Operation Mode)
- Fehleranzeigen (wenn vorhanden)
- Konfigurationsstatus (z.B. Verifiziert)
- Sicherheitsnummer des Moduls
- Firmwareversion
- Zykluszeit, max. Zykluszeit und eingestellte max. Zykluszeit in [ms]
- Belegter Speicher im Flash in [byte]
- Gesamtspeicher im Flash in [byte]
- Belegter Speicher im RAM in [byte]
- Gesamtspeicher im RAM in [byte]

Für die beiden Controller werden folgende Daten getrennt angezeigt:

- Abarbeitungszeit der Applikation in [ms]
- Max. Abarbeitungszeit der Applikation in [ms]
- Zeit, welche die Tests benötigen in [ms]
- Max. Zeit, welche die Tests benötigen in [ms]
- Belegter Speicher im RAM in [byte]
- Größe des Betriebssystem-Images in [byte]

12. Export der Download-Daten für eine Safety-CPU

Die Download-Daten einer Safety-CPU können in eine Binärdatei exportiert werden.

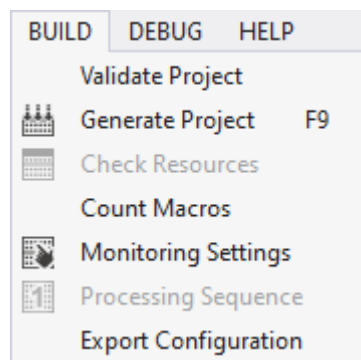
Aufbau der Binärdatei:

- Exportheader
- Konfiguration (identisch zur Konfiguration beim Download über den XN Safety Designer)
- eine zusätzliche CRC am Ende

Diese Binärdatei kann dann ohne den XN Safety Designer in eine Safety-CPU eingespielt werden.

12.1 Erstellen der Binärdatei

Der Export kann über das Menü **Build** mit **Export Configuration** aufgerufen werden.



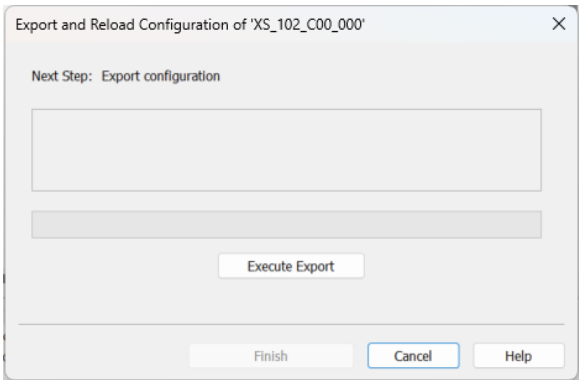
Das Projekt muss kompilierbar sein, um einen Export durchzuführen. Der Export läuft in mehreren Schritten über einen Wizard ab.

Mit dem Button **Execute Export** wird zuerst eine Binärdatei exportiert (Name der Binärdatei setzt sich aus Projektnamen und Safety-CPU Namen zusammen, z.B. Project-<Name>-Safety-CPU-<Name>.bin).

Der Export erfolgt in das Projektverzeichnis. Nach dem Schreiben der Datei wird automatisch ein externes Tool gestartet (CheckConfig.exe), welches die Binärdatei lädt und die CRC der Konfiguration auf Gültigkeit prüft. Das externe Tool ersetzt die geprüfte CRC durch eine zweite CRC und speichert die Binärdatei unter einem anderen Namen wieder ab. Bei einer fehlerhaften Prüfung wird ein Fehler ausgegeben und die Binärdatei nicht erneut gespeichert.

12. Export der Download-Daten für eine Safety-CPU

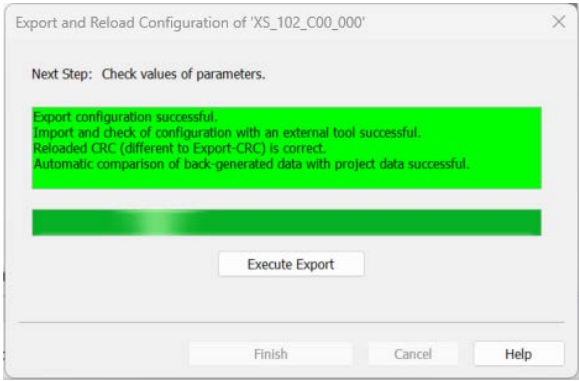
12.1 Erstellen der Binärdatei



12.1.1 Fehler bei der Prüfung im externen Tool:

Fehler	Beschreibung
External tool 'CheckConfig.exe' not found	Das externe Tool konnte nicht gestartet werden. Das Tool muss im Installationsverzeichnis liegen.
Import file <Name> failed	Die Binärdatei konnte nicht geladen werden.
CRC is incorrect	Die CRC-Prüfung ist fehlgeschlagen.
Export file <Name> failed	Die neue Binärdatei konnte nicht geschrieben werden

Es folgt ein „Reload and Compare“. Dabei wird die vom externen Tool erstellte Binärdatei eingelesen und die neue CRC auf Gültigkeit geprüft; zusätzlich wird die rückgelesene Konfiguration mit dem Projekt verglichen.



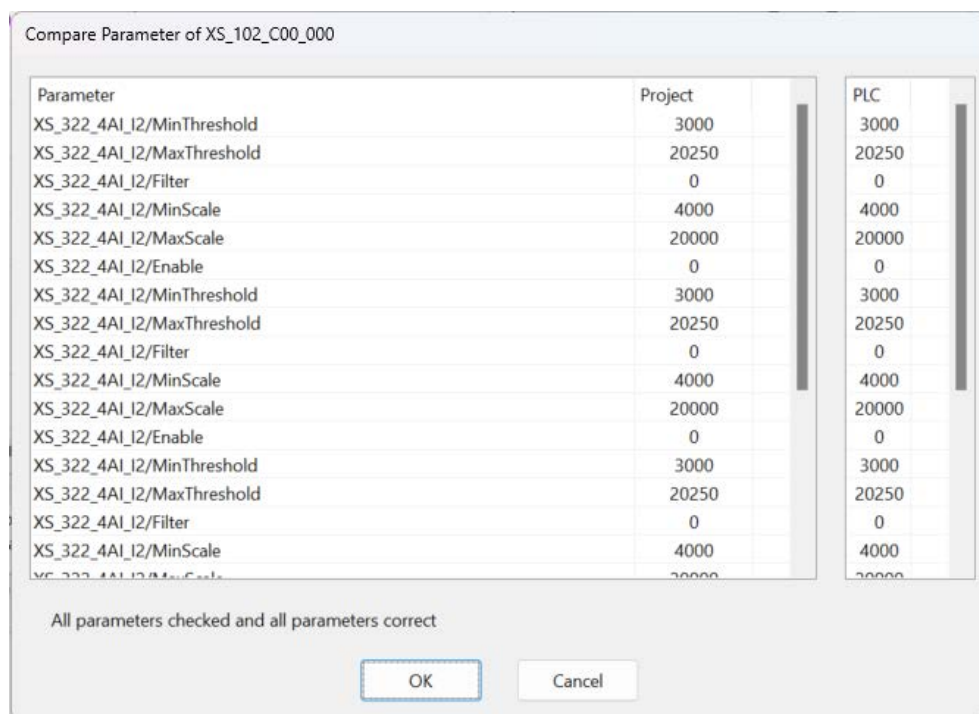
12. Export der Download-Daten für eine Safety-CPU

12.1 Erstellen der Binärdatei

12.1.2 Fehler beim Reload and Compare:

Fehler	Beschreibung
Reloaded CRC (different to Export-CRC) is incorrect	Die vom externen Tool generierte CRC ist nicht korrekt.
Automatic comparison of back-generated data with project data failed	Die rückgewandelten Daten sind unterschiedlich zum Projekt.

Nach dem erfolgreichen Vorgang, öffnet sich der Check Parameter Dialog. Die Parameter müssen vom Anwender geprüft werden und deren Richtigkeit bestätigt werden.



Mit dem Button „Finish“ wird die Binärdatei wiederum gespeichert (Name ist identisch zum ersten Namen). Die vom externen Tool erzeugte CRC wird durch die ursprüngliche CRC ersetzt. Die vom externen Tool erzeugte CRC wird ans Ende der Binärdatei geschrieben. Die Binärdatei liegt im Projektverzeichnis.

12. Export der Download-Daten für eine Safety-CPU

12.2 Binärdatei auf eine Safety-CPU spielen

12.2 Binärdatei auf eine Safety-CPU spielen

Der Anwender muss für den Download auf eine Safety-CPU zwei Schritte durchführen.

Schritt 1: Auswertung und Prüfung des Exportheaders der Binärdatei

Aufbau des Exportheaders:

- CRC des Headers (4 Byte)
- Länge des Headers (4 Byte)
- Typ des Headers (4 Byte), der Typ ist mit 0 belegt
- Länge des Hardware-Pfads (4 Byte)
- Hardware-Pfad (Pfad ist auf 4 Byte ausgerichtet, Länge des Pfads ohne 4 Byte-Ausrichtung steht am Anfang des Pfades)
- Länge des Projektnamens (4 Byte)
- Projektname
- Länge der Revisionsnummer des Projekts (4 Byte)
- Revisionsnummer des Projekts
- Länge des Namens der Safety-CPU (4 Byte)
- Safety-CPU Name

Die Informationen im Exportheader (Projektnamen, Revision, ...) müssen vom Anwender auf Richtigkeit geprüft werden.

Schritt 2: Download der Binärdatei

Der Anwender muss sich auf der Safety-CPU einloggen und kann den Download der Konfiguration (Binärdatei ohne den Exportheader) ausführen. Nach dem erfolgreichen Download muss der Anwender den Download bestätigen sowie die Safety-CPU starten und verifizieren.

13. Funktionsblöcke

13.1 Datentypen

Prinzipiell wird zwischen Safe- und Unsafe-Datentypen unterschieden. Um die Komplexität niedrig zu halten, gibt es nur folgende Datentypen:

1. SAFEBOOL (4 Byte Wert)
2. SAFEDINT (8 Byte Wert)
3. DINT (4 Byte Wert)

Wie bei der Liste der Datentypen ersichtlich, werden auch boolsche Werte als 4 Byte übertragen. Dabei wird der Wert 0 als `False` interpretiert, alle anderen Werte ungleich 0 werden als `True` interpretiert.

13. Funktionsblöcke

13.2 Besonderheiten bei der Implementierung

13.2 Besonderheiten bei der Implementierung

- BOOL (nicht SAFE_BOOL) Werte werden nur auf FALSE bzw. ungleich FALSE abgefragt.
- Wird bei einem Funktionsblock eine bestimmte Zeit überwacht, die dynamisch eingestellt werden kann und kein Error State (Complex Function Blocks) definiert ist, so darf diese Einstellung einen 31 Bit (also 0x7FFFFFFF) Wert nicht überschreiten. Ist die eingestellte Zeit höher, wird ein Fehler (ERR_BAD_INPUT_VALUE) an die Firmware des Safe-Moduls zurückgeliefert.
- Bei INIT wird der FB initialisiert. Die SAFEBOOL-Ausgänge werden auf SAFEBOOL_FALSE gesetzt. Die BOOL-Ausgänge werden auf FALSE gesetzt. Die Flankenmerker werden mit dem Wert des dazugehörigen Eingangs initialisiert. Somit wird beim ersten Durchlauf keine Flanke erkannt. Es wird auch ein möglicher ungültiger Wert auf den Flankenmerker geschrieben. D.h. die SAFEBOOL-Eingänge werden nicht überprüft. Bei den Complex Function Blocks wird der DiagCode auf 0x0000 gesetzt. Nach der Initialisierung wird der FB verlassen.
- Am Ende des FBs wird der Eingangswert auf den Flankenmerker geschrieben.
- Ist ein SAFEBOOL Eingang undefiniert (weder True, False oder Error) werden alle SAFEBOOL-Ausgänge auf SAFEBOOL_ERROR gesetzt und es wird ein Fehler-Code an die Firmware des Safe-Moduls zurückgeliefert. Dies gilt auch für den eventuell beim Funktionsblock vorhandenen internen Speicher zur Flankenerkennung. Bei den Complex FBs bleiben die Ausgänge 'READY', 'ERROR' und 'DiagCode' auf den bereits gesetzten Wert.
- Liegt bei einem SAFEBOOL Eingang SAFEBOOL_ERROR an, werden alle SAFEBOOL-Ausgänge auf SAFEBOOL_ERROR gesetzt. Zusätzlich wird bei Complex Function Blöcken der Ausgang DiagCode auf State 0000 gesetzt. Somit wird erreicht, dass der FB neu anlaufen muss, wenn am Eingang der SAFE-BOOL_ERROR wieder auf SAFEBOOL_TRUE oder SAFEBOOL_FALSE geht und die jeweiligen Bedingungen der FBs wieder erfüllt sein müssen, um in den State zu gelangen, in dem der SAFEBOOL_ERROR aufgetreten ist.
- Bei SAFEDINT Eingängen kann ein Wert oder ein SAFEDINT_ERROR anliegen. Liegt ein SAFE-DINT_ERROR an, wird am Ausgang ein SAFEDINT_ERROR angezeigt.
- Ist ein SAFEDINT ungültig, werden die Ausgänge auf SAFEDINT_ERROR gesetzt und es wird ein Fehler-Code an die Firmware des Safety-Moduls gesendet. Die Safety-CPU kann dadurch einen Fehlerzustand anzeigen. Der Fehler muss zurückgesetzt werden.
- Die Anzeige der Fehlerwerte im Onlinebetrieb erfolgt durch eine s_error bei den Instanzen der Funktionsblöcke.
- Ist der Eingang eines Funktionsblocks konstant (Einstellung Constant Value im Property Browser) kann keine unsichere Variable angeschlossen werden. Unsichere

Variablen können indirekt über die Ausgänge von Funktionsblöcken angeschlossen werden, der Anwender muss hier die Korrektheit der Verschaltung verifizieren.

13.3 Zeitüberwachung bei Funktionsblöcken

Die Zeiten, die bei den Funktionsblöcken eingestellt werden können, sind konstant und können nur mittels Safety Designer konfiguriert werden. Die Werte können während des Betriebs nicht geändert werden.

Die erlaubten Zeiten sind bei den Beschreibungen der einzelnen Funktionsblöcke angeführt; kann für Funktionsblöcke der maximal gültige Bereich eingestellt werden, sind die Zeiten nicht explizit angegeben.

Für die nachfolgenden Funktionsblöcke gilt der maximale Wertebereich der Zeitüberwachung, der bei einem 4 Byte-Wert von 0-0x7FFFFFFF ist:

- TON
- TOFF
- TP
- SF_Equivalent-Funktion
- SF_Antivalent-Funktion
- SF_ModeSelector-Funktion
- SF_GuardMonitoring-Funktion
- SF_SafetyRequest-Funktion
- SF_EDM-Funktion
- Path Speed
- Speed
- Direction

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.1 Allgemeine Informationen

Die Funktionsblöcke belegen den RAM-Speicher, der in der Spalte RAM-Speicher aufgelistet wird. Einige Funktionsblöcke benötigen einen SRAM-Speicher, da Werte gespeichert werden müssen. Diese Werte stehen dann beim nächsten Start der Safety-CPU zur Verfügung. Der Speicher wird auch von den remanenten Memory-Variablen belegt.

Der SRAM-Speicher beträgt

- XS-102-C00-000: 120 Byte (30 Werte à 4 Byte)

Die Speicherbelegung der Funktionsblöcke wird in der Spalte SRAM-Speicher aufgelistet.

Funktionsblock	RAM-Speicher (Byte)	SRAM-Speicher (Byte)
AND	16	
AND_5I	28	
AND_NS	16	
AND_XI	110	
ABS	20	
ABS_NS	12	
ADD	32	
ADD_NS	24	
AND_BW	28	
Bool_to_Safebool	12	
Collision_Detection	60	
CTD	28	
CTU	28	
CTUD	44	
Dint_to_SafeDint	16	
DINT_Compare	36	
EQ	24	
EQ_NS	16	
F_TRIG	16	
FW_Check	20	
GT	24	
GT_NS	16	
GE	24	
GE_NS	16	
LT	24	
LT_NS	16	

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

Funktionsblock	RAM-Speicher (Byte)	SRAM-Speicher (Byte)
LE	24	
LE_NS	16	
LIMIT	36	
LIMIT_NS	20	
MULDIV	52	
MULDIV_NS	32	
MIN	28	
MIN_NS	16	
MAX	28	
MAX_NS	16	
NOT	12	
NOT_BW	20	
OR_BW	28	
OR	16	
OR_NS	16	
OR_5I	28	
OR_XI	110	
Restart_Interlock	36	
R_TRIG	20	
SUB	32	
SUB_NS	20	
SHL	24	
SHL_NS	16	
SHR	24	
SHR_NS	16	
Safebool_to_Bool	12	
SafeDInt_to_Dint	16	
SError_to_SFalse	12	
SDINT_Compare	40	
Safe_Area	60	
SampleHold	24	
SF_Antivalent	44	
SF_EDM	72	
SF_EmergencyStop	44	
SF_EnableSwitch	48	
SF_Equivalent	44	
SF_Espe	44	
SF_GuardLocking	64	
SF_ModeSelector	128	
SF_MutingPar	108	
SF_MutingPar_2Sensor	80	
SF_MutingSeq	108	
SF_Optional_PWD	64	4
SF_Optional_PWDII	72	4
SF_Optional_Switch	20	
SF_Optional_Switch_SDINT	32	

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

Funktionsblock	RAM-Speicher (Byte)	SRAM-Speicher (Byte)
SF_Optional_Switch_DINT	20	
SF_OutControl	56	
SF_SafetyRequest	56	
SF_TestableSafetySensor	84	
SF_TwoHandControlTypell	32	
SF_TwoHandControlTypelll	40	
SF_LimitComparator	68	
SF_LimitComparator_Dint	20	
SF_NumericSelector	132	
SF_RampMonitor	80	
SF_DirectionMonitor	68	
SF_SkewMonitor	92	8
SF_HGW_Login	104	4
TOFF	32	
TON	32	
TP	32	
XOR_BW	28	
XOR	16	

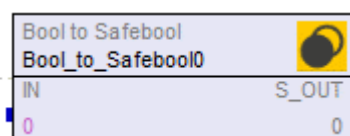
13.4.2 Funktionsblöcke zur Typkonvertierung

Wenn nicht anders angegeben steht jeder Funktionsblock als SAFE Variante zur Verfügung. Reine UNSAFE-Funktionsblöcke sind nicht vorgesehen.

13.4.2.1 BOOL_TO_SAFEBOOL

Konvertierung eines BOOL-Wertes in einen SAFEBOOL-Wert.

- Beim Wandeln wird der Eingang „In“ auf den Wert des SAFEBOOL-Wertes gesetzt.
- Liegt kein gültiger Wert am Eingang an, wird SAFEBOOL_ERROR aus-gegeben.



Eingangsvariablen

IN Eingang (BOOL)

Ausgangsvariablen

S_OUT Ausgang (SAFEBOOL)

13.4.2.2 DINT_TO_SAFEDINT

Konvertierung eines DINT-Wertes in einen SAFEDINT-Wert. Ein SAFEDINT-Wert besteht aus einem DINT-Wert und einem Wert, welche angibt, ob der Wert gültig ist.

- Beim Wandeln wird der Eingang „In“ auf den Wert des SAFEDINT-Wertes gesetzt; der zusätzliche Wert wird auf den Zustand gültig gesetzt.



Eingangsvariablen

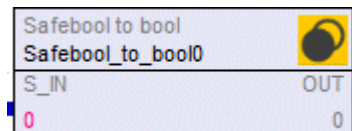
IN Eingang (DINT)

Ausgangsvariablen

S_OUT Ausgang (SAFEDINT)

13.4.2.3 SAFEBOOL_TO_BOOL

Konvertierung eines SAFEBOOL-Wertes in einen BOOL-Wert.



Eingangsvariablen

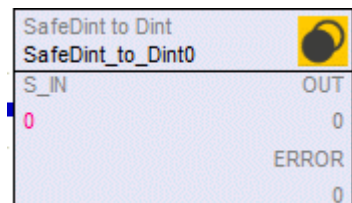
S_IN Eingang (SAFEBOOL)

Ausgangsvariablen

OUT Ausgang (BOOL)

13.4.2.4 SAFEDINT_TO_DINT

Konvertierung eines SAFEDINT-Wertes in einen DINT-Wert.



Eingangsvariablen	S_IN	Eingang (SAFEDINT)
Ausgangsvariablen	OUT	Ausgang (DINT)
	ERROR	TRUE, wenn Eingang ungültig (BOOL)

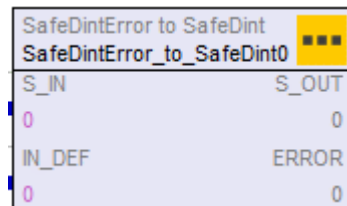
13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.2.5 SAFEDINTERROR_TO_SAFEDINT

Umwandlung von SAFEDINT_ERROR-Zuständen in einen Default-Wert.

- Bei einem Wert SAFEDINT_ERROR am Eingang „S_IN“ wird der Wert am Eingang „IN_DEF“ (DINT) als SAFE-DINT-Wert am Ausgang „S_OUT“ ausgegeben, ansonsten wird der Wert am Eingang „S_IN“ durchgeschaltet.



Eingangsvariablen

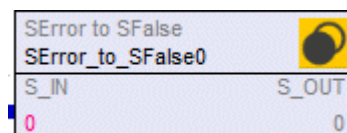
S_IN	Eingang (SAFEDINT)
IN_DEF	Ausgabewert für den Fall von SAFEDINT_ERROR an S_IN (DINT)

Ausgangsvariablen

S_OUT	Ausgang (SAFEDINT)
ERROR	Flag für SAFEDINT_ERROR an S_IN (BOOL)

13.4.2.6 ERROR_TO_SFALSE

Umwandlung von SAFEBOOL_ERROR - Zuständen in SAFEBOOL_FALSE. Ausgang S_Out wird nur SAFE-BOOL_TRUE, wenn auch der Eingang S_IN SAFEBOOL_TRUE ist, sonst ist S_OUT immer SAFE-BOOL_FALSE.



Eingangsvariablen

S_IN	Eingang (SAFEBOOL)
------	--------------------

Ausgangsvariablen

S_OUT	Ausgang (SAFEBOOL)
-------	--------------------

13.4.3 Numerische Standard-Funktionsblöcke

Tab. 3: Übersicht

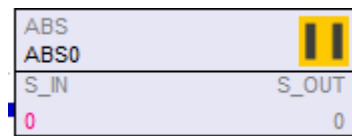
→ "ABS (Absolutwert)", Seite 173 → "ABS_NS (Absolutwert, not safe)", Seite 173		Absolutbetrag
→ "ADD (Addition)", Seite 173 → "ADD_NS (Addition, not safe)", Seite 174		Addition
→ "SUB (Subtraktion)", Seite 174 → "SUB_NS (Subtraktion, not safe)", Seite 175		Subtraktion
→ "MULDIV (Multiplikation und Division)", Seite 176 → "MULDIV_NS (Multiplikation und Division, not safe)", Seite 175		Multiplikation und Division
→ "SF_SampleHold", Seite 176		Umschalten bei Haltefunktion
→ "SHL (Shift-Left)", Seite 177 → "SHL_NS (Shift-Left, not safe)", Seite 177		Shift-Left eines SAFEDINT- oder DINT-Eingangs
→ "SHR (Shift-Right)", Seite 178 → "SHR_NS (Shift-Right, not safe)", Seite 178		Shift-Right eines SAFEDINT- oder DINT-Eingangs
→ "AND_BW (UND, bitweise)", Seite 179		Bitweise UND-Verknüpfung mit SAFEDINT-Eingängen
→ "OR_BW (ODER, bitweise)", Seite 179		Bitweise ODER-Verknüpfung mit SAFEDINT-Eingängen
→ "XOR_BW (XOR, bitweise)", Seite 180		Bitweise XOR-Verknüpfung mit SAFEDINT-Eingängen
→ "NOT_BW (NOT, bitweise)", Seite 180		Bitweise Umkehrung eines SAFEDINT-Eingangs
→ "Restart_Interlock", Seite 181		Der Restart_Interlock FUB prüft einen Eingang von analogen Eingangskarten auf seine Richtigkeit
→ "FW_Check", Seite 182		Der FUB prüft die Online FW-Version mit der des Eingangs
→ "Path Speed", Seite 183		Berechnung der Geschwindigkeit in einem 3 dimensionalen Raum
→ "Speed", Seite 185		Berechnung der Geschwindigkeit aus einer Positionsänderung
→ "Direction", Seite 187		Berechnung der Bewegungsrichtung aus der Positionsänderung

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.1 ABS (Absolutwert)

Vorzeichenloser Wert wird retourniert.



Eingangsvariablen

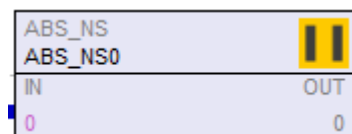
S_IN	Eingang (SAFEDINT)
------	--------------------

Ausgangsvariablen

S_OUT	Ausgang (SAFEDINT)
-------	--------------------

13.4.3.2 ABS_NS (Absolutwert, not safe)

Wert mit positivem Vorzeichen wird retourniert.



Eingangsvariablen

IN	Eingang (DINT)
----	----------------

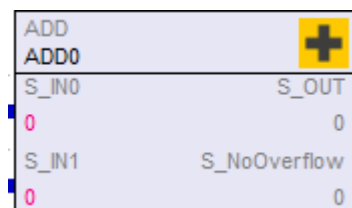
Ausgangsvariablen

OUT	Ausgang (DINT)
-----	----------------

13.4.3.3 ADD (Addition)

Addiert S_IN0 und S_IN1.

Übertrag wird nicht behandelt, bei einem Überlauf wird wieder mit 1 begonnen.



Eingangsvariablen

S_IN0	Eingang 1 (SAFEDINT)
-------	----------------------

S_IN1	Eingang 2 (SAFEDINT)
-------	----------------------

Ausgangsvariablen

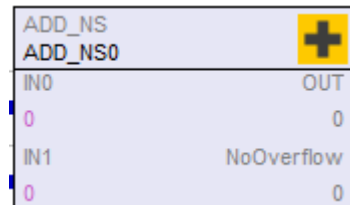
S_OUT	Ausgang (SAFEDINT)
-------	--------------------

S_NoOverflow	Überlauf hat nicht stattgefunden (SAFEBOOL)
--------------	---

13.4.3.4 ADD_NS (Addition, not safe)

Addiert IN0 und IN1.

Übertrag wird nicht behandelt, bei einem Überlauf wird wieder mit 1 begonnen.



Eingangsvariablen

IN0	Eingang 1 (DINT)
IN1	Eingang 2 (DINT)

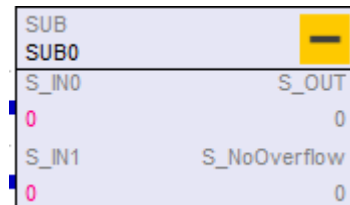
Ausgangsvariablen

OUT	Ausgang (DINT)
NoOverflow	Überlauf hat nicht stattgefunden (BOOL)

13.4.3.5 SUB (Subtraktion)

Subtrahiert S_IN1 von S_IN0.

Übertrag wird nicht behandelt, bei einem Überlauf wird wieder mit 1 begonnen.



Eingangsvariablen

S_IN0	Eingang 1 (SAFEDINT)
S_IN1	Eingang 2 (SAFEDINT)

Ausgangsvariablen

S_OUT	Ausgang S_IN0 - S_IN1 (SAFEDINT)
S_NoOverflow	Überlauf hat nicht stattgefunden (SAFEBOOL)

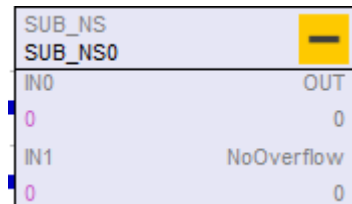
13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.6 SUB_NS (Subtraktion, not safe)

Subtrahiert IN1 von IN0.

Übertrag wird nicht behandelt, bei einem Überlauf wird wieder mit $2^{31}-1$ begonnen.



Eingangsvariablen

IN0 Eingang 1 (DINT)

IN1 Eingang 2 (DINT)

Ausgangsvariablen

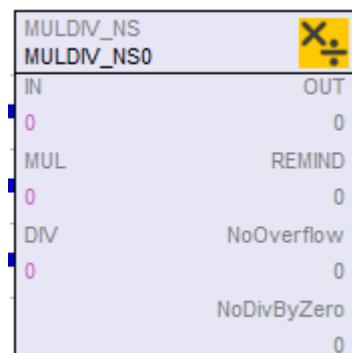
OUT Ausgang $S_IN0 - S_IN1$ (DINT)

NoOverflow Überlauf hat nicht stattgefunden (BOOL)

13.4.3.7 MULDIV_NS (Multiplikation und Division, not safe)

Multipliziert IN mit MUL und dividiert das anschließende Ergebnis durch DIV.

Das Vorzeichen von REMIND ist abhängig vom Ergebnis der Multiplikation. Das Zwischenergebnis der Multiplikation besitzt die doppelte Datenlänge (64 Bit).



Eingangsvariablen

IN Eingangswert (DINT)

MUL Faktor (DINT)

DIV Divisor (DINT)

Ausgangsvariablen

OUT Ausgang $(IN * MUL) / DIV$ (DINT)

REMIND Rest der Division (DINT)

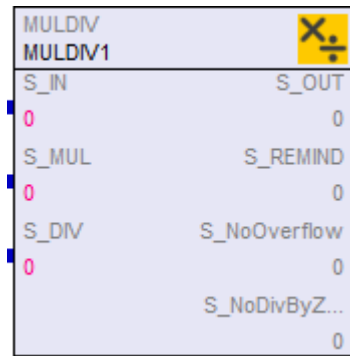
NoOverflow Überlauf hat nicht stattgefunden (BOOL)

NoDivByZero Keine Division durch 0 (BOOL)

13.4.3.8 MULDIV (Multiplikation und Division)

Multipliziert S_IN mit S_MUL und dividiert das anschließende Ergebnis durch S_DIV.

Das Vorzeichen von S_REMIND ist abhängig vom Ergebnis der Multiplikation. Das Zwischenergebnis der Multiplikation besitzt die doppelte Datenlänge (64 Bit).



Eingangsvariablen

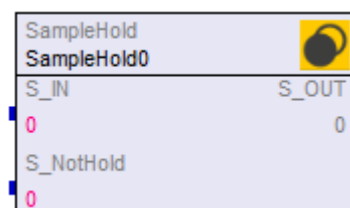
S_IN	Eingangswert (SAFEDINT)
S_MUL	Faktor (SAFEDINT)
D_DIV	Divisor (SAFEDINT)

Ausgangsvariablen

S_OUT	Ausgang (S_IN * S_MUL) / S_DIV (SAFEDINT)
S_REMIND	Rest der Division (SAFEDINT)
S_NoOverflow	Überlauf hat nicht stattgefunden (SAFEBOOL)
S_NoDivByZero	Keine Division durch 0 (SAFEBOOL)

13.4.3.9 SF_SampleHold

Schaltet den S_IN auf S_OUT, wenn S_NotHold gesetzt ist. Wenn S_NotHold nicht gesetzt, wird S_OUT am alten Wert gehalten.



Eingangsvariablen

S_IN	Eingang (SAFEDINT)
S_NotHold	Aktivierung der Haltefunktion. FALSE: Der Wert an S_OUT wird gehalten. TRUE: Der Wert S_IN wird an S_OUT durchgeschaltet. (SAFEBOOL).

Ausgangsvariablen

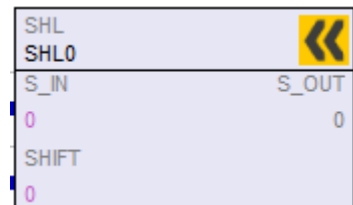
S_OUT	Ausgang (SAFEDINT)
-------	--------------------

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.10 SHL (Shift-Left)

Shift-Left Operation eines SAFEDINT-Werts. Der Eingang wird bitweise um die angegebenen Bit nach links verschoben.



Eingangsvariablen

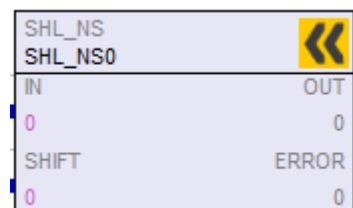
S_IN	Eingang (SAFEDINT)
SHIFT	Anzahl der Bits, um die verschoben wird (DINT)

Ausgangsvariablen

S_OUT	Ergebnis (SAFEDINT)
-------	---------------------

13.4.3.11 SHL_NS (Shift-Left, not safe)

Shift-Left Operation eines DINT-Werts. Der Eingang wird bitweise um die angegebenen Bit nach links verschoben.



Eingangsvariablen

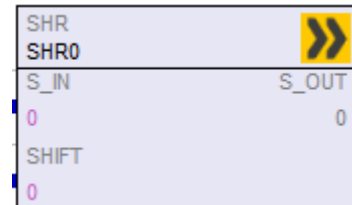
IN	Eingang (DINT)
SHIFT	Anzahl der Bits, um die verschoben wird (DINT)

Ausgangsvariablen

OUT	Ergebnis (DINT)
ERROR	TRUE, wenn Fehler auftreten

13.4.3.12 SHR (Shift-Right)

Shift-Right Operation eines SAFEDINT-Werts. Der Eingang wird bitweise um die angegebenen Bit nach rechts verschoben.



Eingangsvariablen

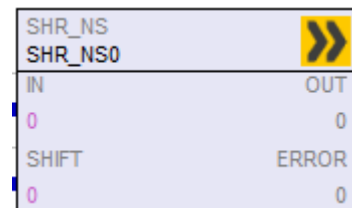
S_IN	Eingang (SAFEDINT)
SHIFT	Anzahl der Bits, um die verschoben wird (DINT)

Ausgangsvariablen

S_OUT	Ergebnis (SAFEDINT)
-------	---------------------

13.4.3.13 SHR_NS (Shift-Right, not safe)

Shift-Right Operation eines DINT-Werts. Der Eingang wird bitweise um die angegebenen Bit nach rechts verschoben.



Eingangsvariablen

IN	Eingang (DINT)
SHIFT	Anzahl der Bits, um die verschoben wird (DINT)

Ausgangsvariablen

OUT	Ergebnis (DINT)
ERROR	TRUE, wenn Fehler auftreten

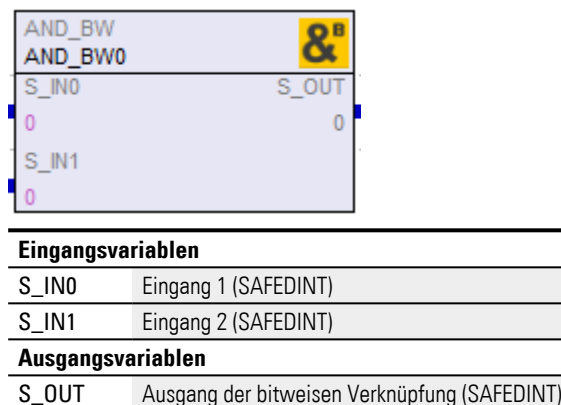
13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.14 AND_BW (UND, bitweise)

Bitweise UND-Verknüpfung von zwei SAFEDINT-Eingängen.

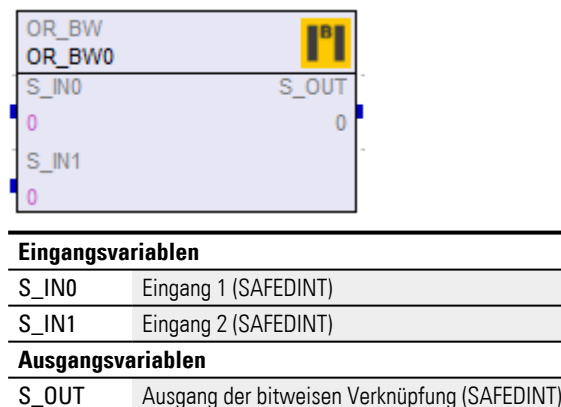
S_IN0 wird bitweise über eine UND-Operation mit S_IN1 verknüpft und das resultierende Bit-Muster wird auf S_OUT ausgegeben.



13.4.3.15 OR_BW (ODER, bitweise)

Bitweise ODER-Verknüpfung von zwei SAFEDINT-Eingängen.

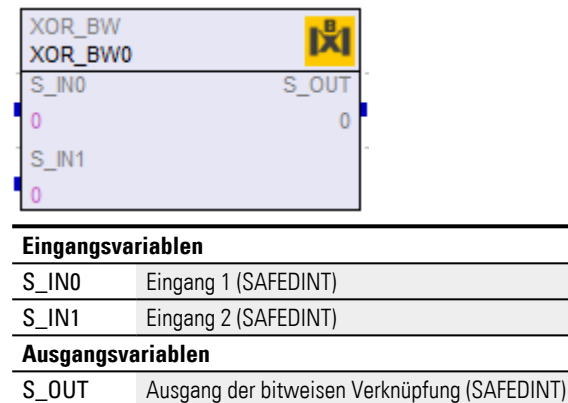
S_IN0 wird bitweise über eine ODER-Operation mit S_IN1 verknüpft und das resultierende Bit-Muster wird auf S_OUT ausgegeben.



13.4.3.16 XOR_BW (XOR, bitweise)

Bitweise XOR-Verknüpfung von zwei SAFEDINT-Eingängen.

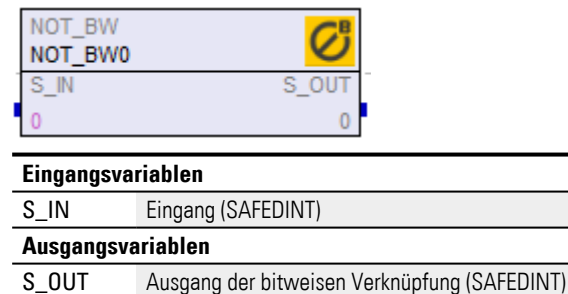
S_IN0 wird bitweise über eine XOR-Operation mit S_IN1 verknüpft und das resultierende Bit-Muster wird auf S_OUT ausgegeben.



13.4.3.17 NOT_BW (NOT, bitweise)

Bitweise Umkehrung eines SAFEDINT-Eingangs.

S_IN wird bitweise invertiert und das resultierende Bit-Muster wird auf S_OUT ausgegeben.



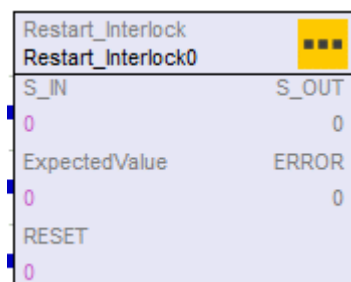
13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.18 Restart_Interlock

Der Restart_Interlock FUB prüft einen Eingang von analogen Eingangskarten auf seine Richtigkeit, wenn der Eingang ExpectedValue auf TRUE steht. Es wird der Wert des Eingangs S_IN auf den Ausgang S_OUT durchgeschaltet. Liegt am Eingang S_IN ein fehlerhafter Wert an, wird dieser am Ausgang S_OUT dauerhaft ausgegeben. Der Ausgang ERROR steht auf TRUE. Es muss über den Eingang RESET ein Rücksetzen des Fehlerfalls ausgelöst werden (Übergang von TRUE auf FALSE).

Ist der Wert des Eingangs ExpectedValue FALSE, wird der Eingang ohne Prüfung auf den Ausgang durchgeschaltet.



Eingangsvariablen

S_IN	Eingang (SAFEDINT)
ExpectedValue	Flag, ob Prüfung aktiv (BOOL)
RESET	Reset Flag (BOOL)

Ausgangsvariablen

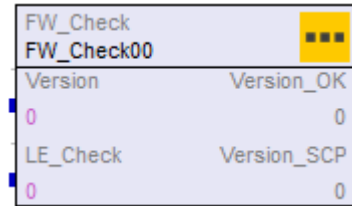
S_OUT	Ergebnis (SAFEDINT)
ERROR	Fehlerfall (BOOL)

13.4.3.19 FW_Check

Der FUB prüft die Online FW-Version mit der des Eingangs. Die FW der Safety-CPU wird in einen nicht sichtbaren Eingang des FUBs kopiert.

Der FUB darf nur ein Mal pro Safety-CPU platziert werden.

Vergleich: Version des Input ist gleich oder kleiner gleich Version auf Safety-CPU.



Eingangsvariablen

Version	Eingang, zu vergleichende Version (DINT)
LE_Check	Flag, für Lower-Equal Vergleich (BOOL) FALSE = Equal Vergleich TRUE = Lower Equal Vergleich

Ausgangsvariablen

Version_OK	Ausgang, Ergebnis des Vergleichs (BOOL)
Version_SCP	Ausgang, welche FW Version auf der Steuerung läuft (DINT)

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.20 Path Speed

Der Baustein PATH_SPEED dient zur Ableitung der Position (X, Y, Z), um daraus die Geschwindigkeit zu erhalten. Er wird somit zur Bestimmung der Geschwindigkeit in einem dreidimensionalen Raum verwendet. Durch die Eingangsvariablen kann das Verhalten des Bausteins konfiguriert werden.

Die Eingangsvariablen S_xPosition, S_yPosition, S_zPosition sind einheitenlos. Es obliegt somit dem Anwender wie er den Positionswert interpretiert. Für die Ausgangsvariable S_Speed kann die Zeiteinheit durch die Eingangsvariable SampleUnit gewählt werden. Die Ausgangsvariable S_Speed gibt die Geschwindigkeit entlang der Bahnkurve an.

Bei sehr kleinen Geschwindigkeiten, im Verhältnis zur Zykluszeit der Safety-CPU, ist keine sinnvolle Berechnung mehr möglich. Um diesen Punkt zu umgehen kann durch die Eingangsvariable SampleTime die „Abtastzeit“ des Objekts eingestellt werden. Wird die Eingangsvariable SampleTime auf 0 gestellt, so wird das Objekt in jedem Zyklus ausgeführt. Wird auf der Eingangsvariable SampleTime ein Wert von 100 (ms) eingestellt, so wird das Objekt erst nach Ablauf von 100 ms erneut ausgeführt.



Bei der Einstellung der SampleTime muss die Abtastzeit der Position berücksichtigt werden. Diese Zeit beträgt z.B. bei Eingängen des XS-322-2INC-Moduls 1 ms. Bei einer Übertragung der Position über den Bus muss diese Zeit ebenfalls berücksichtigt werden, um eine korrekte Berechnung zu ermöglichen. Die eingestellte SampleTime sollte immer höher sein als die Zeit zwischen Positionsänderungen.

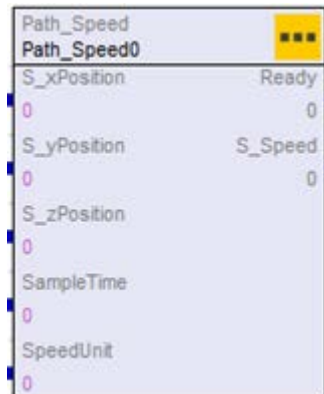
In diesem Baustein wird davon ausgegangen, dass bei den Eingangsvariablen S_xPosition, S_yPosition, S_zPosition kein Überlauf auftreten darf. Da ein Überlauf bei einer räumlich begrenzten Achse nicht sinnvoll ist.

Die Benennung des Bausteins mit PATH_SPEED ist spezifisch gewählt, um die Funktion für den Anwender deutlicher zu machen. Der Baustein kann aber allgemein für die Ableitung dreier Größen über die Zeit mit Berechnung der euklidischen Norm der einzelnen Terme verwendet werden. So ist es auch möglich den Baustein für die Ermittlung der Beschleunigung zu verwenden.

Die Filterung der Geschwindigkeit kann nachfolgend mit einem PT1-Block durchgeführt werden.

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke



Eingangsvariablen

S_xPosition	Position der X-Achse [einheitenlos] (SAFEDINT)
S_yPosition	Position der Y-Achse [einheitenlos] (SAFEDINT)
S_zPosition	Position der Z-Achse [einheitenlos] (SAFEDINT)
SampleTime	Über diese Eingangsvariable wird das Zeitintervall zur Ausführung des Objekts festgelegt. Wird der Wert 0 angegeben, wird das Objekt in jedem Zyklus ausgeführt. Wird ein Wert größer 0 angegeben, so muss diese Zeitdauer seit dem letzten Durchlauf vergangen sein, damit die Geschwindigkeit am Ausgang berechnet wird. Der Wert wird in ms angegeben. Es sind nur positive Werte erlaubt. (DINT)
SpeedUnit	Einheit der Ausgabe (0 = ms, 1 = s, 2 = min) (DINT)

Ausgangsvariablen

Ready	Zeigt an, ob der Funktionsblock aktiv ist (BOOL)
S_Speed	Berechnete Geschwindigkeit [Position-Unit pro SpeedUnit] (SAFEDINT)

Der Funktionsblock besitzt eine Anlaufsperr, welche durchlaufen werden muss, bis der Funktionsblock richtig funktioniert. Nach dem Initialisieren oder nach dem automatischen Reset eines Fehlers muss der Funktionsblock gegebenenfalls mehrere Zyklen durchlaufen, bis er ein richtiges Ergebnis am Ausgang ausgibt und Ready auf TRUE geschaltet wird. Dies ist abhängig von der eingestellten SampleTime. Es müssen mindestens 3 Zyklen der Safety-CPU durchlaufen werden, bis eine korrekte Berechnung durchgeführt werden kann.

Formel der Berechnung:

$$v = \sqrt{\left(\frac{x(t+h) - x(t)}{h}\right)^2 + \left(\frac{y(t+h) - y(t)}{h}\right)^2 + \left(\frac{z(t+h) - z(t)}{h}\right)^2}$$

v Berechnete Geschwindigkeit

x x-Position

y y-Position

z z-Position

t Zeitstempel der letzten Berechnung in [ms]

h Zeitintervall für die Berechnung (SampleTime)

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.3.21 Speed

Der Baustein SPEED dient zur Ableitung der Position, um daraus die Geschwindigkeit zu erhalten. Er kann somit zur Bestimmung der Geschwindigkeit einer Einzel-Achse verwendet werden. Durch die Eingangsvariablen kann das Verhalten des Bausteins konfiguriert werden.

Die Eingangsvariable S_Position ist einheitenlos. Es obliegt somit dem Anwender wie er den Positionswert interpretiert. Für die Ausgangsvariable S_Speed kann die Zeiteinheit durch die Eingangsvariable SpeedUnit gewählt werden.

Bei sehr kleinen Geschwindigkeiten, im Verhältnis zur Zykluszeit der Safety-CPU, ist keine sinnvolle Berechnung mehr möglich. Um diesen Punkt zu umgehen kann durch die Eingangsvariable SampleTime die „Abtastzeit“ des Objekts eingestellt werden. Wird die Eingangsvariable SampleTime auf 0 gestellt, so wird das Objekt in jedem Zyklus ausgeführt. Wird auf der Eingangsvariable SampleTime ein Wert von 100 (ms) eingestellt, so wird das Objekt erst nach Ablauf von 100 ms erneut ausgeführt.



Bei der Einstellung der SampleTime muss die Abtastzeit der Position berücksichtigt werden. Diese Zeit beträgt z.B. bei Eingängen des XS-322-2INC-Moduls 1 ms. Bei einer Übertragung der Position über den Bus muss diese Zeit ebenfalls berücksichtigt werden, um eine korrekte Berechnung zu ermöglichen. Die eingestellte SampleTime sollte immer höher sein als die Zeit zwischen Positionsänderungen.

Da der Positionswert am Eingang S_Position überlaufen kann, muss dies in der Berechnung der Geschwindigkeit beachtet werden. Hierfür muss der Wert, an dem die Position überläuft, bekannt sein. Dieser Wert wird durch die Eingangsvariable Overflow angegeben.

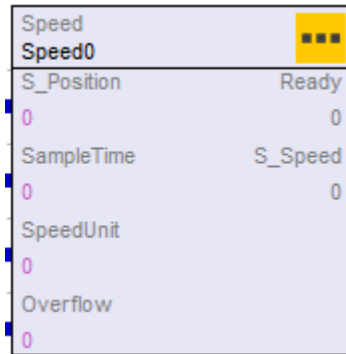
Bei sehr großen Geschwindigkeiten ist darauf zu achten, dass in der bei SampleTime eingestellten Zeit nicht mehr als die Hälfte des in Overflow angegebenen Weg zurückgelegt wird.

Die Benennung des Bausteins mit SPEED ist spezifisch gewählt, um die Funktion für den Anwender deutlicher zu machen. Der Baustein an sich kann aber allgemein für die Ableitung einer Größe über die Zeit verwendet werden. So ist es auch möglich den Baustein für die Ermittlung der Beschleunigung zu verwenden (mit Overflow 0).

Die Filterung der Geschwindigkeit kann nachfolgend mit einem PT1-Block durchgeführt werden.

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke



Eingangsvariablen

S_Position	Position der Achse. Für den Baustein ist dieser Wert einheitenlos. (SAFEDINT)
SampleTime	Über diese Eingangsvariable wird das Zeitintervall zur Ausführung des Objekts festgelegt. Wird der Wert 0 angegeben wird das Objekt in jedem Zyklus ausgeführt. Wird ein Wert größer 0 angegeben, so muss diese Zeitdauer seit dem letzten Durchlauf vergangen sein, damit die Geschwindigkeit am Ausgang berechnet wird. Der Wert wird in ms angegeben. Es sind nur positive Werte erlaubt. (DINT)
SpeedUnit	Einheit der Ausgabe für S_Speed (0 = ms, 1 = s, 2 = min) (DINT)
Overflow	An der Eingangsvariable Overflow muss der Wert eingetragen werden, an dem die Position (S_Position) einen Überlauf aufweist. Bei einem eingestellten Wert wird die Position von 0 bis zum eingestellten Wert überprüft. Wird der Wert 0 angegeben, so wird als Überlauf der Minimalwert und Maximalwert von DINT verwendet. (DINT)

Ausgangsvariablen

Ready	Zeigt an, ob der Funktionsblock aktiv ist (BOOL)
S_Speed	Berechnete Geschwindigkeit [Position-Unit pro SpeedUnit] (SAFEDINT)

Der Funktionsblock besitzt eine Anlaufsperr, welche durchlaufen werden muss, bis der Funktionsblock richtig funktioniert. Nach dem Initialisieren oder nach dem automatischen Reset eines Fehlers muss der FUB gegebenenfalls mehrere Zyklen durchlaufen, bis er ein richtiges Ergebnis am Ausgang ausgibt und Ready auf TRUE geschaltet wird. Dies ist abhängig von der eingestellten SampleTime. Es müssen mindestens 3 Zyklen der Safety-CPU durchlaufen werden, bis eine korrekte Berechnung durchgeführt werden kann. Am Ausgang S_Speed wird 0 ausgegeben, solange der Ausgang Ready auf false steht.

Formel der Berechnung:

$$v = \frac{x(t+h) - x(t)}{h}$$

- v Berechnete Geschwindigkeit (S_Speed)
- x(t+h) Positionswert (s_Pos) im aktuellen Aufruf
- x(t) Positionswert (s_Pos) im letzten Aufruf
- t Zeitstempel der letzten Berechnung in [ms]
- h Zeitintervall für die Berechnung (SampleTime)

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

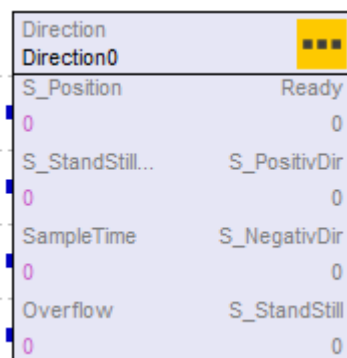
13.4.3.22 Direction

Der Funktionsblock berechnet aus der Positionsänderung die Bewegungsrichtung. Es kann eine Toleranz (S_StandStillWindow) eingestellt werden, welche überschritten werden muss, dass die Berechnung richtig ausgegeben wird. Wird die Toleranz nicht überschritten gibt der Funktionsblock die Bewegungsrichtung aus. Der Ausgang S_PositiveDir wechselt auf `SAFEBOOL_TRUE` wenn in positive Richtung gefahren wird. Der Ausgang S_NegativeDir wechselt auf `SAFEBOOL_TRUE` wenn in negative Richtung gefahren wird. Der Ausgang `_Standstill` wechselt auf `SAFEBOOL_TRUE` wenn in keine Richtung gefahren wird.



Bei der Einstellung der `SampleTime` muss die Abtastrate der Position berücksichtigt werden. Diese Zeit beträgt z.B bei Eingängen des XS-322-2INC-Moduls 1 ms. Bei einer Übertragung der Position über den Bus muss diese Zeit ebenfalls berücksichtigt werden, um eine korrekte Berechnung zu ermöglichen. Die eingestellte `SampleTime` sollte immer höher sein als die Zeit zwischen Positionsänderungen.

Bei sehr großen Geschwindigkeiten ist darauf zu achten, dass in der bei `SampleTime` eingestellten Zeit nicht mehr als die Hälfte des in `Overflow` angegebenen Weg zurückgelegt wird.



Eingangsvariablen

S_Position	Position der Achse. Für den Baustein ist dieser Wert einheitenlos. (SAFEDINT)
S_StandStillWindow	Toleranz der Positionsänderung. Es ist die maximal erlaubte Änderung der Position innerhalb der eingestellten Zeit (SampleTime), um einen Stillstand zu erkennen. Der eingestellte Wert wirkt in positive und negative Richtung. (SAFEDINT)
SampleTime	Zeitintervall der Berechnung in ms. Wird der Wert 0 eingestellt, wird immer über einen Zyklus der Safety-CPU berechnet. (DINT)
Overflow	An der Eingangsvariable Overflow muss der Wert eingetragen werden, an dem die Position (S_Position) einen Überlauf aufweist. Bei einem eingestellten Wert wird die Position von 0 bis zum eingestellten Wert überprüft. Wird der Wert 0 angegeben so wird als Überlauf der Minimalwert und Maximalwert von DINT verwendet. (DINT)

Ausgangsvariablen

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

Ready	Zeigt an, ob der Funktionsblock aktiv ist (BOOL)
S_PositiveDir	Berechnete Bewegungsrichtung (SAFEDINT) SAFEBOOL_TRUE = positive Bewegungsrichtung SAFEBOOL_FALSE = negative Bewegungsrichtung und Stillstand
S_NegativeDir	Berechnete Bewegungsrichtung (SAFEDINT) SAFEBOOL_TRUE = negative Bewegungsrichtung SAFEBOOL_FALSE = positive Bewegungsrichtung und Stillstand
S_StandStill	Berechnete Bewegungsrichtung (SAFEDINT) SAFEBOOL_TRUE = Stillstand SAFEBOOL_FALSE = positive oder negative Bewegungsrichtung

Der Funktionsblock besitzt eine Anlaufsperrung, welche durchlaufen werden muss, bis der Funktionsblock richtig funktioniert. Nach dem Initialisieren oder nach dem automatischen Reset eines Fehlers muss der Funktionsblock gegebenenfalls mehrere Zyklen durchlaufen, bis er ein richtiges Ergebnis am Ausgang ausgibt und Ready auf TRUE geschaltet wird. Dies ist abhängig von der eingestellten SampleTime. Es müssen mindestens 3 Zyklen der Safety-CPU durchlaufen werden, bis eine korrekte Berechnung durchgeführt werden kann.

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

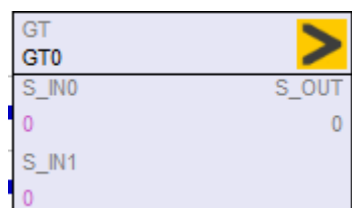
13.4.4 Vergleichende Funktionsblöcke

Tab. 4: Übersicht

→ "GT (größer als)", Seite 189 → "GT_NS (größer als, not safe)", Seite 190		Größer als
→ "GE (größer gleich)", Seite 190 → "GE_NS (größer gleich, not safe)", Seite 190		Größer gleich
→ "LT (kleiner als)", Seite 191 → "LT_NS (kleiner als, not safe)", Seite 191		Kleiner als
→ "LE (kleiner gleich)", Seite 191 → "LE_NS (kleiner gleich, not safe)", Seite 192		Kleiner gleich
→ "EQ (gleich)", Seite 192 → "EQ_NS (gleich, not safe)", Seite 192		Gleich
→ "MIN (Minimum)", Seite 193 → "MIN_NS (Minimum, not safe)", Seite 193		Minimum
→ "MAX (Maximum)", Seite 194 → "MAX_NS (Maximum, not safe)", Seite 194		Maximum
→ "LIMIT (innerhalb Grenzen)", Seite 195 → "LIMIT_NS (innerhalb Grenzen, not safe)", Seite 195		Innerhalb von Schranken
→ "SDINT_Compare", Seite 196		Vergleich von zwei SAFEDINT-Eingängen über einen Zeitraum
→ "DINT_Compare", Seite 197		Vergleich von zwei DINT-Eingängen über einen Zeitraum

13.4.4.1 GT (größer als)

Gibt an, ob S_IN0 größer als S_IN1 ist.



Eingangsvariablen

S_IN0 Eingang 1 (SAFEDINT)

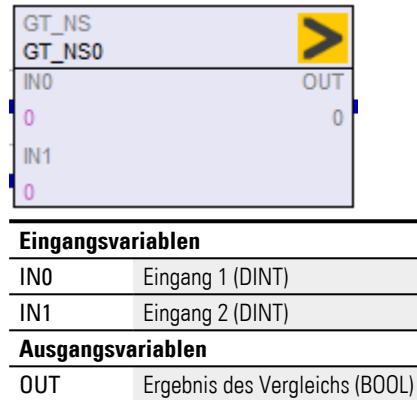
S_IN1 Eingang 2 (SAFEDINT)

Ausgangsvariablen

S_OUT Ergebnis des Vergleichs (SAFEBOOL)

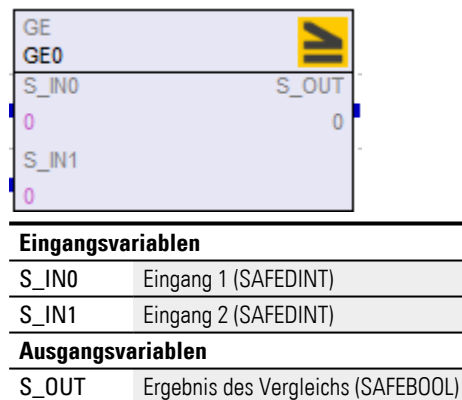
13.4.4.2 GT_NS (größer als, not safe)

Gibt an, ob IN0 größer als IN1 ist.



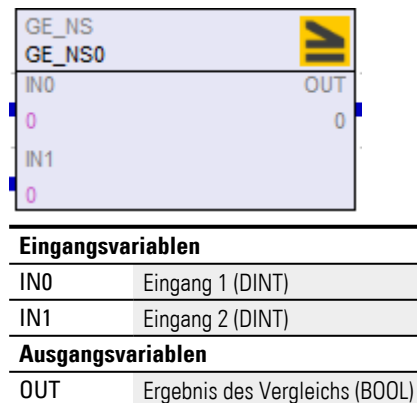
13.4.4.3 GE (größer gleich)

Gibt an, ob S_IN0 größer oder gleich S_IN1 ist.



13.4.4.4 GE_NS (größer gleich, not safe)

Gibt an, ob IN0 größer oder gleich IN1 ist.

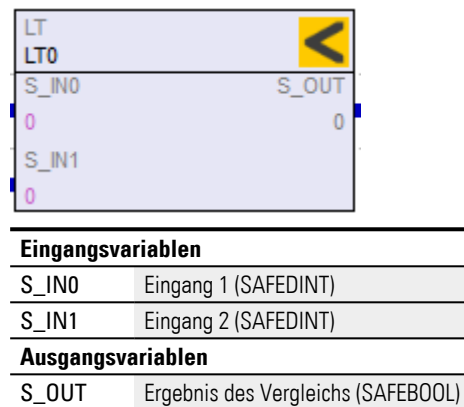


13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

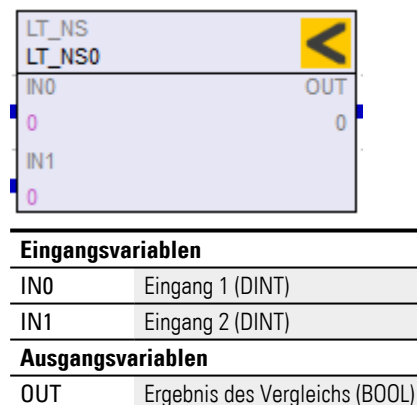
13.4.4.5 LT (kleiner als)

Gibt an, ob S_IN0 kleiner als S_IN1 ist.



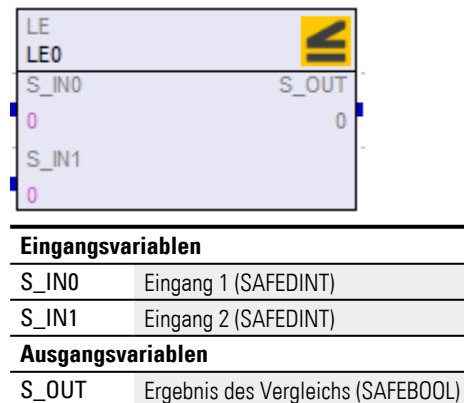
13.4.4.6 LT_NS (kleiner als, not safe)

Gibt an, ob IN0 kleiner als IN1 ist.



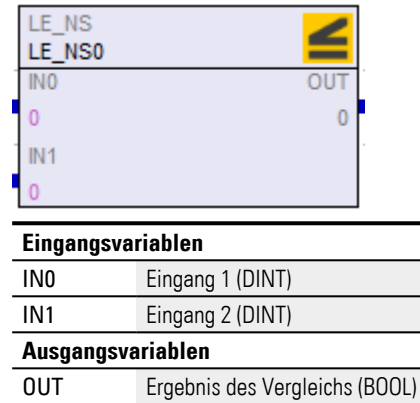
13.4.4.7 LE (kleiner gleich)

Gibt an, ob S_IN0 kleiner oder gleich S_IN1 ist.



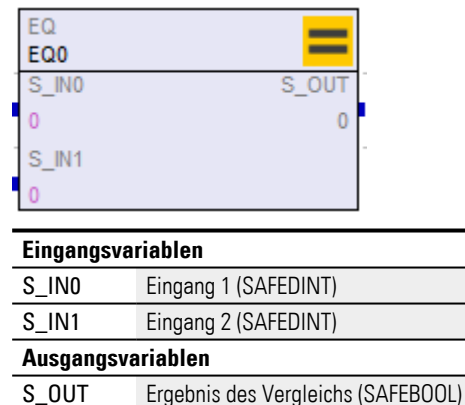
13.4.4.8 LE_NS (kleiner gleich, not safe)

Gibt an, ob IN0 kleiner oder gleich IN1 ist.



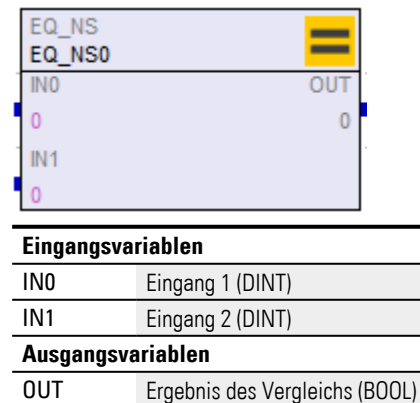
13.4.4.9 EQ (gleich)

Gibt an, ob S_IN0 gleich S_IN1 ist.



13.4.4.10 EQ_NS (gleich, not safe)

Gibt an, ob IN0 gleich IN1 ist.



13. Funktionsblöcke

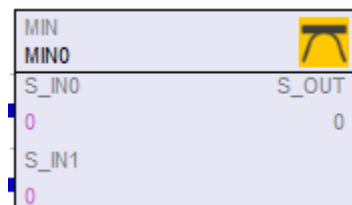
13.4 Standard-Funktionsblöcke

13.4.4.11 MIN (Minimum)

Gibt den kleineren Wert von S_IN0 und S_IN1 aus.



Wird an einen Eingang eine Konstante angeschlossen, bildet sie eine obere Schranke für den anderen Wert.



Eingangsvariablen

S_IN0	Eingang 1 (SAFEDINT)
-------	----------------------

S_IN1	Eingang 2 (SAFEDINT)
-------	----------------------

Ausgangsvariablen

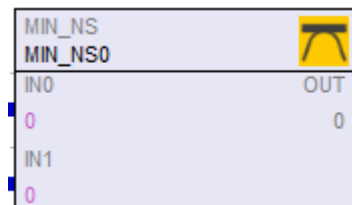
S_OUT	Minimum der Eingänge (SAFEDINT)
-------	---------------------------------

13.4.4.12 MIN_NS (Minimum, not safe)

Gibt den kleineren Wert von IN0 und IN1 aus.



Wird an einen Eingang eine Konstante angeschlossen, bildet sie eine obere Schranke für den anderen Wert.



Eingangsvariablen

IN0	Eingang 1 (DINT)
-----	------------------

IN1	Eingang 2 (DINT)
-----	------------------

Ausgangsvariablen

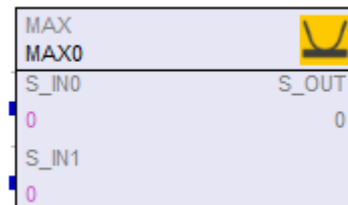
OUT	Minimum der Eingänge (DINT)
-----	-----------------------------

13.4.4.13 MAX (Maximum)

Gibt den größeren Wert von S_IN0 und S_IN1 aus.



Wird an einen Eingang eine Konstante angeschlossen, bildet sie eine untere Schranke für den anderen Wert.



Eingangsvariablen

S_IN0	Eingang 1 (SAFEDINT)
S_IN1	Eingang 2 (SAFEDINT)

Ausgangsvariablen

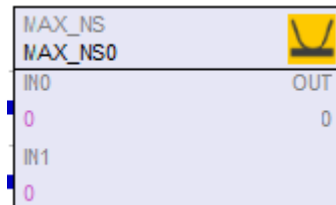
S_OUT	Maximum der Eingänge (SAFEDINT)
-------	---------------------------------

13.4.4.14 MAX_NS (Maximum, not safe)

Gibt den größeren Wert von IN0 und IN1 aus.



Wird an einen Eingang eine Konstante angeschlossen, bildet sie eine untere Schranke für den anderen Wert.



Eingangsvariablen

IN0	Eingang 1 (DINT)
IN1	Eingang 2 (DINT)

Ausgangsvariablen

OUT	Maximum der Eingänge (DINT)
-----	-----------------------------

13. Funktionsblöcke

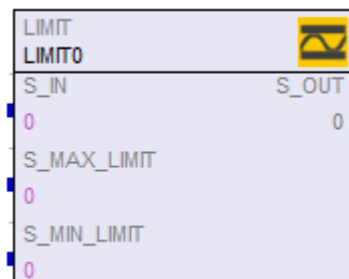
13.4 Standard-Funktionsblöcke

13.4.4.15 LIMIT (innerhalb Grenzen)

Liefert einen Eingangswert innerhalb von oberer und unterer Schranke. Ist S_IN größer als S_MAX_LIMIT dann wird S_MAX_LIMIT ausgegeben, ist S_IN kleiner als S_MIN_LIMIT, dann S_MIN_LIMIT, ansonsten S_IN.



Falls S_MAX_LIMIT kleiner als S_MIN_LIMIT ist, wird SAFEDINT_ERROR ausgegeben.



Eingangsvariablen

S_IN	Eingang (SAFEDINT)
S_MAX_LIMIT	Obere Schranke (SAFEDINT)
S_MIN_LIMIT	Untere Schranke (SAFEDINT)

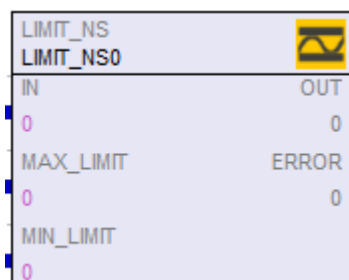
Ausgangsvariablen

S_OUT	Ergebnis (SAFEDINT), SAFEDINT_ERROR im Fehlerfall
-------	--

13.4.4.16 LIMIT_NS (innerhalb Grenzen, not safe)

Liefert einen Eingangswert innerhalb von oberer und unterer Schranke. Ist IN größer als MAX_LIMIT dann wird MAX_LIMIT ausgegeben, ist IN kleiner als MIN_LIMIT, dann MIN_LIMIT, ansonsten IN.

Ist MAX_LIMIT < MIN_LIMIT, dann wird ERROR = TRUE und OUT = IN gesetzt.



Eingangsvariablen

IN	Eingang (DINT)
MAX_LIMIT	Obere Schranke (DINT)
MIN_LIMIT	Untere Schranke (DINT)

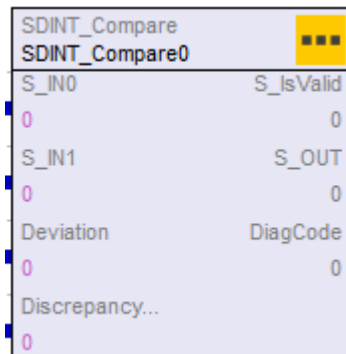
Ausgangsvariablen

OUT	Ergebnis (DINT)
ERROR	TRUE, wenn MAX_LIMIT < MIN_LIMIT

13.4.4.17 SDINT_Compare

Vergleicht zwei SAFEDINT-Eingänge S_IN0 und S_IN1 miteinander und gibt den Durchschnittswert aus.

Deviation gibt an, wie groß die Differenz sein darf. DiscrepancyTime ist dabei die Zeitspanne, innerhalb der die Differenz auch größer sein darf. Ist der Absolutbetrag der Differenz von S_IN0 und S_IN1 länger als „DiscrepancyTime“ größer als „Deviation“ wird S_IsValid auf SAFEBOOL_FALSE und S_OUT auf 0 gesetzt, solange bis die Differenz wieder kleiner gleich „Deviation“ ist.



Eingangsvariablen

S_IN0	Eingang 1 (SAFEDINT)
S_IN1	Eingang 2 (SAFEDINT)
Deviation	Erlaubte Differenz (DINT)
DiscrepancyTime	Zeit, innerhalb der die Differenz größer sein darf (DINT)

Ausgangsvariablen

S_IsValid	SAFEBOOL_TRUE, wenn S_OUT gültig ist (SAFEBOOL)
S_OUT	Durchschnittswert von S_IN0 und S_IN1 bzw. 0 im Fehlerfall (SAFEDINT)
DiagCode	Statusanzeige (HDINT)

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C000	SafeDintError	An S_IN0 oder S_IN1 liegt SAFEDINT_ERROR an
C001	DeviationError	Wert für Deviation ist ungültig (< 0)
C002	DiscrepancyError	Wert für DiscrepancyTime ist ungültig (< 0)
C003	DiffError	Die Differenz von S_IN0 und S_IN1 ist länger als DiscrepancyTime größer als Deviation
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle/Init	Der Funktionsblock ist nicht aktiv (Ausgangszustand) oder Activate = FALSE Ready = FALSE S_Y_Value = 0 Error = FALSE Error = FALSE
8000	NoError	S_OUT ist gültiger Wert

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.4.18 DINT_Compare

Vergleicht zwei DINT-Eingänge „IN0“ und „IN1“ miteinander und gibt den Durchschnittswert aus.

Deviation gibt an, wie groß die Differenz sein darf. DiscrepancyTime ist dabei die Zeitspanne, innerhalb der die Differenz auch größer sein darf. Ist der Absolutbetrag der Differenz von IN0 und IN1 länger als „DiscrepancyTime“ größer als „Deviation“ wird IsValid auf FALSE und OUT auf 0 gesetzt, solange bis die Differenz wieder kleiner gleich „Deviation“ ist.

DINT_Compare		***
DINT_Compare0		
IN0	IsValid	
0	0	
IN1	OUT	
0	0	
Deviation	DiagCode	
0	0	
Discrepancy...		
0		

Eingangsvariablen

IN0	Eingang 1 (DINT)
IN1	Eingang 2 (DINT)
Deviation	Erlaubte Differenz (DINT)
DiscrepancyTime	Zeit, innerhalb der die Differenz größer sein darf (DINT)

Ausgangsvariablen

IsValid	SAFEBOOL_TRUE, wenn S_OUT gültig ist (BOOL)
OUT	Durchschnittswert von IN0 und IN1 bzw. 0 im Fehlerfall (DINT)
DiagCode	Statusanzeige (HDINT)

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	DeviationError	Wert für Deviation ist ungültig (< 0)
C002	DiscrepancyError	Wert für DiscrepancyTime ist ungültig (< 0)
C003	DiffError	Die Differenz von S_IN0 und S_IN1 ist länger als DiscrepancyTime größer als Deviation
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle/Init	Der Funktionsblock ist nicht aktiv (Ausgangszustand) oder Activate = FALSE Ready = FALSE S_Y_Value = 0 Error = FALSE
8000	NoError	OUT ist gültiger Wert

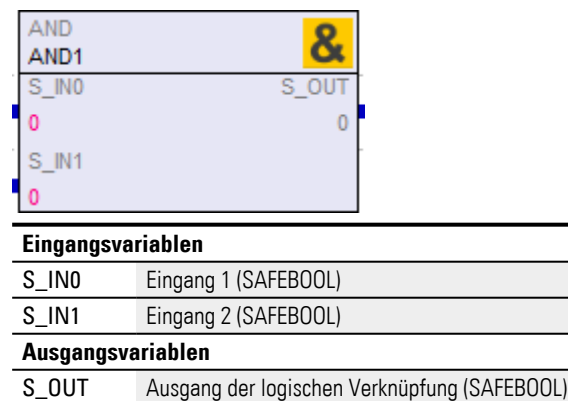
13.4.5 Logical-Funktionsblöcke

Tab. 5: Übersicht

→ "AND", Seite 198		Logische UND-Verknüpfung
→ "AND_5I", Seite 199		Logische UND-Verknüpfung mit 5 Eingängen
→ "AND_NS - AND NOT SAFE", Seite 199		Logische UND-Verknüpfung mit einem BOOL- und einem SAFEBOOL-Eingang
→ "NOT", Seite 200		Umkehrung des Eingangs
→ "OR", Seite 200		Logische ODER-Verknüpfung
→ "OR_5I", Seite 201		Logische ODER-Verknüpfung mit 5 Eingängen
→ "XOR", Seite 201		Exklusiv ODER-Verknüpfung
→ "AND_XI", Seite 202		Logische UND-Verknüpfung mit variabler Anzahl von Eingängen
→ "OR_XI", Seite 203		Logische ODER-Verknüpfung mit variabler Anzahl von Eingängen
→ "SAFEBOOL_TO_BOOL", Seite 170		Umwandlung eines SAFE-Einganges in einen UNSAFE-Ausgang
→ "SAFEDINT_TO_DINT", Seite 170		Umwandlung eines SAFEDINT-Einganges in einen UNSAFE-Ausgang
→ "SERROR_TO_SFALSE", Seite 171		Umwandlung von SAFEBOOL_ERROR - Zuständen in SAFEBOOL_FALSE.

13.4.5.1 AND

Logische UND-Verknüpfung von zwei SAFEBOOL-Eingängen.



13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.5.2 AND_5I

Logische UND-Verknüpfung mit 5 SAFEBOOL-Eingängen.

AND_5I
AND_5I0

0

0

0

0

0

0

S_OUT

0

Eingangsvariablen

S_IN0	Eingang 1 (SAFEBOOL)
S_IN1	Eingang 2 (SAFEBOOL)
S_IN2	Eingang 3 (SAFEBOOL)
S_IN3	Eingang 4 (SAFEBOOL)
S_IN4	Eingang 5 (SAFEBOOL)

Ausgangsvariablen

S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)
-------	--

13.4.5.3 AND_NS - AND NOT SAFE

Logische UND-Verknüpfung von einem BOOL- und einem SAFEBOOL-Eingang.

AND_NS
AND_NS0

0

0

S_OUT

0

Eingangsvariablen

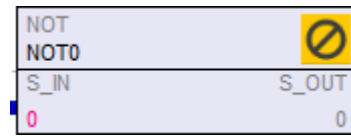
IN	Eingang 1 (BOOL)
S_IN	Eingang 2 (SAFEBOOL)

Ausgangsvariablen

S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)
-------	--

13.4.5.4 NOT

Umkehrung des Eingangs.



Eingangsvariablen

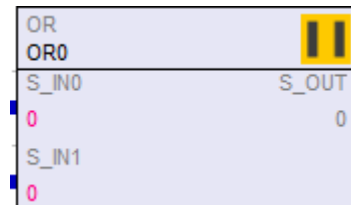
S_IN	Eingang (SAFEBOOL)
------	--------------------

Ausgangsvariablen

S_OUT	Ausgang (SAFEBOOL)
-------	--------------------

13.4.5.5 OR

Logische ODER-Verknüpfung von zwei SAFEBOOL-Eingängen.



Eingangsvariablen

S_IN0	Eingang 1 (SAFEBOOL)
-------	----------------------

S_IN1	Eingang 2 (SAFEBOOL)
-------	----------------------

Ausgangsvariablen

S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)
-------	--

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.5.6 OR_5I

Logische ODER-Verknüpfung von 5 SAFEBOOL-Eingängen.

OR_5I

OR_5I0

S_IN0

0

S_IN1

0

S_IN2

0

S_IN3

0

S_IN4

0

S_OUT

0

5

Eingangsvariablen	
S_IN0	Eingang 1 (SAFEBOOL)
S_IN1	Eingang 2 (SAFEBOOL)
S_IN2	Eingang 3 (SAFEBOOL)
S_IN3	Eingang 4 (SAFEBOOL)
S_IN4	Eingang 5 (SAFEBOOL)

Ausgangsvariablen	
S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)

13.4.5.7 XOR

Logische Exklusiv ODER-Verknüpfung von zwei SAFEBOOL-Eingängen.

XOR

XOR0

S_IN0

0

S_IN1

0

S_OUT

0

2

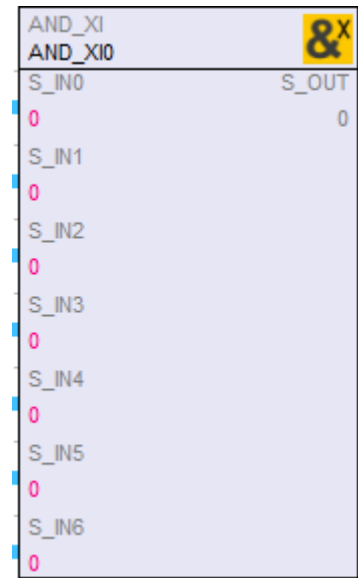
Eingangsvariablen	
S_IN0	Eingang 1 (SAFEBOOL)
S_IN1	Eingang 2 (SAFEBOOL)

Ausgangsvariablen	
S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)

13.4.5.8 AND_XI

Logische Und-Verknüpfung mit einer variablen Anzahl von SAFEBOOL-Eingängen (maximal 20).

Die variablen Eingänge sind an den blauen Symbolen zu erkennen.

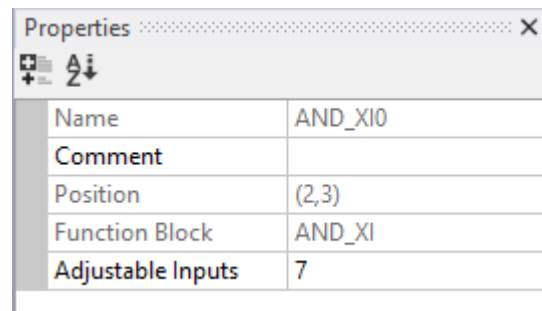


Eingangsvariablen	
S_IN0	Eingang 1 (SAFEBOOL)
S_IN1	Eingang 2 (SAFEBOOL)
...	...
S_IN18	Eingang 19 (SAFEBOOL)
S_IN19	Eingang 20 (SAFEBOOL)

Ausgangsvariablen	
S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)

Einstellen der Eingänge

Beim Anklicken des Funktionsblocks erscheint ein Dialog **Properties** mit dem Eintrag **Adjustable Inputs**. Hier wird angegeben, wie viele Eingänge verwendet werden sollen.



Properties	
Name	AND_XI0
Comment	
Position	(2,3)
Function Block	AND_XI
Adjustable Inputs	7

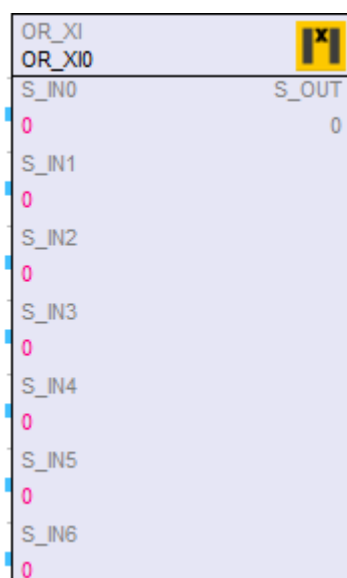
13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.5.9 OR_XI

Logische Oder-Verknüpfung mit einer variablen Anzahl von SAFEBOOL-Eingängen (maximal 20).

Die variablen Eingänge sind an den blauen Symbolen zu erkennen.



Eingangsvariablen

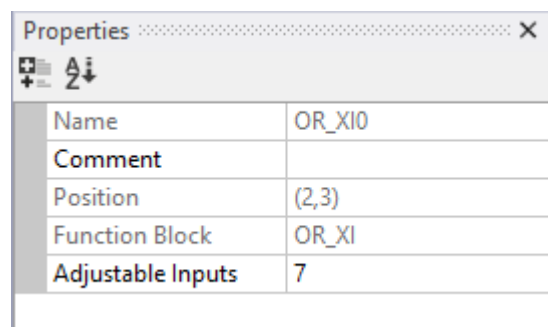
S_IN0	Eingang 1 (SAFEBOOL)
S_IN1	Eingang 2 (SAFEBOOL)
...	...
S_IN18	Eingang 19 (SAFEBOOL)
S_IN19	Eingang 20 (SAFEBOOL)

Ausgangsvariablen

S_OUT	Ausgang der logischen Verknüpfung (SAFEBOOL)
-------	--

Einstellen der Eingänge

Beim Anklicken des Funktionsblocks erscheint ein Dialog **Properties** mit dem Eintrag **Adjustable Inputs**. Hier wird angegeben, wie viele Eingänge verwendet werden sollen.



13.4.6 Timer-Funktionsblöcke

Tab. 6: Übersicht

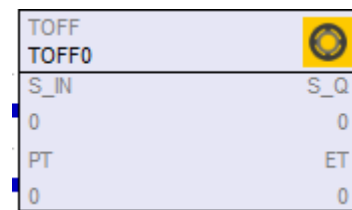
→ "TOFF", Seite 204		Ausschaltverzögerung
→ "TON", Seite 205		Einschaltverzögerung
→ "TP", Seite 205		Pulszeit

Besonderheiten bei den Timer FBs Ton, Toff, TP, TP1:

Wenn Flankenmerker (Hilfsvariable zur Erkennung einer steigenden bzw. fallenden Flanke) auf **SAFEBOOL_ERROR** steht, wird der Ausgang auf **SAFEBOOL_FALSE** gesetzt und der Timer gestoppt. Flankenmerker bleibt auf **SAFEBOOL_ERROR** bis Eingang wieder **SAFEBOOL_FALSE** ist.

13.4.6.1 TOFF

Ausschaltverzögerung eines **SAFEBOOL**-Signals.



Eingangsvariablen	
S_IN	Fallende Flanke startet den Timer (SAFEBOOL)
PT	Verzögerungsdauer in ms (DINT)
Ausgangsvariablen	
S_Q	FALSE wenn die Verzögerungszeit abgelaufen ist (SAFEBOOL)
ET	Bereits verstrichene Dauer der Verzögerung in ms (DINT)

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.6.2 TON

Einschaltverzögerung eines SAFEBOOL-Signals.

TON	
TON0	
S_IN	S_Q
0	0
PT	ET
0	0

Eingangsvariablen	
S_IN	Steigende Flanke startet den Timer (SAFEBOOL)
PT	Verzögerungsdauer in ms (DINT)
Ausgangsvariablen	
S_Q	TRUE wenn die Verzögerungszeit abgelaufen ist (SAFEBOOL)
ET	Bereits verstrichene Dauer der Verzögerung in ms (DINT)

13.4.6.3 TP

Impulszeit eines SAFEBOOL-Signals.

TP	
TP0	
S_IN	S_Q
0	0
PT	ET
0	0

Eingangsvariablen	
S_IN	Steigende Flanke startet den Timer (SAFEBOOL)
PT	Zeitdauer des Impuls in ms (DINT)
Ausgangsvariablen	
S_Q	TRUE während des Zeitablaufes (SAFEBOOL)
ET	Bereits verstrichene Dauer der Verzögerung in ms (DINT)

13.4.7 Trigger-Funktionsblöcke

Tab. 7: Übersicht

→ "F_TRIG", Seite 206		Triggererkennung einer fallenden Flanke
→ "R_TRIG", Seite 206		Triggererkennung einer steigenden Flanke

13.4.7.1 F_TRIG

Dient zur Erkennung einer fallenden Flanke am Eingang.

F_TRIG F_TRIG0 S_IN 0 S_Q 0	
Eingangsvariablen S_IN Bei fallender Flanke wird Q für einen Zyklus auf TRUE gesetzt (SAFEBOOL)	
Ausgangsvariablen S_Q (SAFEBOOL)	

13.4.7.2 R_TRIG

Dient zur Erkennung einer steigenden Flanke am Eingang.

R_TRIG R_TRIG0 S_IN 0 S_Q 0	
Eingangsvariablen S_IN Bei steigender Flanke wird Q für einen Zyklus auf TRUE gesetzt (SAFEBOOL)	
Ausgangsvariablen S_Q (SAFEBOOL)	

13.4.8 Counter-Funktionsblöcke

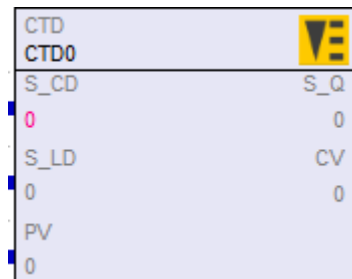
Tab. 8: Übersicht

→ "CTD - Abwärtszähler", Seite 207		Counter rückwärtszählend
→ "CTU - Aufwärtszähler", Seite 207		Counter vorwärtszählend
→ "CTUD - Auf-/ Abwärtszähler", Seite 208		Counter vorwärts- und rückwärtszählend

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.8.1 CTD - Abwärtszähler



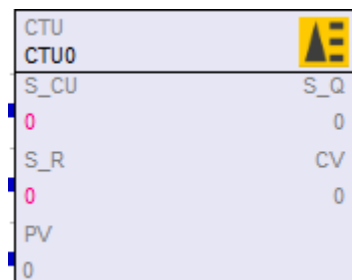
Eingangsvariablen

S_CD	Bei steigender Flanke wird um 1 rückwärts gezählt (SAFEBOOL)
S_LD	Initialisiert CV zu PV (SAFEBOOL)
PV	Anfangswert von wo aus gezählt wird (DINT)

Ausgangsvariablen

S_Q	TRUE, wenn CV das Minimum von 0 erreicht (SAFEBOOL)
CV	Aktueller Wert des Counters (DINT)

13.4.8.2 CTU - Aufwärtszähler



Eingangsvariablen

S_CU	Bei steigender Flanke wird um 1 vorwärts gezählt (SAFEBOOL)
S_R	Initialisiert CV zu 0 (SAFEBOOL)
PV	Maximalwert (DINT)

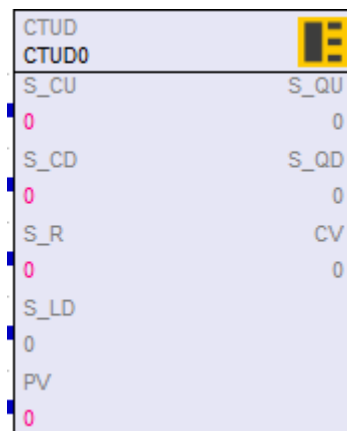
Ausgangsvariablen

S_Q	TRUE, wenn CV das Maximum (PV) erreicht (SAFEBOOL)
CV	Aktueller Wert des Counters (DINT)

13.4.8.3 CTUD - Auf-/ Abwärtszähler

Beim CTUD wird zuerst der Reset-Eingang geprüft. Dieser hat Vorrang gegenüber dem LD-Eingang. Ist der Reset-Eingang oder der LD gesetzt, so wird keine weitere Prüfung der Zählrichtungs-Eingänge vorgenommen. Ist dies nicht der Fall wird vorher die positive Zählrichtung und danach die negative Zählrichtung abgearbeitet. D.h. ist der Zählwert (CV) 0 oder < PV ändert sich CV nicht, da zuerst CV inkrementiert und dann wieder dekrementiert wird. Ist CV = PV wird CV verringert, da CV nicht mehr inkrementiert werden kann und anschließend CV dekrementiert wird.

Ist PV = 0 werden beide Ausgänge (QU und QD) auf SAFEBOOL_TRUE gesetzt.



Eingangsvariablen

S_CU	Bei steigender Flanke wird um 1 vorwärts gezählt (SAFEBOOL)
S_CD	Bei steigender Flanke wird um 1 rückwärts gezählt (SAFEBOOL)
S_R	Initialisiert CV zu 0 (SAFEBOOL)
S_LD	Initialisiert CV zu PV (SAFEBOOL)
PV	Anfangswert von wo aus gezählt wird (DINT)

Ausgangsvariablen

S_QU	TRUE wenn CV das Maximum (PV) erreicht (SAFEBOOL)
S_QD	TRUE wenn CV das Minimum von 0 erreicht (SAFEBOOL)
CV	Aktueller Wert des Counters (DINT)

13.4.9 FlipFlop-Funktionsblöcke

Tab. 9: Übersicht

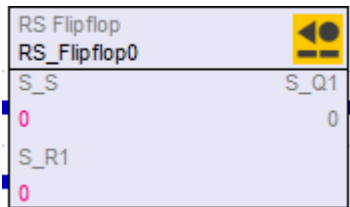
→ "RS_Flipflop", Seite 209		Override zurückgesetzt
→ "SR_Flipflop", Seite 209		Override gesetzt

13. Funktionsblöcke

13.4 Standard-Funktionsblöcke

13.4.9.1 RS_Flipflop

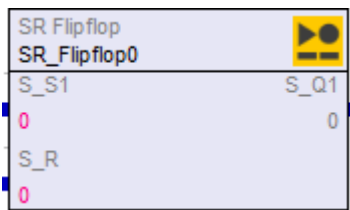
FlipFlop mit vorrangigem Rücksetzen.



Eingangsvariablen	
S_S	Wenn TRUE, dann wird Q1 auf TRUE gesetzt (SAFEBOOL)
S_R1	Wenn TRUE, dann wird Q1 auf FALSE gesetzt, auch wenn S_S auf TRUE ist (SAFEBOOL)
Ausgangsvariablen	
S_Q1	Ausgang der logischen Verknüpfung (SAFEBOOL)

13.4.9.2 SR_Flipflop

FlipFlop mit vorrangigem Setzen.



Eingangsvariablen	
S1	Wenn TRUE, dann wird Q1 permanent auf TRUE gesetzt (SAFEBOOL)
R	Wenn TRUE, dann wird Q1 auf FALSE gesetzt, wenn S1 auf FALSE ist (SAFEBOOL)
Ausgangsvariablen	
Q1	Ausgang der logischen Verknüpfung (SAFEBOOL)

13.5 Komplexe Funktionsblöcke



Drive-spezifische Funktionsblöcke sind nicht aufgeführt.

Tab. 10: Übersicht

→ "SF_Optional_PwdII-Funktion", Seite 297	Dient zur Kennwortabfrage und Kennwortüberprüfung bei optionalen Modulen. Dieser Funktionsblock ist nur zusammen mit dem Funktionsblock SF_Optional_Switch zu verwenden.
→ "SF_Optional_Switch-Funktion", Seite 303	Dieser Funktionsblock dient bei optionalen Modulen zum Wegschalten von Eingängen bzw. zum Umschalten auf eine andere Datenquelle (z.B. Konstante). Dieser Funktionsblock ist nur zusammen mit dem Funktionsblock SF_Optional_Pwd/SF_Optional_PwdII zu verwenden.
→ "SF_Optional_Switch_SDint", Seite 304	Dieser Funktionsblock dient bei optionalen Modulen zum Wegschalten von Eingängen bzw. zum Umschalten auf eine andere Datenquelle (z.B. Konstante).
→ "SF_Optional_Switch_Dint", Seite 305	Dieser Funktionsblock dient bei optionalen Modulen zum Wegschalten von Eingängen bzw. zum Umschalten auf eine andere Datenquelle (z.B. Konstante).
→ "SF_OutControl-Funktion", Seite 306	Dieser Funktionsblock dient zur Kontrolle eines Safety-Ausgangs mit einem Signal von der funktionalen Applikation und einem Safety-Signal mit optionaler Starthemmung.
→ "SF_SafetyRequest-Funktion", Seite 311	Dieser Funktionsblock dient als Schnittstelle zu einem generischen Aktuator, z. B. Safety Drive oder Sicherheitsventil, um den Aktuator in einen sicheren Zustand zu versetzen.
→ "SF_TestableSafetySensor-Funktion", Seite 316	Dieser Funktionsblock erkennt z. B. den Verlust einer Sensor-Erkennung, eine längere Reaktionszeit als angegeben und statische ON-Signale in Einzel-Kanal-Sensor-Systemen. Der Funktionsblock kann für extern testbare Sicherheits-Sensoren (ESPE) verwendet werden.
→ "SF_TwoHandControlTypII-Funktion", Seite 325	Dieser Funktionsblock bietet eine Zwei-Hand-Control-Funktionalität (siehe EN 574, Abschnitt 4 Typ II).
→ "SF_TwoHandControlTypIII-Funktion", Seite 330	Dieser Funktionsblock bietet eine Zwei-Hand-Control-Funktionalität (siehe EN 574, Abschnitt 4 Typ III). Eine fix definierte Zeitdifferenz von 500 ms ist vorgegeben.
→ "SF_HGW_Login", Seite 335	Der Funktionsblock dient zur Anbindung eines HGW Moduls. Er verfügt über zwei getrennt Anmelde-mechanismen. Mit dem ersten Anmelde-mechanismus kann eine Maschinenummer für das System gesetzt werden. Der zweite Anmelde-mechanismus dient zum sicheren Verbindungsaufbau zwischen HGW und Safety-CPU.
Numerische Funktionsblöcke	
→ "SF_LimitComparator", Seite 341	Überprüft das Eingangssignal auf oberes und unteres Limit.
→ "SF_LimitComparator_Dint", Seite 345	Überprüft das Eingangssignal auf oberes und unteres Limit.
→ "SF_Numeric_Selector", Seite 346	Wie SF_ModeSelector, jedoch mit SafeDint-Werten. Mit den S_ModeX SafeBool-Eingängen kann ein SafeDint-Eingang ausgewählt werden, der auf dem Ausgang S_Out geschaltet wird.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

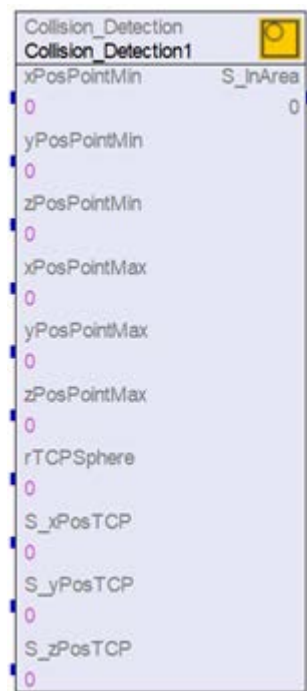
→ "SF_RampMonitor", Seite 350	RampMonitor überwacht die Geschwindigkeitsreduktion im Falle eines NOT-Halts. Eine steigende Flanke bei S_MonitoringOn bewirkt das Übernehmen der aktuellen Geschwindigkeit als Ausgangsgeschwindigkeit für die Überwachungsfunktion.
→ "SF_DirectionMonitor", Seite 354	Der Eingang S_Direction bestimmt die Richtung in welche der Eingangswert geprüft werden soll (positive Fahrt S_Direction = FALSE, negative Fahrt S_Direction = TRUE).
→ "SF_SkewMonitor", Seite 358	Eine steigende Flanke beim Eingang S_Calibrate startet die Kalibrierung und setzt dabei S_CalibrationOk auf FALSE.

13.5.1 Collision_Detection-Funktion

Der Funktionsblock Collision_Detection wurde entwickelt, um eine Kollision zwischen dem Tool Center Point (TCP) eines Roboters und anderen Objekten in seiner Arbeitsumgebung zu erkennen. Der Funktionsblock akzeptiert zwei Punkte, die zusammen einen Arbeitsraum aufspannen. Dieser Arbeitsraum ist der Bereich, in dem sich der TCP bewegen darf. Um den TCP kann eine Hüllkugel angegeben werden. Die Hüllkugel ist dabei eine mathematische Darstellung, die die äußere Begrenzung des TCP am Roboter in seiner aktuellen Konfiguration darstellt.

Am Ausgang liefert der Funktionsblock die Information, ob sich die Koordinaten des TCP mit Hüllkugel innerhalb des Arbeitsraums befinden. Ist dies der Fall so wird der Wert s_true geliefert. Wird der Arbeitsraum verlassen so wird am Ausgang s_false ausgegeben.

Die Information am Ausgang kann verwendet werden, um Maßnahmen zu ergreifen den Roboter zu stoppen bzw. eine Kollision zu verhindern.



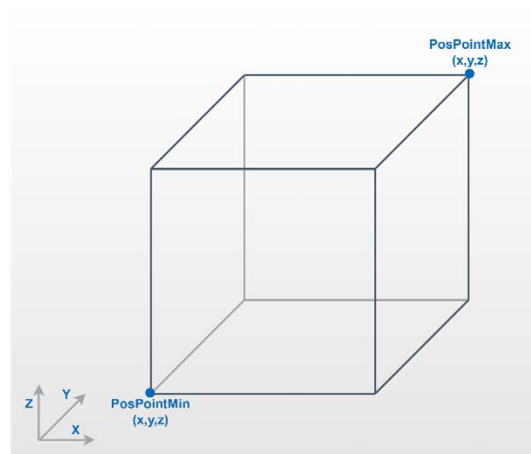
13.5.1.1 Funktionalitäten

Folgende Funktionalitäten sind im Funktionsblock enthalten:

- Der Funktionsbaustein überwacht die Position eines Roboters (TCP) im Arbeitsraum, legt eine Hüllkugel über seine aktuelle Position und erkennt Kollisionen, indem er die Hüllkugel mit den definierten Arbeitsraumgrenzen vergleicht.
- Die Information, ob die Arbeitsraumgrenzen verletzt wurden, wird am Ausgang als Bool-Wert zur Verfügung gestellt

13.5.1.2 Arbeitsraum

Der Arbeitsraum wird durch die XYZ-Koordinaten der Punkte PosPointMin und PosPointMax definiert. Diese beiden Punkte spannen dabei ein Quader auf. Für die Funktion des Funktionsblocks ist es notwendig das PosPointMax in allen drei Koordinaten (x,y,z) größer als PosPointMin ist. Die Koordinaten beziehen sich dabei auf das Basiskoordinatensystem der Kinematik. Um den Arbeitsraum über die Punkte PosPointMin und PosPointMax zu definieren können DINT-Konstanten verwendet werden.



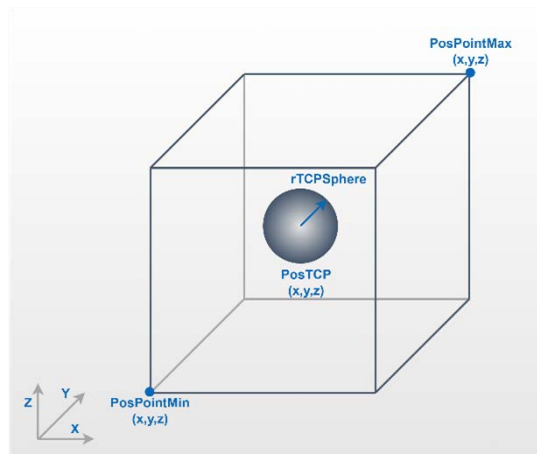
Die Abbildung zeigt das Basiskoordinatensystem und den von den Punkten PosPointMin und PosPointMax aufgespannten Arbeitsraum.

13.5.1.3 Tool Center Point und Hüllkugel

Die Koordinaten für den Tool Center Point (TCP) können abhängig von der Roboterkinematik direkt von einem sicheren Gebersystem übernommen oder durch den Funktionsblocks DH-Transformation berechnet werden. Um den TCP wird eine Hüllkugel gelegt. Der Radius der Hüllkugel wird durch den Wert von rTCPSphere definiert. Dieser Wert kann über eine DINT-Konstante angelegt werden.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

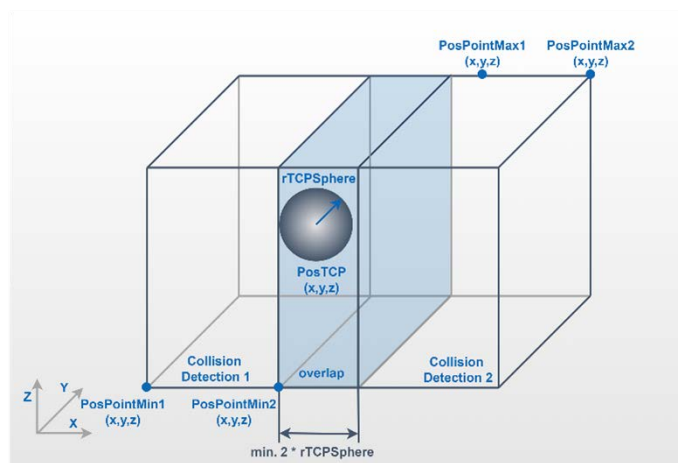


Die Abbildung zeigt die Hüllkugel im Arbeitsraum. Die Hüllkugel wird dabei durch PosTCP und rTCPSphere definiert.

13.5.1.4 Erstellung eines komplexen Arbeitsraums

Durch die logische Verknüpfung einzelner Arbeitsräume können komplexe Arbeitsräume erstellt werden. Hierfür müssen die Ausgänge mehrerer Collision_Detection Funktionsblöcke mit einem OR verknüpft werden. Der Ausgang des OR-Bausteins bildet damit die Information, ob sich der TCP im komplexen Arbeitsraum befindet.

Für einen komplexen Arbeitsraum müssen sich die einzelnen räumlichen Bereiche der Arbeitsräume so überlappen, dass die Hüllkugel in allen Achsen in den überschneidenden Bereich passt. Die Überlappung muss somit größer als 2-mal rTCPSphere sein.



In der Abbildung sind zwei überlappende Arbeitsbereiche zu sehen. Die Überlappung ist dabei größer als 2-mal rTCPSphere und der TCP kann sich somit von Arbeitsbereich 1 in den Arbeitsbereich 2 bewegen.

13.5.1.5 Schnittstellen

Eingangsvariablen	Beschreibung
xPosPointMin	Über diesen Eingang wird die minimale X-Position des Arbeitsraumes definiert. (DINT)
yPosPointMin	Über diesen Eingang wird die minimale Y-Position des Arbeitsraumes definiert. (DINT)
zPosPointMin	Über diesen Eingang wird die minimale Z-Position des Arbeitsraumes definiert. (DINT)
zPosPointMin	Über diesen Eingang wird die maximale X-Position des Arbeitsraumes definiert. (DINT)
yPosPointMax	Über diesen Eingang wird die maximale Y-Position des Arbeitsraumes definiert. (DINT)
zPosPointMax	Über diesen Eingang wird die maximale Z-Position des Arbeitsraumes definiert. (DINT)
rTCPSphere	Über diesen Eingang wird der Kugelradius der Hüllkugel definiert. (DINT)
S_xPosTCP	Über diesen Eingang wird die X-Position des zu überwachenden TCP definiert. (SAFEDINT)
S_yPosTCP	Über diesen Eingang wird die Y-Position des zu überwachenden TCP definiert. (SAFEDINT)
S_zPosTCP	Über diesen Eingang wird die Z-Position des zu überwachenden TCP definiert. (SAFEDINT)
Ausgangsvariablen	Beschreibung
S_InArea	Die Ausgangsvariable ist s_true, wenn sich die Kugel vollständig im aufgespannten Arbeitsraum befindet. Die Ausgangsvariable ist s_false, wenn die Kugel die Grenzen des Bereiches schneidet oder verlässt. (SAFEBOOL)

13.5.1.6 Beispiel zur Beschaltung des Funktionsblock

Die nachfolgende Abbildung zeigt, wie der Arbeitsraum mit Hilfe von DINT-Konstanten definiert werden kann. Die Position kann zum Beispiel durch den Funktionsblock DH-Transformation berechnet werden.

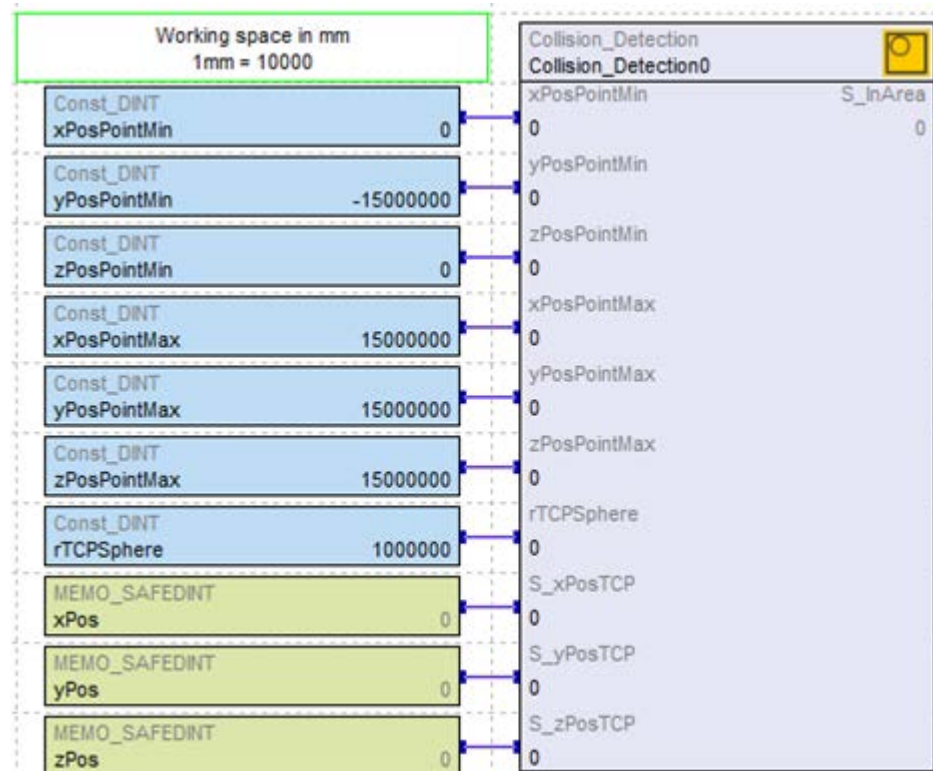


Abb. 3: Beispiel 6-Achs-Roboter

13. Funktionsblöcke

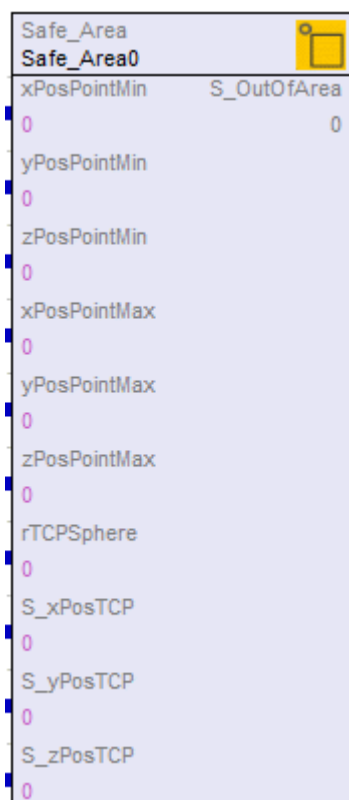
13.5 Komplexe Funktionsblöcke

13.5.2 Safe_Area-Funktion

Der Funktionsblock Safe_Area wurde entwickelt, um den Eintritt des Tool Center Point (TCP) eines Roboters in den sicheren Arbeitsraum zu erkennen. Der Funktionsblock akzeptiert zwei Punkte, die zusammen einen sicheren Arbeitsraum aufspannen. Dieser Arbeitsraum ist der Bereich, den der TCP nicht berühren oder schneiden darf. Um den TCP kann eine Hüllkugel angegeben werden. Die Hüllkugel ist dabei eine mathematische Darstellung, die die äußere Begrenzung des TCP am Roboter in seiner aktuellen Konfiguration darstellt.

Am Ausgang liefert der Funktionsblock die Information, ob sich die Koordinaten des TCP mit Hüllkugel außerhalb des sicheren Arbeitsraums befinden. Ist dies der Fall so wird der Wert s_true geliefert. Wird der Sicherheitsbereich berührt oder betreten so wird am Ausgang s_false ausgegeben.

Die Information am Ausgang kann verwendet werden, um Maßnahmen zu ergreifen den Roboter zu stoppen bzw. eine Kollision zu verhindern.



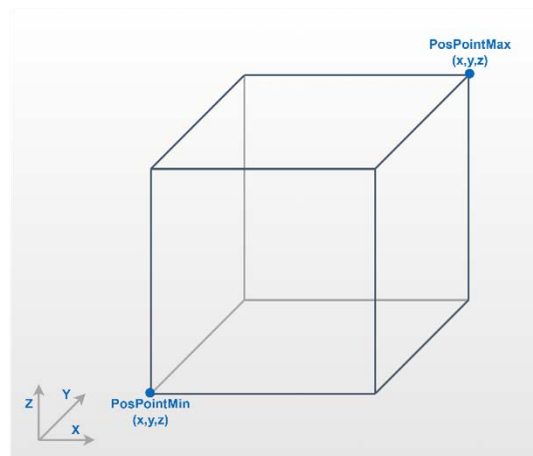
13.5.2.1 Funktionalitäten

Folgende Funktionalitäten sind im Funktionsblock enthalten:

- Der Funktionsbaustein überwacht die Position eines Roboters (TCP) außerhalb des sicheren Arbeitsraums, legt eine Hüllkugel über seine aktuelle Position und erkennt Kollisionen, indem er die Hüllkugel mit den definierten sicheren Arbeitsraumgrenzen vergleicht.
- Die Information, ob die sicheren Arbeitsraumgrenzen verletzt wurden, wird am Ausgang als Bool-Wert zur Verfügung gestellt.

13.5.2.2 Arbeitsraum

Der Sicherheitsbereich wird durch die XYZ-Koordinaten der Punkte PosPointMin und PosPointMax definiert. Diese beiden Punkte spannen dabei einen Quader auf. Für die Funktion des Funktionsblocks ist es notwendig das PosPointMax in allen drei Koordinaten (x,y,z) größer als PosPointMin ist. Die Koordinaten beziehen sich dabei auf das Basiskoordinatensystem der Kinematik. Um den sicheren Arbeitsraum über die Punkte PosPointMin und PosPointMax zu definieren können DINT-Konstanten verwendet werden.



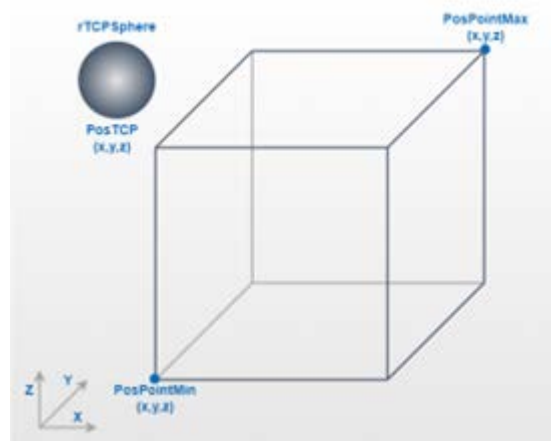
Die Abbildung zeigt das Basiskoordinatensystem und dem von den Punkten PosPointMin und PosPointMax aufgespannten sicheren Arbeitsraum.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.2.3 Tool Center Point und Hüllkugel

Die Koordinaten für den Tool Center Point (TCP) können abhängig von der Roboterkinematik direkt von einem sicheren Gebersystem übernommen oder durch den Funktionsblocks DH-Transformation berechnet werden. Um den TCP wird eine Hüllkugel gelegt. Der Radius der Hüllkugel wird durch den Wert von $r_{TCP\text{Sphere}}$ definiert. Dieser Wert kann über eine DINT-Konstante angelegt werden.



Die Abbildung zeigt die Hüllkugel außerhalb des sicheren Arbeitsraum. Die Hüllkugel wird dabei durch PosTCP und $r_{TCP\text{Sphere}}$ definiert.

13.5.2.4 Schnittstellen

Eingangsvariablen	Beschreibung
xPosPointMin	Über diesen Eingang wird die minimale X-Position des Sicherheitsbereichs definiert. (DINT)
yPosPointMin	Über diesen Eingang wird die minimale Y-Position des Sicherheitsbereichs definiert. (DINT)
zPosPointMin	Über diesen Eingang wird die minimale Z-Position des Sicherheitsbereichs definiert. (DINT)
xPosPointMax	Über diesen Eingang wird die maximale X-Position des Sicherheitsbereichs definiert. (DINT)
yPosPointMax	Über diesen Eingang wird die maximale Y-Position des Sicherheitsbereichs definiert. (DINT)
zPosPointMax	Über diesen Eingang wird die maximale Z-Position des Sicherheitsbereichs definiert. (DINT)
rTCP Sphere	Über diesen Eingang wird der Kugelradius der Hüllkugel definiert. (DINT)
S_xPosTCP	Über diesen Eingang wird die X-Position des zu überwachenden TCP definiert. (SAFEDINT)
S_yPosTCP	Über diesen Eingang wird die Y-Position des zu überwachenden TCP definiert. (SAFEDINT)
S_zPosTCP	Über diesen Eingang wird die Z-Position des zu überwachenden TCP definiert. (SAFEDINT)
Ausgangsvariablen	Beschreibung
S_InArea	Die Ausgangsvariable ist s_true, wenn sich die Kugel vollständig außerhalb des aufgespannten Sicherheitsbereichs befindet. Die Ausgangsvariable ist s_false, wenn die Kugel die Grenzen des Bereiches berührt oder eintritt. (SAFEBOOL)

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.2.5 Beispiel zur Beschaltung des Funktionsblock

Die nachfolgende Abbildung zeigt, wie der Sicherheitsbereich mit Hilfe von DINT-Konstanten definiert werden kann. Die Position kann zum Beispiel durch den Funktionsblock DH-Transformation berechnet werden.

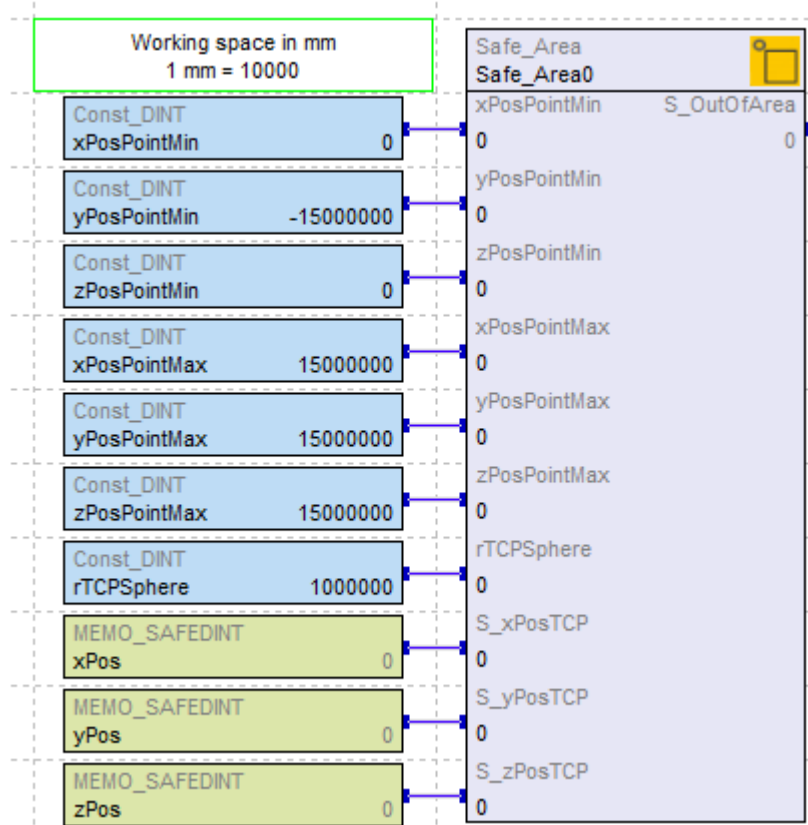
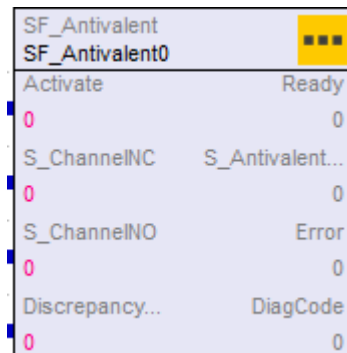


Abb. 4: Beispiel: 6-Achsen Roboter

13.5.3 SF_Antivalent-Funktion

Dieser Funktionsblock konvertiert zwei antivalente SAFEBOOL-Eingänge (NO / NC Paar) zu einem SAFEBOOL-Ausgang mit Diskrepanzzeit Überwachung. Dieser Funktionsblock sollte nicht als stand-alone verwendet werden, da keine Wiederanlaufsperrung vorhanden ist. Es ist erforderlich den Ausgang mit anderen sicherheitsrelevanten Funktionen zu verknüpfen.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL).
S_ChannelINC	NC steht für Öffner. Input für die NC-Verbindung. (SAFEBOOL) FALSE: Öffner geöffnet TRUE: Öffner geschlossen
S_ChannelNO	NO steht für Schließer. Input für die NO-Verbindung. (SAFEBOOL) FALSE: Schließer geöffnet TRUE: Schließer geschlossen
DiscrepancyTime	Maximale Überwachungszeit für den Diskrepanzstatus der beiden Eingänge in ms. (DINT)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output Ergebnisse gültig sind (BOOL).
S_AntivalentOut	Ausgang der logischen Verknüpfung (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Beide Eingangskanäle sind voneinander abhängig. Die Ausgabe des Funktionsblocks zeigt das Bewertungsergebnis der beiden Kanäle.

Wenn S_AntivalentOut = TRUE und sich einer der sicherheitsrelevanten Eingänge verändert, dann schaltet der Ausgang sofort auf FALSE.

Diskrepanzzeitüberwachung: Die Diskrepanzzeit ist der maximale Zeitraum, wo beide Eingänge die gleichen Zustände haben können (z.B. beide Eingänge sind entweder TRUE oder FALSE) ohne, dass der Funktionsblock einen Fehler erkennt. Die Diskrepanzzeit Überwachung beginnt bei einer Statusänderung eines Eingangs. Der Funktionsblock erkennt einen Fehler, wenn beide Eingänge keine antivalenten Werte beim Ablauf der Diskrepanzzeit enthalten. Die Eingänge müssen symmetrisch

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

geschaltet werden. Dies bedeutet, dass eine Überwachung sowohl beim Einschalt- als auch beim Abschaltprozess durchgeführt wird.

13.5.3.1 Fehlererkennung

Der Funktionsblock überwacht die Diskrepanzzeit zwischen den Kanälen NO und NC.

13.5.3.2 Fehlerverhalten

Die Ausgabe S_AntivalentOut wird auf FALSE gesetzt. Error wird auf TRUE gesetzt ist. DiagCode zeigt die Fehlerzustände.

Es ist kein Reset-Eingang definiert, der einen Fehler zurücksetzt. Tritt ein Fehler an den Eingängen auf, dann müssen die neuen Eingangswerte mit korrekten Werten das Fehler-Flag zurücksetzen können. (Beispiel: wurde ein defekter Schalter getauscht, führt die Wiederwendung des Schalters zu einer korrekten Ausgabe)

13.5.3.3 Funktionsblock-spezifische Fehler und Status-Codes

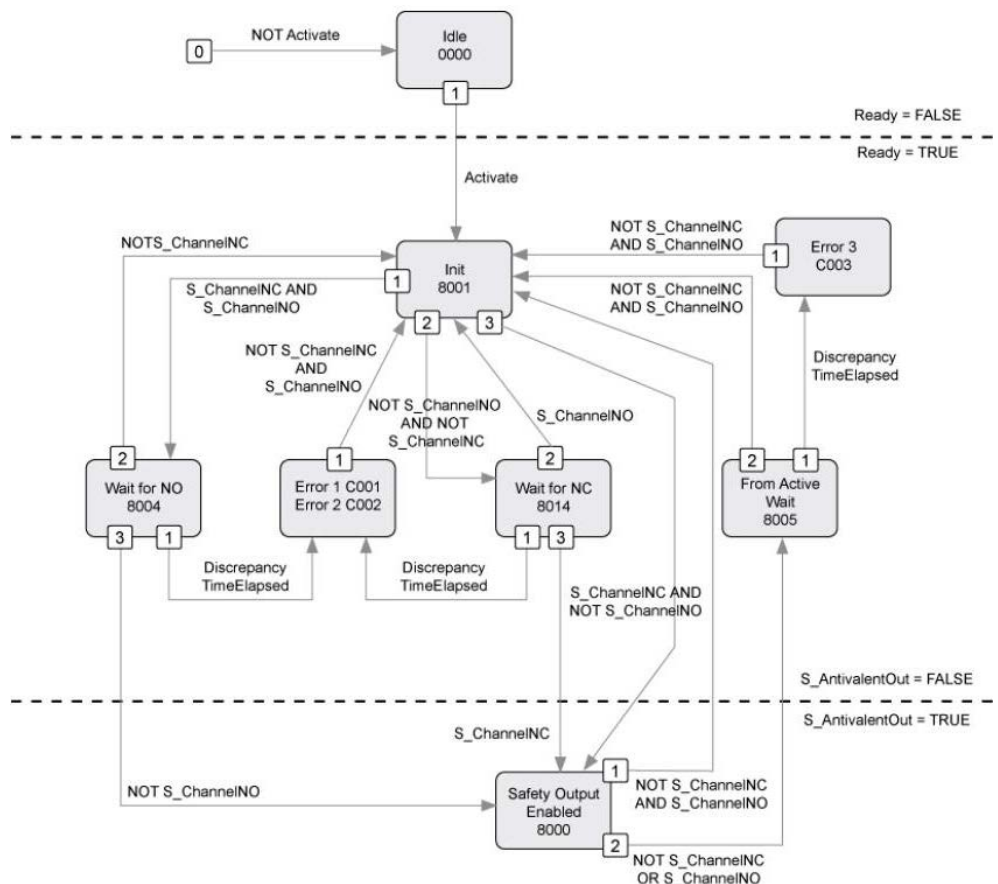
DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Error 1	Diskrepanzzeit im Zustand 8004 abgelaufen. Ready = TRUE S_AntivalentOut = FALSE Error = TRUE
C002	Error 2	Diskrepanzzeit im Zustand 8014 abgelaufen. Ready = TRUE S_AntivalentOut = FALSE Error = TRUE
C003	Error 3	Diskrepanzzeit im Zustand 8005 abgelaufen. Ready = TRUE S_AntivalentOut = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_AntivalentOut = FALSE Error = FALSE
8001	Init	Eine Aktivierung wurde vom FB erkannt und der Status ist nun aktiviert. Ready = TRUE S_AntivalentOut = FALSE Error = FALSE
8000	Safety Output Enabled	Die Eingänge sind im Aktivzustand im antivalenten Modus. Ready = TRUE S_AntivalentOut = TRUE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8004	Wait for NO	ChannelINC wurde auf TRUE geschaltet und wartet darauf, dass ChannelINO auf FALSE geschaltet wird. Diskrepanz-Timer gestartet. Ready = TRUE S_AntivalentOut = FALSE Error = FALSE
8014	Wait for NC	ChannelINO wurde auf TRUE geschaltet und wartet darauf, dass ChannelINC auf FALSE geschaltet wird. Diskrepanz-Timer gestartet. Ready = TRUE S_AntivalentOut = FALSE Error = FALSE
8005	From Active Wait	Ein Kanal wurde auf inaktiv geschaltet und wartet auf den zweiten Kanal, bis dieser auch auf inaktiv geschaltet wird. Ready = TRUE S_AntivalentOut = FALSE Error = FALSE

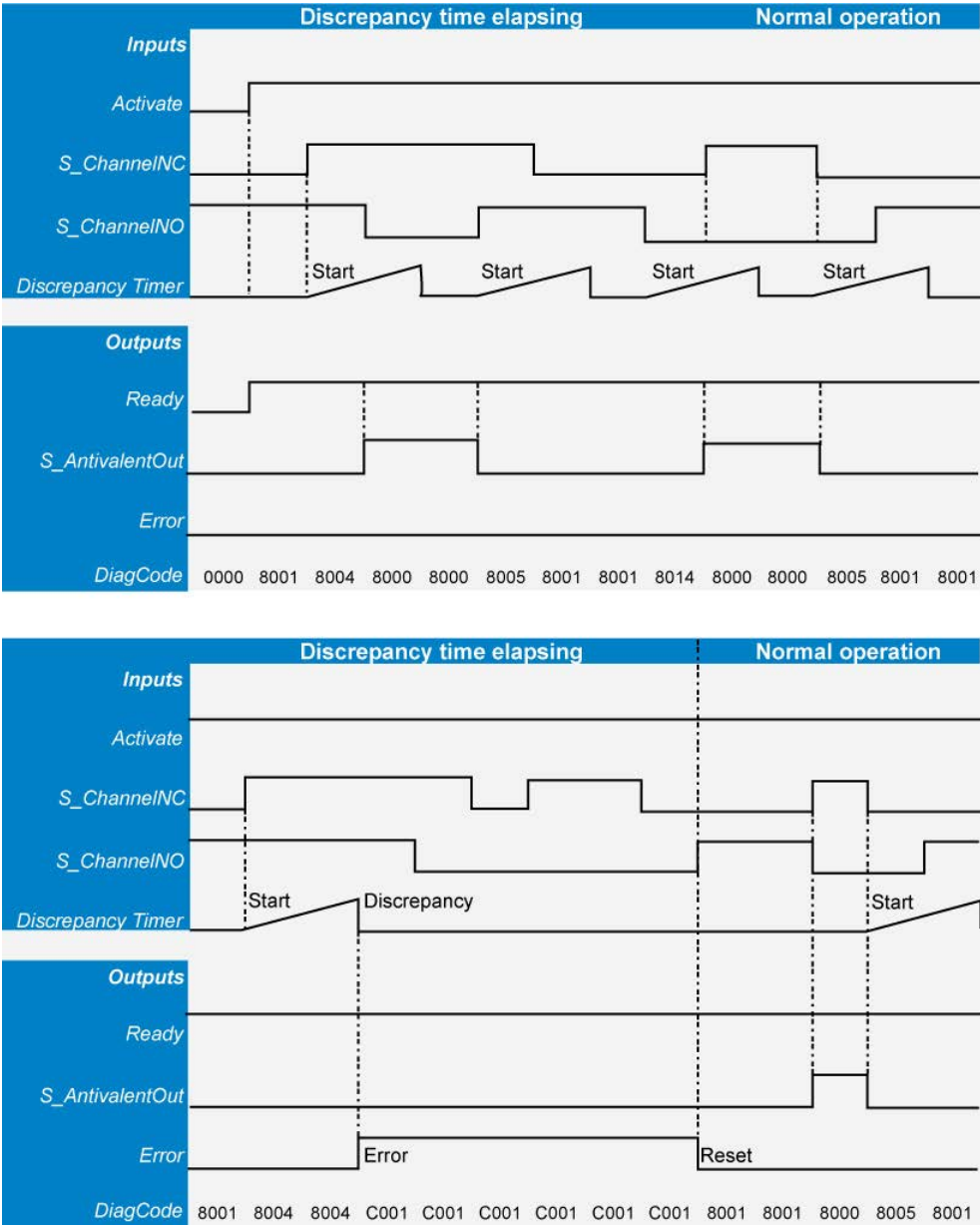
13.5.3.4 Status Diagramm SF_Antivalent



13. Funktionsblöcke

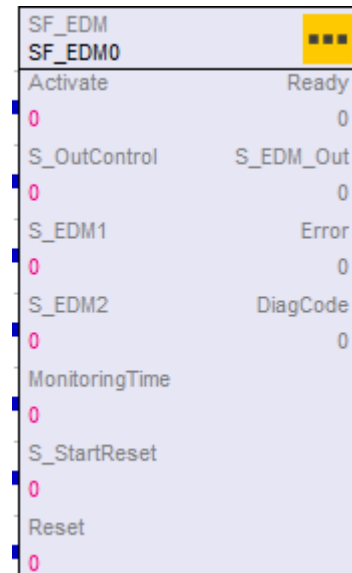
13.5 Komplexe Funktionsblöcke

13.5.3.5 Zeit-Diagramm SF_Antivalent



13.5.4 SF_EDM-Funktion

Externe Geräteüberwachung (external device monitoring). Dieser Funktionsblock kontrolliert einen Safety-Ausgang und überwacht Aktuatoren.



Eingangsvariablen

S_OutControl	Kontrollsignal von den vorherigen Safety-Funktionsblöcken. Typische Signale von folgenden FBs (z. B. SF_OutControl, SF_TwoHandControlTypell, etc.). (SAFEBOOL) FALSE: Deaktiviert den Safety-Ausgang TRUE: Aktiviert den Safety-Ausgang
S_EDM1	Feedback-Signal des ersten verbundenen Aktuators. (SAFEBOOL) FALSE: Schaltzustand des ersten verbundenen Aktuators TRUE: Ausgangszustand des ersten verbundenen Aktuators
S_EDM2	Feedback-Signal des zweiten verbundenen Aktuators. Wird nur ein Signal in der Anwendung verwendet, muss dieses Signal mit S_EDM1 und S_EDM2 verbunden werden. S_EDM1 und S_EDM2 werden dann durch das gleiche Signal kontrolliert. (SAFEBOOL) FALSE: Schaltzustand des zweiten verbundenen Aktuators TRUE: Ausgangszustand des zweiten verbundenen Aktuators
MonitoringTime	Max. Reaktionszeit der angeschlossenen und überwachten Aktuatoren (DINT)
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

AusgangsvARIABLEN

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind (BOOL)
S_EDM_Out	Kontrolliert den Aktuator. Das Ergebnis wird durch das Feedback-Signal S_EDMx überwacht. (SAFEBOOL) FALSE: Deaktiviert den angeschlossenen Aktuator TRUE: Aktiviert den angeschlossenen Aktuator

Der SF_EDM-Funktionsblock kontrolliert einen Safety-Ausgang und überwacht gesteuerte Aktuatoren.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Dieser Funktionsblock überwacht den ursprünglichen Zustand der Aktuatoren über die Feedback-Signale (S_EDM1 und S_EDM2), bevor die Aktuatoren vom FB aktiviert werden.

Der Funktionsblock überwacht den Schaltzustand der Aktuatoren (MonitoringTime), nachdem die Aktuatoren durch den FB aktiviert wurden.

Zwei einzelne Feedback-Signale müssen für eine genaue Diagnose der angeschlossenen Aktuatoren verwendet werden. Ein gemeinsames Feedback-Signal aus den beiden angeschlossenen Aktuatoren muss für eine beschränkte und dennoch einfache Diagnosefunktion der angeschlossenen Aktuatoren verwendet werden. Dabei muss der Benutzer dieses gemeinsame Signal sowohl mit dem S_EDM1-Eingang als auch mit dem S_EDM2-Eingang verbinden. S_EDM1 und S_EDM2 werden dann durch das gleiche Signal gesteuert.

Die in der Sicherheitsfunktion verwendeten Schaltgeräte sollten von der Kategorie ausgewählt werden, die in der Risikoanalyse (EN ISO 13849) festgelegt ist.

Optionale Start-Verhinderung:

- Start Verhinderung im Fall einer Blockaktivierung.

Der S_StartReset-Eingang sollte nur aktiviert werden, wenn sichergestellt ist, dass keine gefährliche Situation beim Starten auftreten kann.

13.5.4.1 Fehlererkennung

Die folgenden Bedingungen zwingen eine Umstellung auf den Fehlerstatus:

- Ungültige statisches Reset-Signal im Prozess.
- Ungültiges EDM-Signal im Prozess.
- S_OutControl und Reset sind durch einen Programmierfehler falsch verbunden.

13.5.4.2 Fehlerverhalten

In Fehlerzuständen sind die Ausgaben wie folgt:

- Im Fall eines Fehlers wird der S_EDM_Out-Ausgang auf FALSE gesetzt und bleibt in diesem sicheren Zustand.
- Eine EDM-Fehlermeldung muss immer durch einen steigenden Trigger beim Reset zurückgesetzt werden.
- Eine Reset-Fehlermeldung (Reset-Signal beim Auftreten eines Fehlers bereits ohne steigende Flanke gesetzt) kann zurückgesetzt werden, indem Reset auf FALSE gesetzt wird.

Nach der Blockaktivierung kann die optionale Start-Verhinderung durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
----------	------------	--

FB-spezifische Fehler-Codes		
-----------------------------	--	--

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C001	Reset Error 1	Statisches Reset-Signal im Stauts 8001. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C011	Reset Error 21	Statisches Reset-Signal oder die gleichen Signale an EDM1 und Reset (steigender Trigger bei Reset und EDM1 zur gleichen Zeit) im Zustand C010. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C021	Reset Error 22	Statisches Reset-Signal oder gleiche Signale an EDM2 und Reset (steigender Trigger bei Reset und EDM2 zur gleichen Zeit) im Zustand C020. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C031	Reset Error 23	Statisches Reset-Signal oder gleiche Signale an EDM1, EDM2 und Reset (steigender Trigger bei Reset, EDM1 und EDM2 zur gleichen Zeit) im Zustand C030. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C041	Reset Error 31	Statisches Reset-Signal oder gleiche Signale an EDM1 und Reset (steigender Trigger bei Reset und EDM1 zur gleichen Zeit) im Zustand C040. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C051	Reset Error 32	Statisches Reset-Signal oder gleiche Signale an EDM2 und Reset (steigender Trigger bei Reset und EDM2 zur gleichen Zeit) im Zustand C050. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C061	Reset Error 33	Statisches Reset-Signal oder gleiche Signale an EDM1, EDM2 und Reset (steigender Trigger bei Reset, EDM1 und EDM2 zur gleichen Zeit) im Zustand C060. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C071	Reset Error 41	Statisches Reset-Signal im Zustand C070. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C081	Reset Error 42	Statisches Reset-Signal im Zustand C080. Ready = TRUE S_EDM_Out = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C091	Reset Error 43	Statisches Reset-Signal im Zustand C090. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C010	EDM Error 11	Das Signal an EDM1 ist im initialen Aktuatorzustand nicht gültig. Im Zustand 8010 ist das EDM1-Signal beim Aktivieren von O_OutControl auf FALSE. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C020	EDM Error 12	Das Signal an EDM2 ist im initialen Aktuatorzustand nicht gültig. Im Zustand 8010 ist das EDM2-Signal beim Aktivieren von O_OutControl auf FALSE. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C030	EDM Error 13	Die Signale bei EDM1 und EDM2 sind im initialen Antriebszustand nicht gültig. Im Zustand 8010 sind die EDM1 und EDM2 Signale beim Aktivieren auf FALSE. O_OutControl. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C040	EDM Error 21	Das Signal an EDM1 ist im initialen Aktuatorzustand nicht gültig. Im Zustand 8010 ist das EDM1 Signal FALSE und die Überwachungszeit abgelaufen. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C050	EDM Error 22	Das Signal an EDM2 ist im initialen Aktuatorzustand nicht gültig. Im Zustand 8010 ist das EDM2 Signal FALSE und die Überwachungszeit abgelaufen. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C060	EDM Error 23	Die Signale bei EDM1 und EDM2 sind im initialen Antriebszustand nicht gültig. Im Zustand 8010 sind die EDM1- und EDM2-Signale auf FALSE und die Überwachungszeit ist abgelaufen. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C070	EDM Error 31	Das Signal an EDM1 ist im Schaltzustand des Aktuators nicht gültig. Im Zustand 8000 ist das EDM1 Signal TRUE und die Überwachungszeit abgelaufen. Ready = TRUE S_EDM_Out = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C080	EDM Error 32	Das Signal an EDM2 ist im Schaltzustand des Aktuators nicht gültig. Im Zustand 8000 ist das EDM2-Signal TRUE und die Überwachungszeit abgelaufen. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C090	EDM Error 33	Die Signale EDM1 und EDM2 sind im initialen Aktuatorzustand nicht gültig. Im Zustand 8000 sind die Signale EDM1 und EDM2 auf TRUE und die Überwachungszeit ist abgelaufen. Ready = TRUE S_EDM_Out = FALSE Error = TRUE
C111	Init Error	Ähnliche Signale wurden am S_OutControl and Reset (R_TRIG im selben Zyklus) erkannt (eventuelle Programmierungsfehler) Ready = TRUE S_EDM_Out = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler):		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_EDM_Out = FALSE Error = FALSE
8001	Init	Blockaktivierung Startup inhibit ist aktiv. Reset erforderlich. Ready = TRUE S_EDM_Out = FALSE Error = FALSE
8010	Output	Deaktivieren der EDM-Steuerung ist nicht aktiv. Timer startet, wenn Zustand der gestartet wird. Ready = TRUE S_EDM_Out = FALSE Error = FALSE
8000	Output	Aktivieren der EDM-Steuerung ist aktiv. Timer startet, wenn der Zustand gestartet wird. Ready = TRUE S_EDM_Out = TRUE Error = FALSE

13.5.4.3 Besonderheiten

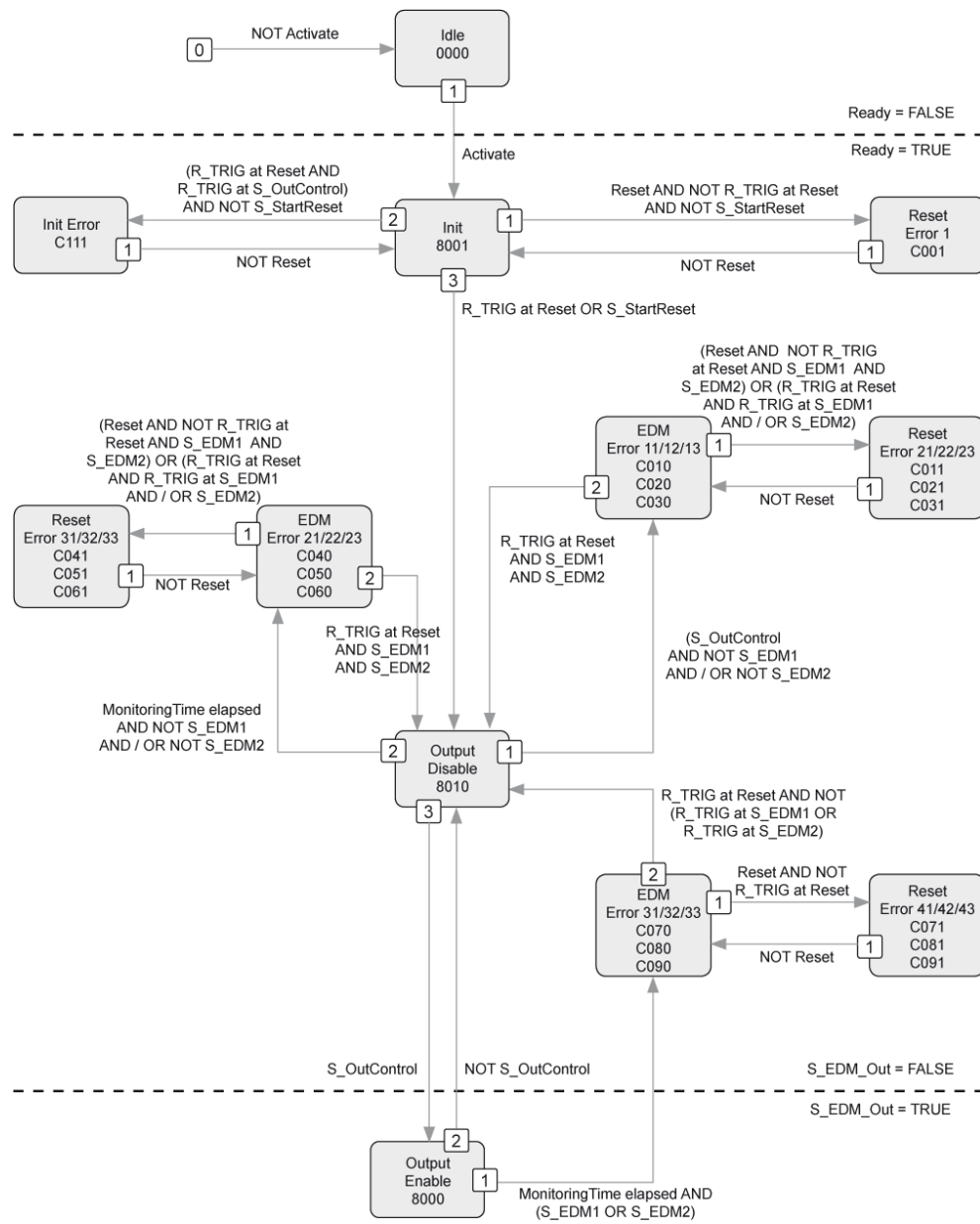
Von State 8010 zu den States C010, C020 bzw. C030 stimmen die Bedingungen im Statediagramm nicht mit den Erklärungen bei den einzelnen States überein.

Die Bedingung(en) wurden folgendermaßen festgelegt (..... AND NOT S_EDM1 AND / OR NOT S_ED M2), wie es auch bei den States C040, C050 bzw. C060 im Statediagramm beschrieben ist.

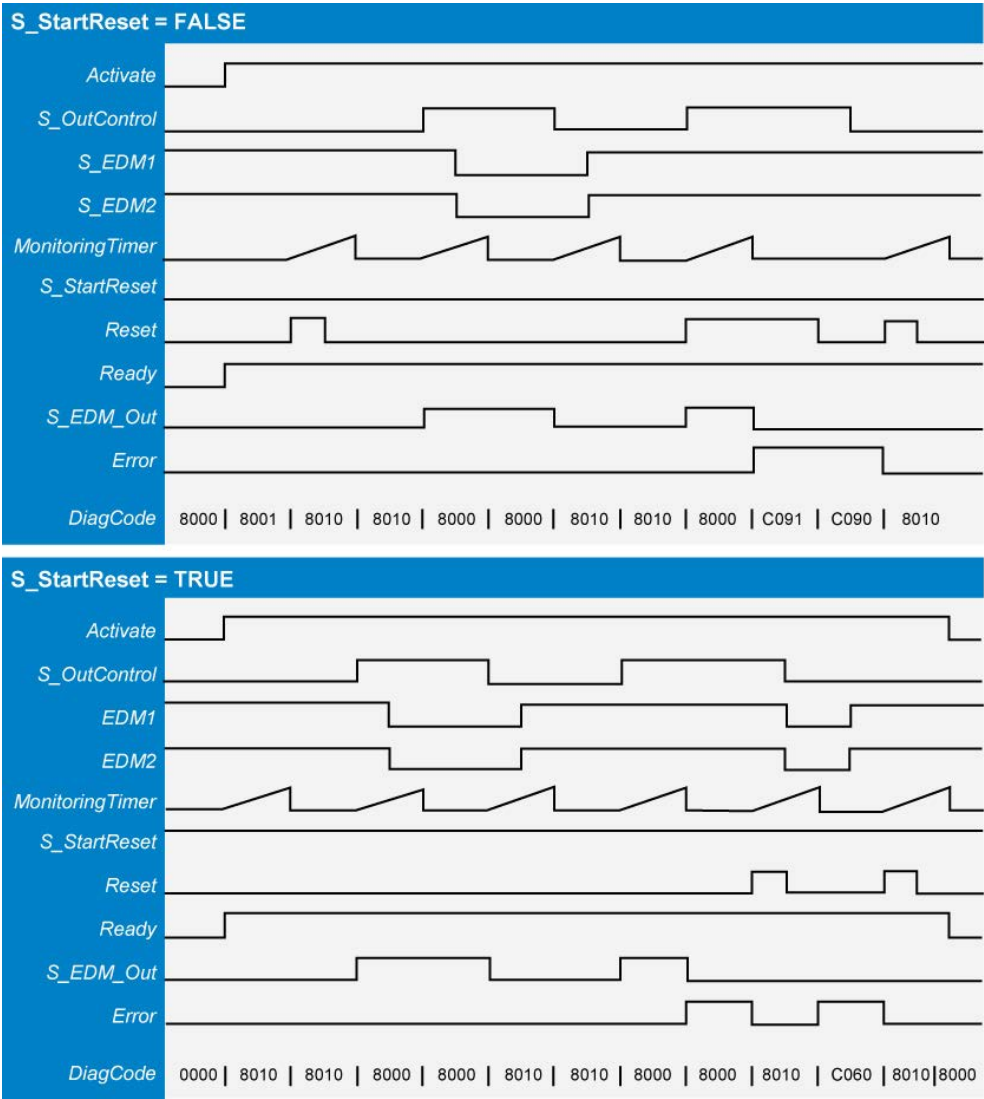
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.4.4 Status Diagramm SF_EDM



13.5.4.5 Zeit-Diagramm SF_EDM

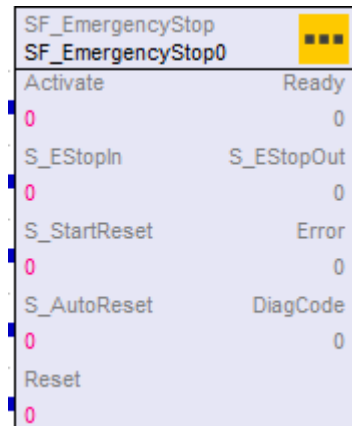


13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.5 SF_EmergencyStop-Funktion

Dieser Funktionsblock überwacht einen Not-Halt-Schalter. Er kann für die Not-Abschaltungsfunktionalität (Stopp-Kategorie 0) oder, mit zusätzlicher Peripherie-Unterstützung, wie ein Not-Halt (Stopp-Kategorie 1 oder 2) verwendet werden.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_EStopIn	Eingang Sicherheitsanfrage. (SAFEBOOL) FALSE: Anfrage zu einer Safety-Reaktion (z. B. Not-Halt-Schalter ist an) TRUE: Keine Anfrage zu einer Safety-Reaktion (z. B. Not-Halt-Schalter nicht an)
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL)
S_AutoReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der Not-Halt-Schalter wieder freigegeben wird (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind (BOOL).
S_EStopOut	Ausgabe der Safety-Reaktion. (SAFEBOOL) FALSE: Safety-Ausgang deaktiviert. Anfrage nach einer Safety-Reaktion (z. B. Not-Halt-Schalter gedrückt, Reset erforderlich oder interner Fehler aktiv) TRUE: Safety-Ausgang aktiviert. Keine Anfrage nach einer Safety-Reaktion (z. B. Not-Halt-Schalter nicht gedrückt, keine internen Fehler aktiv).
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Das S_EStopOut Enable-Signal wird wieder auf FALSE gesetzt, sobald der S_EstopIn-Eingang auf FALSE gesetzt wird. Das S_EStopOut Enable-Signal wird wieder auf TRUE gesetzt, wenn der S_EstopIn-Eingang auf TRUE gesetzt und ein Reset ausgeführt wird. Das Enable-Reset hängt vom definierten S_StartReset, S_AutoReset und den Reset-Eingängen ab.

Wenn S_AutoReset = TRUE erfolgt eine automatische Quittierung.

Wenn S_AutoReset = FALSE muss eine steigende Flanke am Reset-Eingang anliegen, um die Aktivierung zu quittieren.

Wenn S_StartReset = TRUE erfolgt die Quittierung automatisch, wenn die Steuerung zum ersten Mal gestartet wurde.

Wenn S_StartReset = FALSE muss eine steigende Flanke am Reset-Eingang anliegen, um die Aktivierung zu quittieren.

Die S_StartReset- und S_AutoReset-Eingänge sollten nur aktiviert werden, wenn sichergestellt ist, dass keine gefährliche Situation beim Starten der Steuerung auftreten kann.

SF_EmergencyStop kann verwendet werden, um sowohl Einzel- als auch Zweikanal-Not-Halt-Taster zu überwachen. Beispielsweise kann für Zweikanal-Anwendungen der zusätzliche Funktionsblock SF_Equivalent verwendet werden, um festzustellen ob die Kontakt-Synchronisation überschritten wurde. Die Einstufung in die Kategorie gemäß EN ISO 13849 ist von den verwendeten endgültigen Elementen abhängig.

SF_EmergencyStop erkennt automatisch ein statisches TRUE-Signal am Reset-Eingang. Weitere Fehlererkennungen z.B. Drahtbruch oder Kurzschluss hängen von der jeweiligen Hardware ab, die verwendet wird.

13.5.5.1 Fehlererkennung

Der Funktionsblock erkennt ein statisches TRUE Signal am Reset-Eingang.

13. Funktionsblöcke

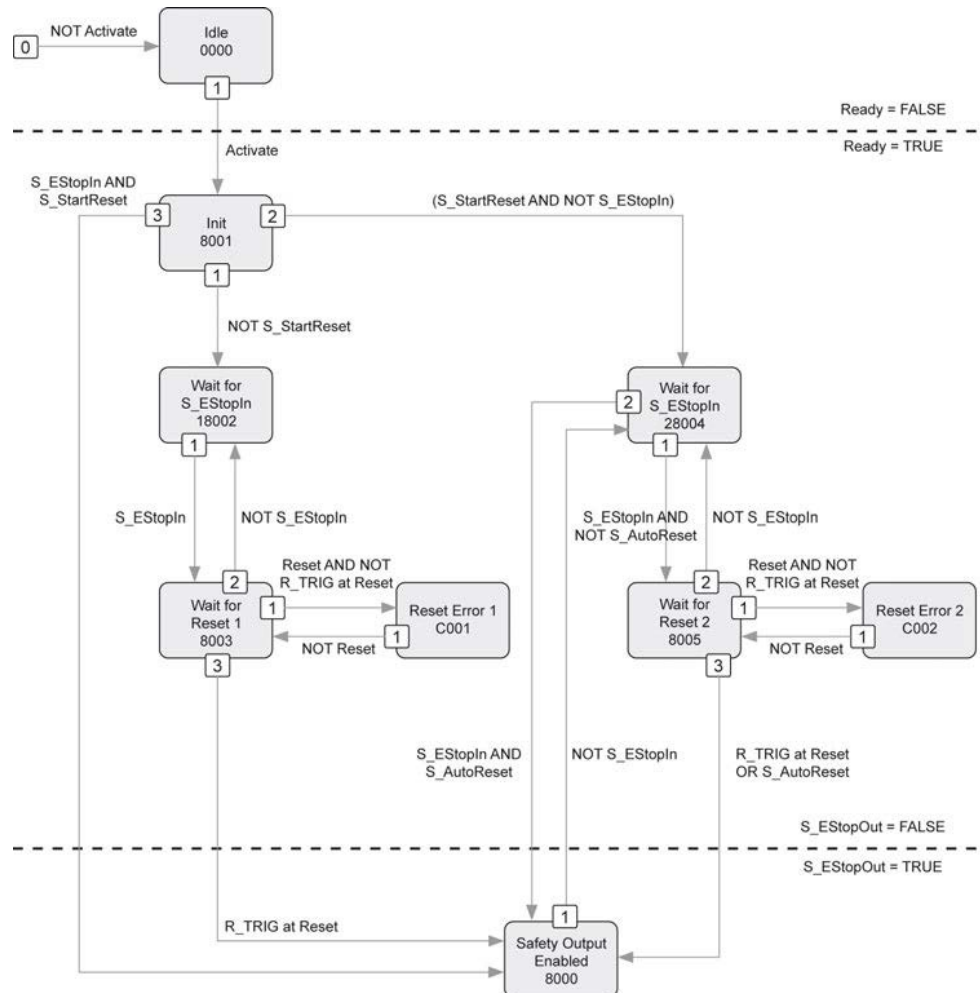
13.5 Komplexe Funktionsblöcke

13.5.5.2 Fehlerverhalten

S_EStopOut ist auf FALSE gesetzt, wenn ein statisches TRUE-Signal am Reset-Eingang anliegt. Der Ausgang DiagCode zeigt den entsprechenden Fehler-Code an und der Error-Ausgang wird auf TRUE gesetzt.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Reset ist TRUE während des Wartens auf S_EStopIn = TRUE. Ready = TRUE S_EStopOut = FALSE Error = TRUE
C002	Reset Error 2	Reset ist TRUE während des Wartens auf S_EStopIn = TRUE. Ready = TRUE S_EStopOut = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler):		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_EStopOut = FALSE Error = FALSE
8001	Init	Die Aktivierung ist TRUE. Der Funktionsblock wurde aktiviert. Überprüfen, ob S_StartReset benötigt wird. S_StartReset wird benötigt. Ready = TRUE S_EStopOut = FALSE Error = FALSE
8002	Wait for S_EstopIn 1	Die Aktivierung ist TRUE. Prüfen, ob Reset FALSE ist und warten bis S_EstopIn = TRUE. Ready = TRUE S_EStopOut = FALSE Error = FALSE
8003	Wait for Reset 1	Die Aktivierung ist TRUE. S_EstopIn = TRUE. Auf eine steigenden Flanke von Reset warten. Ready = TRUE S_EStopOut = FALSE Error = FALSE
8004	Wait for S_EstopIn 2	Die Aktivierung ist TRUE. Safety-Nachfrage festgestellt. Prüfen, ob Reset FALSE ist und warten bis S_EstopIn = TRUE. Ready = TRUE S_EStopOut = FALSE Error = FALSE
8005	Wait for Reset 2	Die Aktivierung ist TRUE. S_EstopIn = TRUE. Prüfen, ob S_AutoReset TRUE ist oder auf eine steigende Flanke von Reset warten. Ready = TRUE S_EStopOut = FALSE Error = FALSE
8000	Safety Output Enabled	Die Aktivierung ist TRUE. S_EstopIn = TRUE. Functional-Modus mit: S_EStopOut = TRUE. Ready = TRUE S_EStopOut = TRUE Error = FALSE

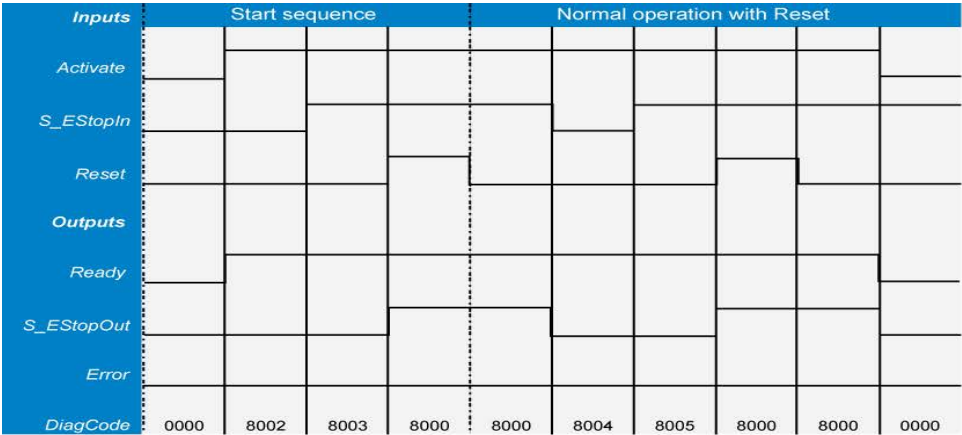
13.5.5.3 Status Diagramm SF_EmergencyStop



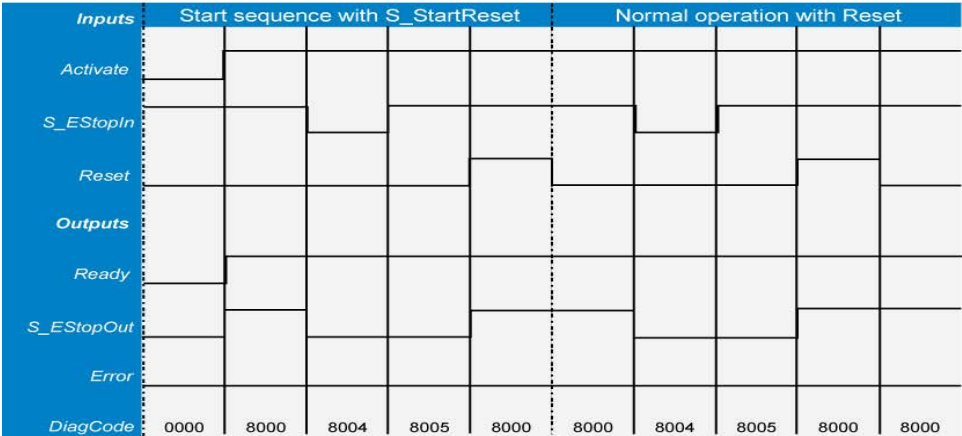
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

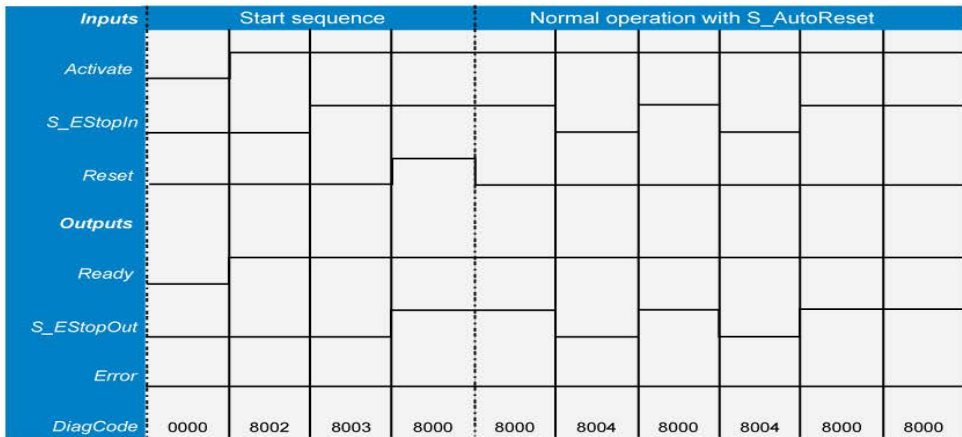
13.5.5.4 Zeit-Diagramm SF_EmergencyStop



Timing diagram for SF_EmergencyStop: S_StartReset = FALSE; S_AutoReset = FALSE; Start, reset, normal operation, safety demand, restart



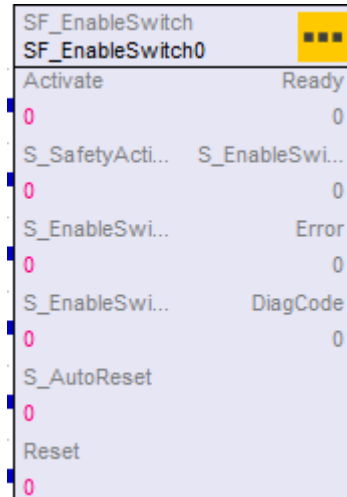
Timing diagram for SF_EmergencyStop: S_StartReset = TRUE; S_AutoReset = FALSE; Start, normal operation, safety demand, restart



Timing diagram for SF_EmergencyStop: S_StartReset = FALSE; S_AutoReset = TRUE; Start, normal operation, safety demand, restart

13.5.6 SF_EnableSwitch-Funktion

Dieser Funktionsblock wertet die Signale eines Betriebsartenschalters mit drei Stellpositionen aus (z.B. Aus/Automatik/Handbetrieb).



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_SafetyActive	Bestätigung des Safe-Modus (Begrenzung der Geschwindigkeit oder die Kraft der Bewegung, die Beschränkung der Reichweite der Bewegung). (SAFEBOOL) FALSE: Safe-Modus ist nicht aktiv TRUE: Safe-Modus ist aktiv
S_EnableSwitchCh1	Signale der Kontakte E1 und E2 des angeschlossenen Betriebsartenschalters (SAFEBOOL) FALSE: Angeschlossene Schalter sind geöffnet TRUE: Angeschlossene Schalter sind geschlossen
S_EnableSwitchCh2	Signale der Kontakte E3 und E4 des angeschlossenen Betriebsartenschalters (SAFEBOOL) FALSE: Angeschlossene Schalter sind geöffnet TRUE: Angeschlossene Schalter sind geschlossen
S_AutoReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der Not-Halt-Taster wieder freigegeben wird (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind (BOOL)
S_EnableSwitchOut	Sicherheitsrelevanter Output: Zeigt das Aussetzen der Überwachung an. (SAFEBOOL) FALSE: Deaktiviert das Aussetzen der Überwachung TRUE: Aktiviert das Aussetzen der Überwachung
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Der SF_EnableSwitch FB unterstützt die Aussetzung der Absicherung (DIN EN 60204 Abschnitt 9.2.4) mittels Aktivierungsschalter (DIN EN 60204 Abschnitt 9.2.5.8), wenn die jeweilige Betriebsart ausgewählt und aktiv ist. Die jeweilige Betriebsart

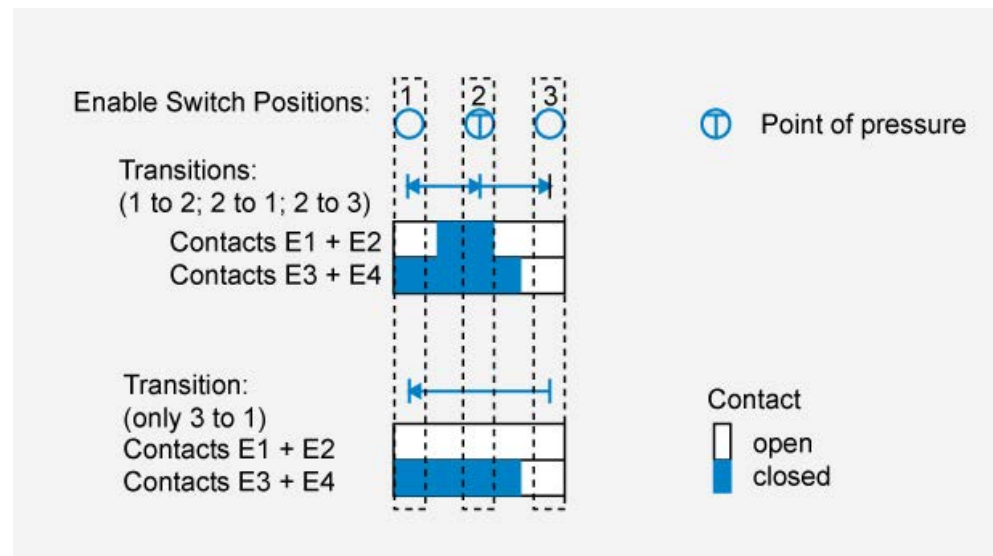
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

(Geschwindigkeits- oder Bewegungskraftbegrenzung, die Beschränkung der Bewegungsfreiheit) müssen außerhalb des FB SF_EnableSwitch ausgewählt werden.

Der SF_EnableSwitch FB bewertet die Signale eines aktivierten Schalters mit drei Stellungen (DIN EN 60204 Abschnitt 9.2.5.8).

Die S_EnableSwitchCh1 und S_EnableSwitchCh2-Eingangsparameter verarbeiten die Kontaktzustände von E1 bis E4:



Die Signale von E1+E2 müssen an den Eingang S_EnableSwitchCh1 angeschlossen werden. Die Signale von E3+E4 müssen an den Eingang S_EnableSwitchCh2 angeschlossen werden. Die Position des Enable-Schalters wird im FB erkannt, der diese Signalfolge verwendet. Der Übergang von der Position 2 bis 3 kann anders als hier dargestellt sein.

Die Schaltrichtung (Position 1 => Position 2/Position 3 => Position 2) kann im FB mit der vorgegebenen Signalfolge der Enable-Schaltkontakte erkannt werden. Die Aussetzung der Absicherung kann nur nach einer Bewegung von Position 1 auf Position 2 durch den FB aktiviert werden. Andere Schaltrichtungen oder Positionen dürfen nicht zur Aktivierung der Aussetzung der Sicherung verwendet werden. Diese Maßnahme entspricht den Anforderungen der EN 60204 Abschnitt 9.2.5.8.

Um die Anforderungen von DIN EN 60204 Abschnitt 9.2.4 zu erfüllen, muss der Anwender ein passendes Schaltgerät verwenden. Darüber hinaus muss der Anwender sicherstellen, dass die jeweilige Betriebsart (DIN EN 60204 Abschnitt 9.2.3) in der Anwendung ausgewählt wird (der automatische Betrieb muss in dieser Betriebsart mittels geeigneten Maßnahmen deaktiviert werden).

Die Betriebsart wird in der Regel von einem Betriebsartenschalter in Verbindung mit dem SF_ModeSelector-FB und dem SF_SafeRequest-FB oder dem SF_SafelyLimitedSpeed-FB angegeben.

Der SF_EnableSwitch-FB verarbeitet die Bestätigung des "safe mode" Zustands über den Eingang "S_SafetyActive". Bei der Umsetzung in eine Anwendung des

abgesicherten Modus ohne Bestätigung wird ein statisches TRUE-Signal an den Eingang "S_SafetyActive" verbunden.

Der Eingang S_AutoReset sollte nur aktiviert werden, wenn sichergestellt ist, dass keine gefährliche Situation beim Starten der PES auftreten kann.

13.5.6.1 Fehlererkennung

Die folgenden Bedingungen zwingen eine Umstellung auf den Fehlerstatus:

- Ungültiges statische Reset-Signal im Prozess.
- Ungültige Schalterstellungen.

13.5.6.2 Fehlerverhalten

Im Falle eines Fehlers wird der sichere Ausgang S_EnableSwitchOut auf FALSE gesetzt und bleibt in diesem sicheren Zustand. Anders als bei anderen FBs, kann ein Reset-Fehler-Zustand durch die Bedingung Reset = FALSE belassen werden oder auch, wenn das Signal S_SafetyActive FALSE ist.

Nachdem der Fehler erfolgreich behoben wurde, muss der Schalter auf die Anfangsposition gestellt werden, die im Prozess festgelegt wurde, bevor man mit dem Aktivierungsschalter den Ausgang S_EnableSwitchOut auf TRUE setzen kann. Wenn S_AutoReset = FALSE, ist eine steigende Flanke am Reset-Eingang erforderlich.

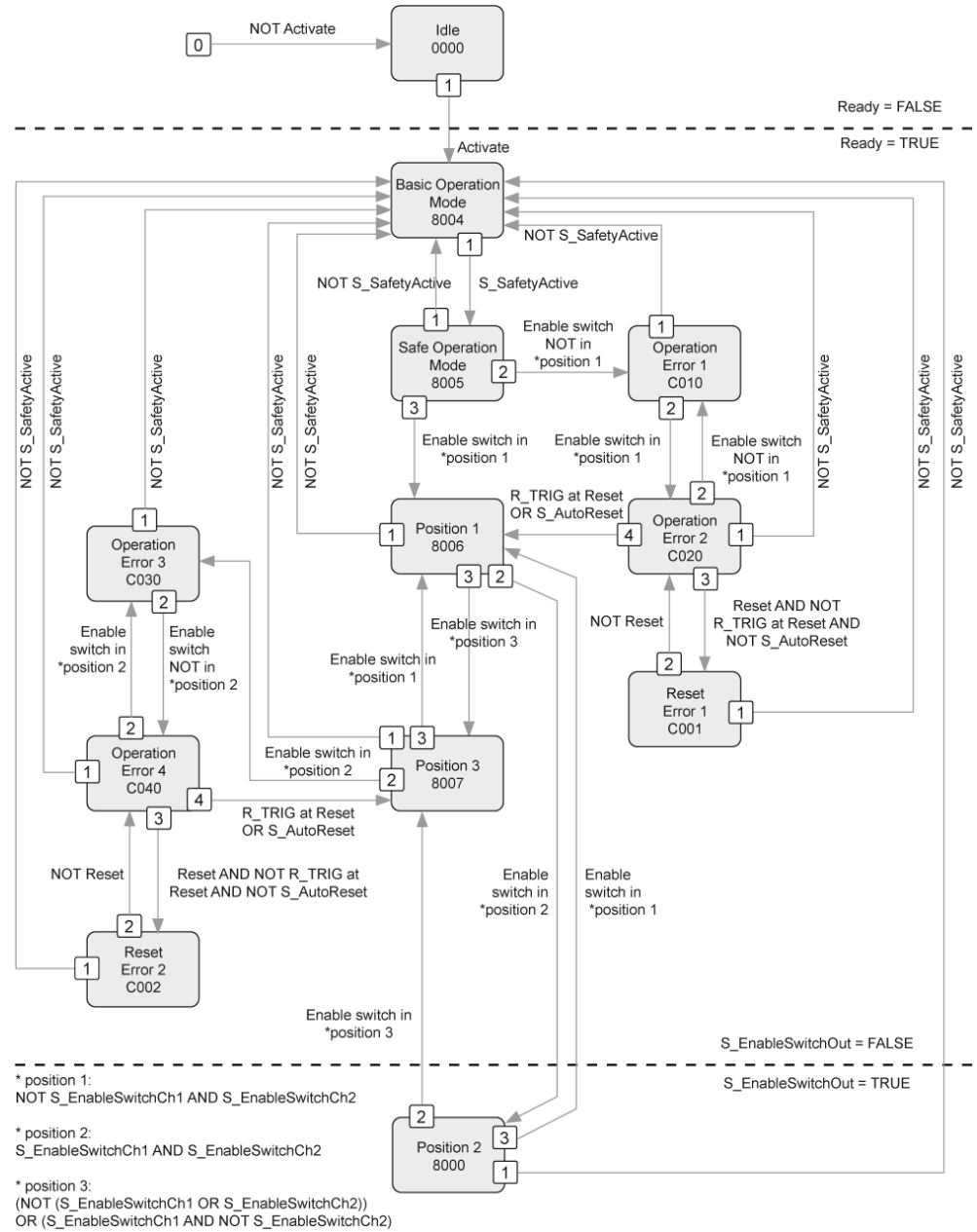
DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal wurde im Zustand C020 erkannt. Ready = TRUE S_EnableSwitchOut = FALSE Error = TRUE
C002	Reset Error 2	Statisches Reset-Signal wurde im Zustand C040 erkannt. Ready = TRUE S_EnableSwitchOut = FALSE Error = TRUE
C010	Operation Error 1	Aktivierungsschalter bei der Aktivierung von S_SafetyActive nicht auf Position 1. Ready = TRUE S_EnableSwitchOut = FALSE Error = TRUE
C020	Operation Error 2	Aktivierungsschalter nach C010 auf Position 1. Ready = TRUE S_EnableSwitchOut = FALSE Error = TRUE
C030	Operation Error 3	Aktivierungsschalter auf Position 2 nach Position 3. Ready = TRUE S_EnableSwitchOut = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C040	Operation Error 4	Aktivierungsschalter nach C030 nicht auf Position 2. Ready = TRUE S_EnableSwitchOut = FALSE Error = TRUE
FB-spezifische Fehler-Codes (kein Fehler):		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_EnableSwitchOut = FALSE Error = FALSE
8004	Basic Operation Mode	Sicherer Betriebsmodus ist nicht aktiv. Ready = TRUE S_EnableSwitchOut = FALSE Error = FALSE
8005	Safe Operation Mode	Sicherer Betriebsmodus ist aktiv. Ready = TRUE S_EnableSwitchOut = FALSE Error = FALSE
8006	Position 1	Sicherer Betriebsmodus ist aktiv und der Aktivierungsschalter ist auf Position 1. Ready = TRUE S_EnableSwitchOut = FALSE Error = FALSE
8007	Position 3	Sicherer Betriebsmodus ist aktiv und der Aktivierungsschalter ist auf Position 3. Ready = TRUE S_EnableSwitchOut = FALSE Error = FALSE
8000	Position 2	Sicherer Betriebsmodus ist aktiv und der Aktivierungsschalter ist auf Position 2. Ready = TRUE S_EnableSwitchOut = TRUE Error = FALSE

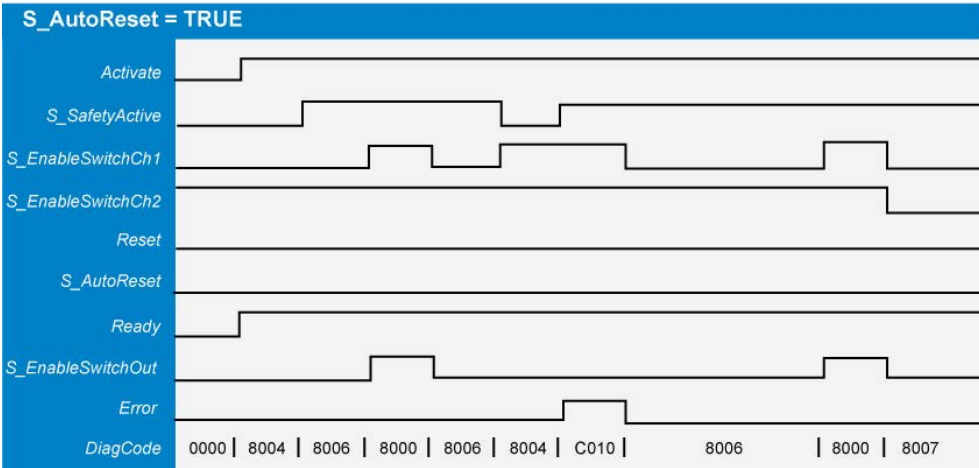
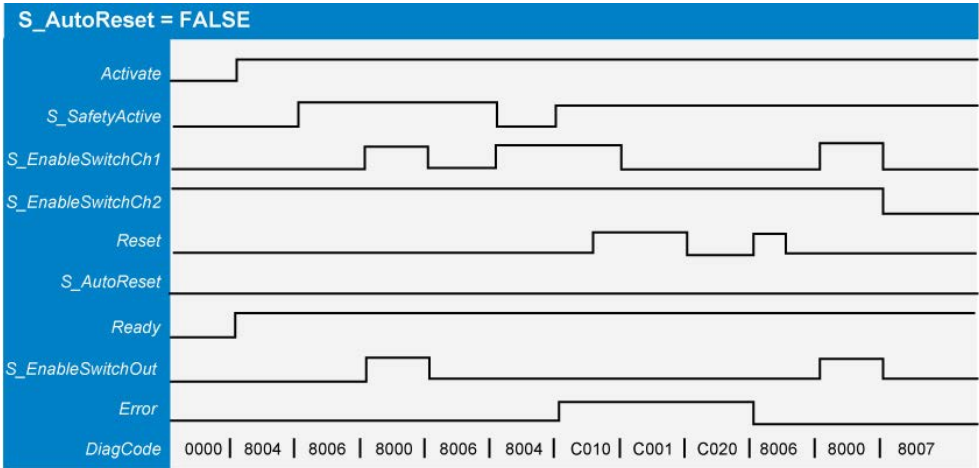
13.5.6.3 Status Diagramm SF_EnableSwitch



13. Funktionsblöcke

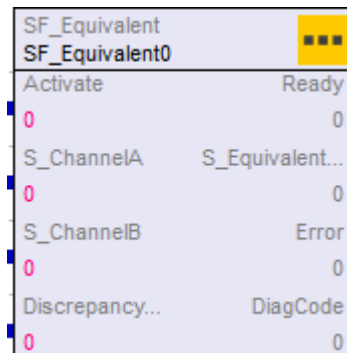
13.5 Komplexe Funktionsblöcke

13.5.6.4 Zeit Diagramm SF_EnableSwitch



13.5.7 SF_Equivalent-Funktion

Dieser Funktionsblock verknüpft zwei gleichwertige SAFEBOOL-Eingänge (beide NO oder NC) zu einem SAFEBOOL-Ausgang mit Diskrepanzzeit Überwachung. Dieser Funktionsblock sollte nicht als stand-alone verwendet werden, da keine Wiederanlaufsperrung vorhanden ist. Es ist erforderlich den Ausgang mit anderen sicherheitsrelevanten Funktionen zu verknüpfen.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_ChannelA	Eingang A für die logische Verknüpfung. (SAFEBOOL) FALSE: Kontakt A geöffnet TRUE: Kontakt A geschlossen
S_ChannelB	Eingang B für die logische Verknüpfung. (SAFEBOOL) FALSE: Kontakt B geöffnet TRUE: Kontakt B geschlossen
DiscrepancyTime	Maximale Überwachungszeit für den Diskrepanzstatus der beiden Eingänge in ms. (DINT)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output Ergebnisse gültig sind (BOOL)
S_EquivalentOut	Ausgang der logischen Verknüpfung (SAFEBOOL) FALSE: Mindestens ein Eingangssignal = „false“ oder eine Statusänderung außerhalb der kontrollierten Zeit. TRUE: Beide Eingangssignale sind „aktiv“ und eine Statusänderung innerhalb der kontrollierten Zeit.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Wechselt ein Kanalsignal von TRUE auf FALSE, schaltet der Ausgang aus Sicherheitsgründen sofort ab (FALSE).

Diskrepanzzeitüberwachung: Die Diskrepanzzeit ist der maximale Zeitraum, währenddessen die beiden Eingänge unterschiedliche Zustände haben dürfen, ohne dass der Funktionsblock einen Fehler erkennt. Die Diskrepanzzeit Überwachung beginnt bei einer Eingangsstatusveränderung. Der Funktionsblock erkennt einen Fehler, wenn beide Eingänge beim Ablauf der Diskrepanzzeit nicht den gleichen Zustand haben.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Die Eingänge müssen symmetrisch geschaltet werden. Dies bedeutet, dass eine Überwachung sowohl für den Einschalt- als auch Abschaltprozess durchgeführt wird.

13.5.7.1 Fehlererkennung

Der Funktionsblock überwacht beim Umschalten auf TRUE und FALSE die Diskrepanzzeit zwischen Kanal A und B.

13.5.7.2 Fehlerverhalten

Es ist kein Reset-Eingang definiert, der einen Fehler zurücksetzt. Tritt ein Fehler an den Eingängen auf, müssen die neuen Eingangswerte mit korrekten Werten das Fehler-Flag zurücksetzen können. (Beispiel: wurde ein defekter Schalter getauscht, führt die Wiederwendung des Schalters zu einer korrekten Ausgabe)

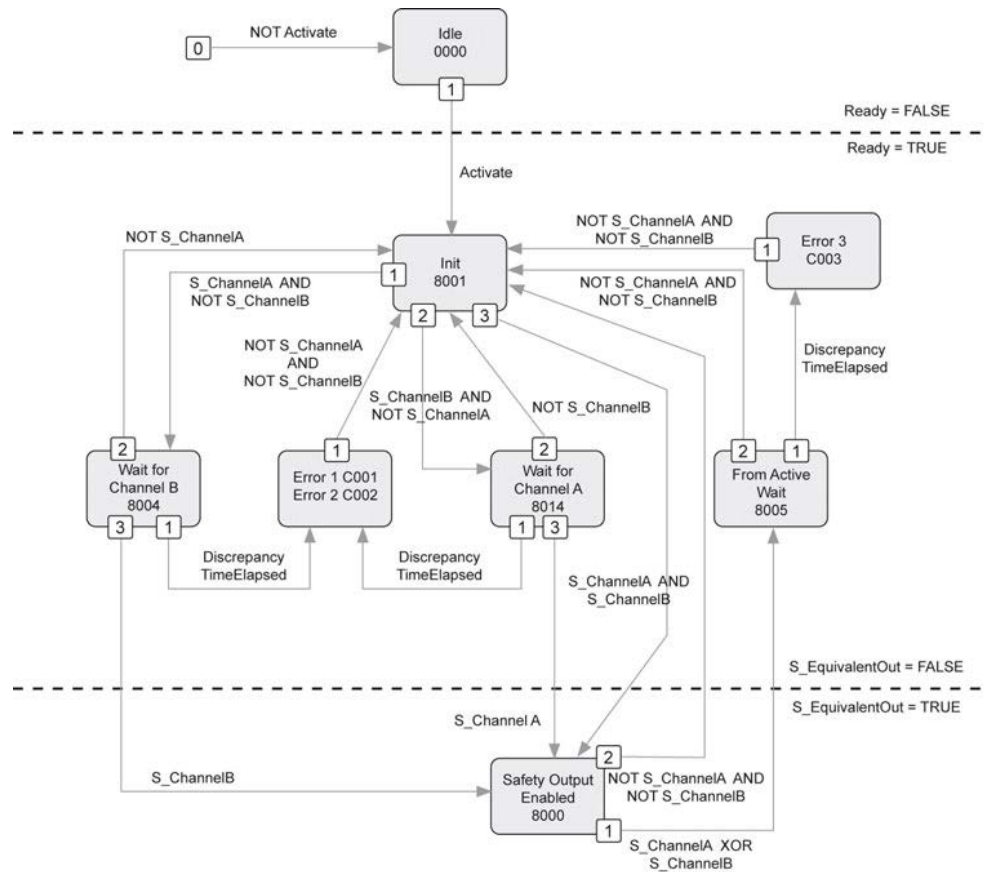
DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Error 1	Diskrepanzzeit im Zustand 8004 abgelaufen. Ready = TRUE S_EquivalentOut = FALSE Error = TRUE
C002	Error 2	Diskrepanzzeit im Zustand 8014 abgelaufen. Ready = TRUE S_EquivalentOut = FALSE Error = TRUE
C003	Error 3	Diskrepanzzeit ist im Zustand 8005 abgelaufen. Ready = TRUE S_EquivalentOut = FALSE Error = TRUE
FB-specific status codes (no error)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_EquivalentOut = FALSE Error = FALSE
8001	Init	Eine Aktivierung wurde vom FB erkannt und ist nun aktiviert. Ready = TRUE S_EquivalentOut = FALSE Error = FALSE
8000	Safety Output Enabled	Die Eingänge sind im entsprechenden Modus auf TRUE gesetzt. Ready = TRUE S_EquivalentOut = TRUE Error = FALSE
8004	Wait for Channel B	Kanal A wurde auf TRUE gesetzt und wartet auf Kanal B; Der Diskrepanz-Timer wurde gestartet. Ready = TRUE S_EquivalentOut = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8014	Wait for Channel A	Kanal B wurde auf TRUE gesetzt und wartet auf Kanal A; Der Dis- krepanz-Timer wurde gestartet. Ready = TRUE S_EquivalentOut = FALSE Error = FALSE
8005	From Active Wait	Ein Kanal wurde auf FALSE gesetzt und wartet nur auf den zweiten Kanal, dass dieser auch auf FALSE geschalten wird; Der Dis- krepanz-Timer wurde gestartet. Ready = TRUE S_EquivalentOut = FALSE Error = FALSE

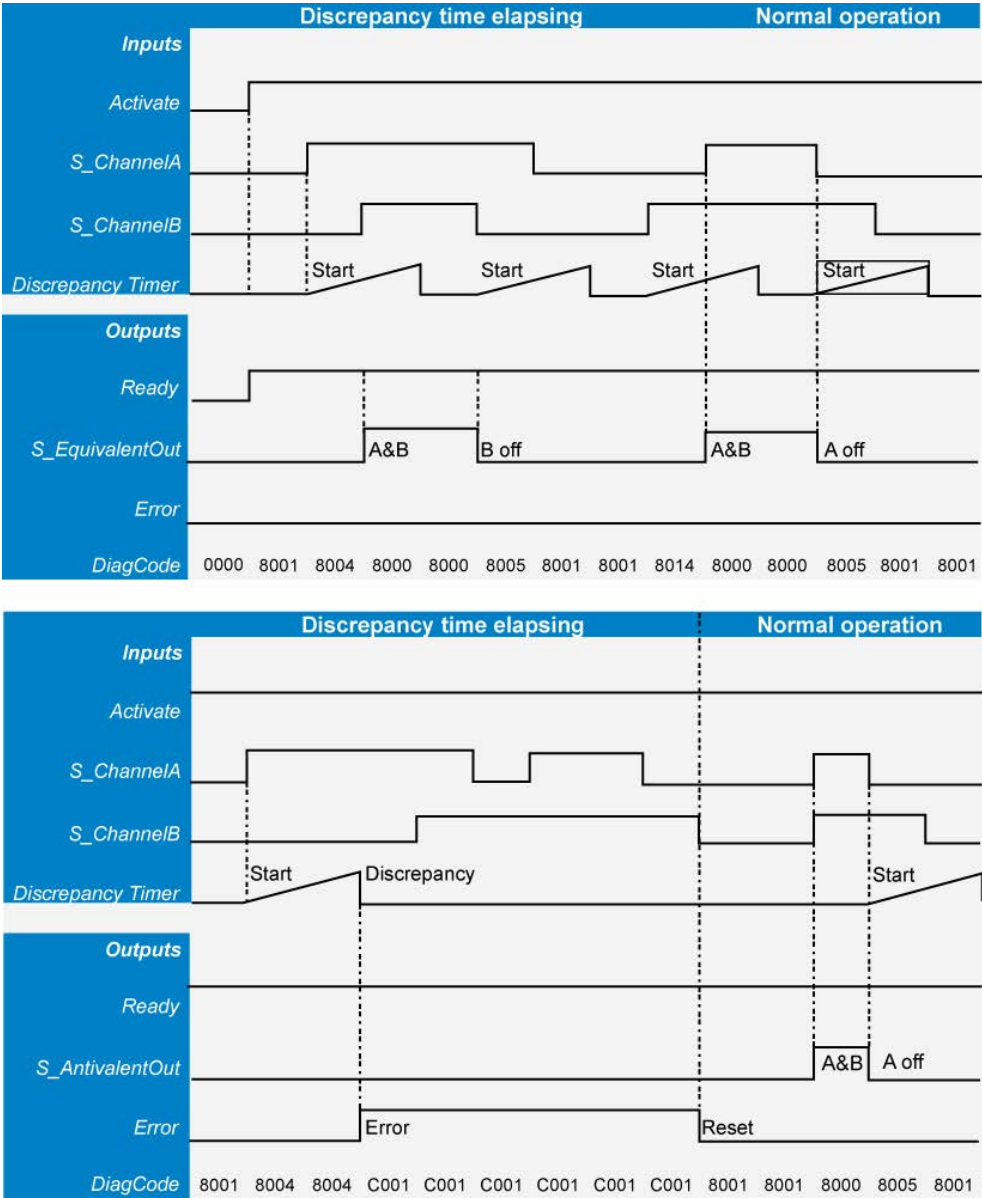
13.5.7.3 Status Diagramm SF_Equivalent



13. Funktionsblöcke

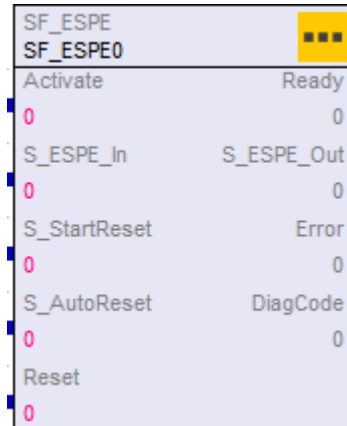
13.5 Komplexe Funktionsblöcke

13.5.7.4 Zeit-Diagramm SF_Equivalent



13.5.8 SF_ESPE-Funktion

Dieser Funktionsblock dient zur Überwachung von electro-sensitive protective equipment (ESPE).



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_ESPE_In	Eingang Sicherheitanfrage. (SAFEBOOL) FALSE: ESPE aktiv, Anfrage nach einer Safety-Reaktion. TRUE: ESPE nicht aktiv, keine Anfrage nach einer Safety-Reaktion. Das Safety-System muss in der Lage sein, sehr kurze Unterbrechungen des Sensors zu erkennen (mindestens 80 ms: spezifiziert in der Norm 61496-1)
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL).
S_AutoReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der Not-Halt-Schalter wieder freigegeben wird. (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output Ergebnisse gültig sind (BOOL)
S_ESPE_Out	Ausgabe der Safety-Reaktion. (SAFEBOOL) FALSE: Safety-Ausgang deaktiviert. Anfrage nach einer Safety-Reaktion. TRUE: Safety-Ausgang aktiviert. Keine Anfrage nach einer Safety-Reaktion.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Dieser Funktionsblock ist ein sicherheitsrelevanter Funktionsblock zur Überwachung der elektro-empfindlichen Schutzeinrichtungen (ESPE). Die Funktion ist identisch mit SF_EmergencyStop. Das Ausgangssignal S_ESPE_Out wird auf FALSE gesetzt, sobald der Eingang S_ESPE_In auf FALSE gesetzt wird. Das Ausgangssignal S_ESPE_Out wird auf TRUE gesetzt, wenn der Eingang S_ESPE_In auf TRUE gesetzt und ein Reset ausgeführt wird. Das Enable-Reset hängt von den definierten S_StartReset-, S_AutoReset- und Reset-Eingängen ab.

Ist S_AutoReset = TRUE, dann erfolgt die Quittierung automatisch.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Wenn S_AutoReset = FALSE, muss eine steigende Flanke am Reset-Eingang anliegen, um die Aktivierung zu quittieren.

Wenn S_StartReset = TRUE, erfolgt die Quittierung automatisch, wenn der PES zum ersten Mal gestartet wird.

Wenn S_StartReset = FALSE, muss eine steigende Flanke am Reset-Eingang anliegen, um die Aktivierung zu quittieren.

Die S_StartReset und S_AutoReset-Eingänge sollten nur aktiviert werden, wenn sichergestellt ist, dass keine gefährliche Situation beim Starten der PES auftreten kann.

Das ESPE muss im Bezug auf die Produktnormen EN IEC 61496-1, -2 und -3 und die erforderlichen Kategorien nach EN ISO 13849 ausgewählt werden.

13.5.8.1 Fehlererkennung

Der Funktionsblock erkennt ein statisches TRUE-Signal am Reset-Eingang.

13.5.8.2 Fehlerverhalten

S_ESPE_Out ist auf FALSE gesetzt, wenn ein statisches TRUE-Signal am Reset-Eingang anliegt. Die Ausgang DiagCode zeigt den entsprechenden Fehler-Code an und der Error-Ausgang wird auf TRUE gesetzt.

Um den Fehlerzustand zu verlassen, muss Reset auf "FALSE" gesetzt werden.

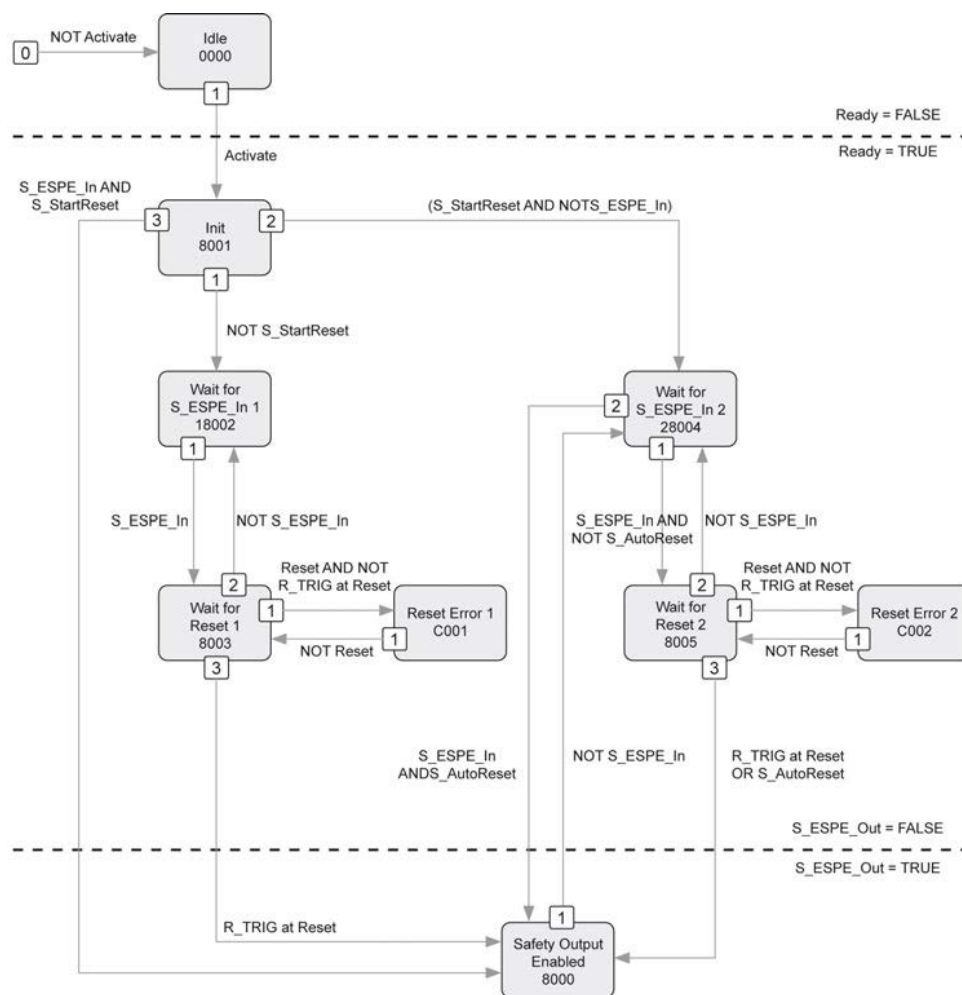
DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Reset ist TRUE beim Warten auf S_ESPE_In = TRUE. Ready = TRUE S_ESPE_Out = FALSE Error = TRUE
C002	Reset Error 2	Reset ist TRUE beim Warten auf S_ESPE_In = TRUE. Ready = TRUE S_ESPE_Out = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_ESPE_Out = FALSE Error = FALSE
8001	Init	Die Aktivierung ist TRUE. Der Funktionsblock wurde aktiviert. Prüfen Sie, ob S_StartReset erforderlich ist. Ready = TRUE S_ESPE_Out = FALSE Error = FALSE
8002	Wait for S_ESPE_In 1	Die Aktivierung ist TRUE. Prüfen Sie, ob Reset FALSE ist und warten Sie auf S_ESPE_In = TRUE. Ready = TRUE S_ESPE_Out = FALSE Error = FALSE
8003	Wait for Reset 1	Die Aktivierung ist TRUE. S_ESPE_In = TRUE. Auf einen steigenden Trigger von Reset warten. Ready = TRUE S_ESPE_Out = FALSE Error = FALSE
8004	Wait for S_ESPE_In 2	Die Aktivierung ist TRUE. Safety-Anfrage festgestellt. Prüfen Sie, ob Reset FALSE ist, und warten Sie auf S_ESPE_In = TRUE. Ready = TRUE S_ESPE_Out = FALSE Error = FALSE
8005	Wait for Reset 2	Die Aktivierung ist TRUE. S_ESPE_In = TRUE. Prüfen Sie, ob S_AutoReset TRUE ist oder auf eine steigende Flanke von Reset warten. Ready = TRUE S_ESPE_Out = FALSE Error = FALSE

13. Funktionsblöcke

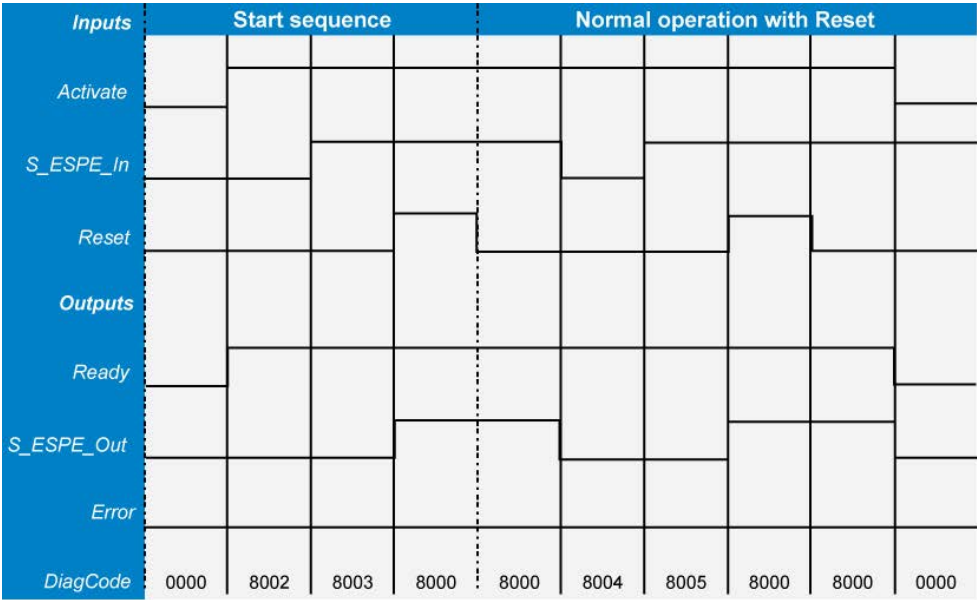
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8000	Safety Output Enabled	Die Aktivierung ist TRUE. S_ESPE_In = TRUE. Functional-Modus mit S_ESPE_Out = TRUE Ready = TRUE S_ESPE_Out = TRUE Error = FALSE

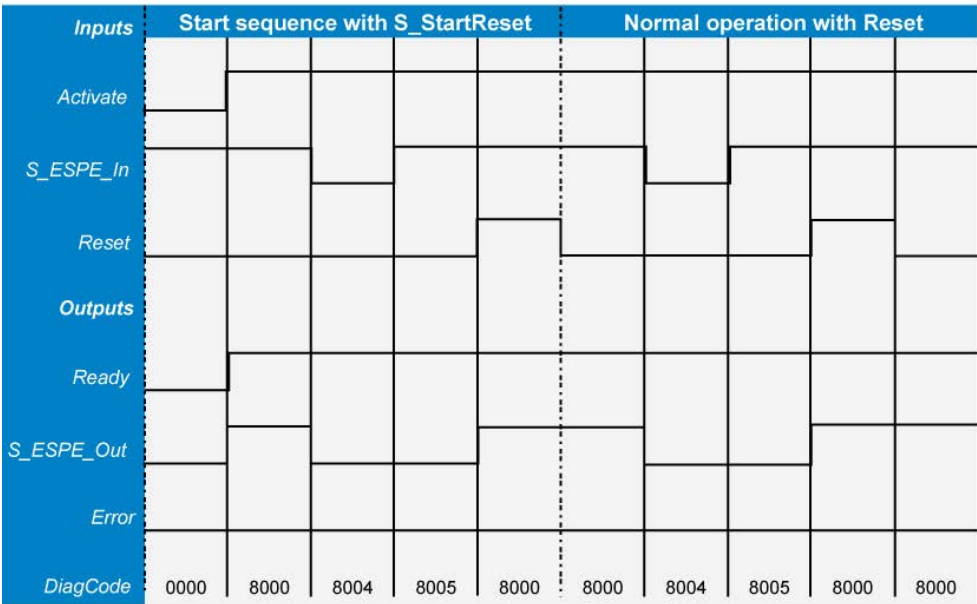
13.5.8.3 Status Diagramm SF_ESPE



13.5.8.4 Zeit-Diagramm SF_ESPE



Timing diagram for SF_ESPE: S_StartReset = FALSE; S_AutoReset = FALSE; Start, reset, normal operation, safety demand, restart



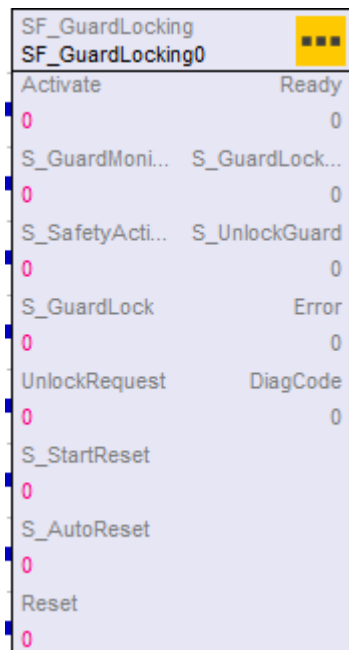
Timing diagram for SF_ESPE: S_StartReset = TRUE, S_AutoReset = FALSE; Start, normal operation, safety demand, restart

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.9 SF_GuardLocking-Funktion

Dieser Funktionsblock kontrolliert den Zugang zu einem Gefahrenbereich über ein verriegelbares Schutzgitter.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock. (BOOL)
S_GuardMonitoring	Überwacht das Schutzgitter. (SAFEBOOL) FALSE: Schutzgitter offen. TRUE: Schutzgitter geschlossen.
S_SafetyActive	Status des Gefahrenbereiches, z. B. auf Grundlage der Überwachung der Geschwindigkeit oder der sicheren Zeitverzögerung. FALSE: Maschine im „nicht sicheren“ Zustand TRUE: Maschine im sicheren Zustand
S_GuardLock	Status der mechanischen Schutzgitter-Verriegelung. FALSE: Schutzgitter ist nicht verriegelt. TRUE: Schutzgitter ist verriegelt.
UnlockRequest	Operator Anfrage zum Entsperrn des Gitters. FALSE: Keine Anfrage TRUE: Anfrage
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird. (SAFEBOOL)
S_AutoReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der Not-Halt-Schalter wieder freigegeben wird. (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind (BOOL)
-------	--

S_GuardLocked	Schnittstelle zum Gefahrenbereich in dem gestoppt werden muss. (SAFEBOOL) FALSE: Kein sicherer Zustand TRUE: Sicherer Zustand
S_UnlockGuard	Signal zum Entriegeln des Schutzgitters (SAFEBOOL) FALSE: Schutzgitter schließen TRUE: Schutzgitter entsperren
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Die Funktion steuert die Schutzverriegelung und überwacht die Position des Schutzes und der Verriegelung. Dieser Funktionsblock kann mit einem mechanischen Schalter gesperrt werden.

Der Anwender verlangt Zugang zum Gefahrenbereich. Die Schutzeinrichtung kann nur dann freigeschaltet werden, wenn der Gefahrenbereich sich in einem sicheren Zustand befindet. Die Schutzeinrichtung kann gesperrt werden, wenn die Schutztür geschlossen ist.

Die Maschine kann gestartet werden, wenn die Schutztür geschlossen und gesperrt ist. Eine offene oder ungesperrte Schutztür wird im Falle einer sicherheitskritischen Situation erkannt.

Die S_StartReset- und S_AutoReset-Eingänge sollten nur aktiviert werden, wenn sichergestellt ist, dass keine gefährliche Situation beim Starten der PES auftreten kann.

13.5.9.1 Betriebssequenz

1. Anfrage, um einen Gefahrenbereich in einen sicheren Zustand zu bringen - nicht Teil dieses FB.
2. Feedback vom geltenden Gefahrenbereich, dass dieser sich in einem sicheren Zustand befindet (über S_SafetyActive).
3. Anfrage vom Operator zum Entsperren der Schutztür (über UnlockRequest).
4. Vom Enable Guard, der geöffnet werden soll (über S_UnlockGuard).
5. Schutztür aufgesperrt (über S_GuardLock). Die Schutztür kann jetzt geöffnet werden. (S_GuardLocked = FALSE).
6. Der Operator öffnet die Schutztür.
7. Bei der Überwachung des Status der Schutztür über S_GuardMonitoring wird signalisiert, wenn die Schutztür wieder geschlossen ist.
8. Feedback vom Operator zum Neustarten des Gefahrenbereichs (Reset).
9. Lock Out Guard (S_UnlockGuard).
10. Überprüfen ob die Schutztür gesperrt ist (S_GuardLock).
11. Gefahrenbereich verlassen, es kann wieder gestartet werden (S_GuardLocked = TRUE).
12. Neustart des Vorgangs im Gefahrenbereich.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.9.2 Fehlererkennung

Statische Signale werden beim Reset erkannt. Fehler werden beim Schutzschalter erkannt.

13.5.9.3 Fehlerverhalten

Bei einem Fehler werden S_GuardLocked und S_UnlockGuard-Ausgänge auf FALSE gesetzt. Der Ausgang DiagCode gibt den entsprechenden Fehler-Code aus und Error wird auf TRUE gesetzt. Ein Fehler muss durch eine steigende Flanke am Reset-Eingang quittiert werden.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error1	Statisches Reset-Signal wurde im Zustand 8001 erkannt. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = TRUE
C002	Reset Error 2	Statisches Reset-Signal wurde im Zustand C004 erkannt. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = TRUE
C003	Reset Error 3	Statisches Reset-Signal wurde im Zustand 8011 erkannt. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = TRUE
C004	Safety Lost	Unsicherer Zustand, Schutztür offen oder entsperrt. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = FALSE
8000	Guard Closed and Locked	Schutztür ist gesperrt. Ready = TRUE S_GuardLocked = TRUE S_UnlockGuard = FALSE Error = FALSE
8001	Init	Funktionsblock wurde aktiviert und initialisiert. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = FALSE
8003	Wait for Reset	Tür ist geschlossen und verriegelt und wartet nun auf ein Operator Reset Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = FALSE

13. Funktionsblöcke

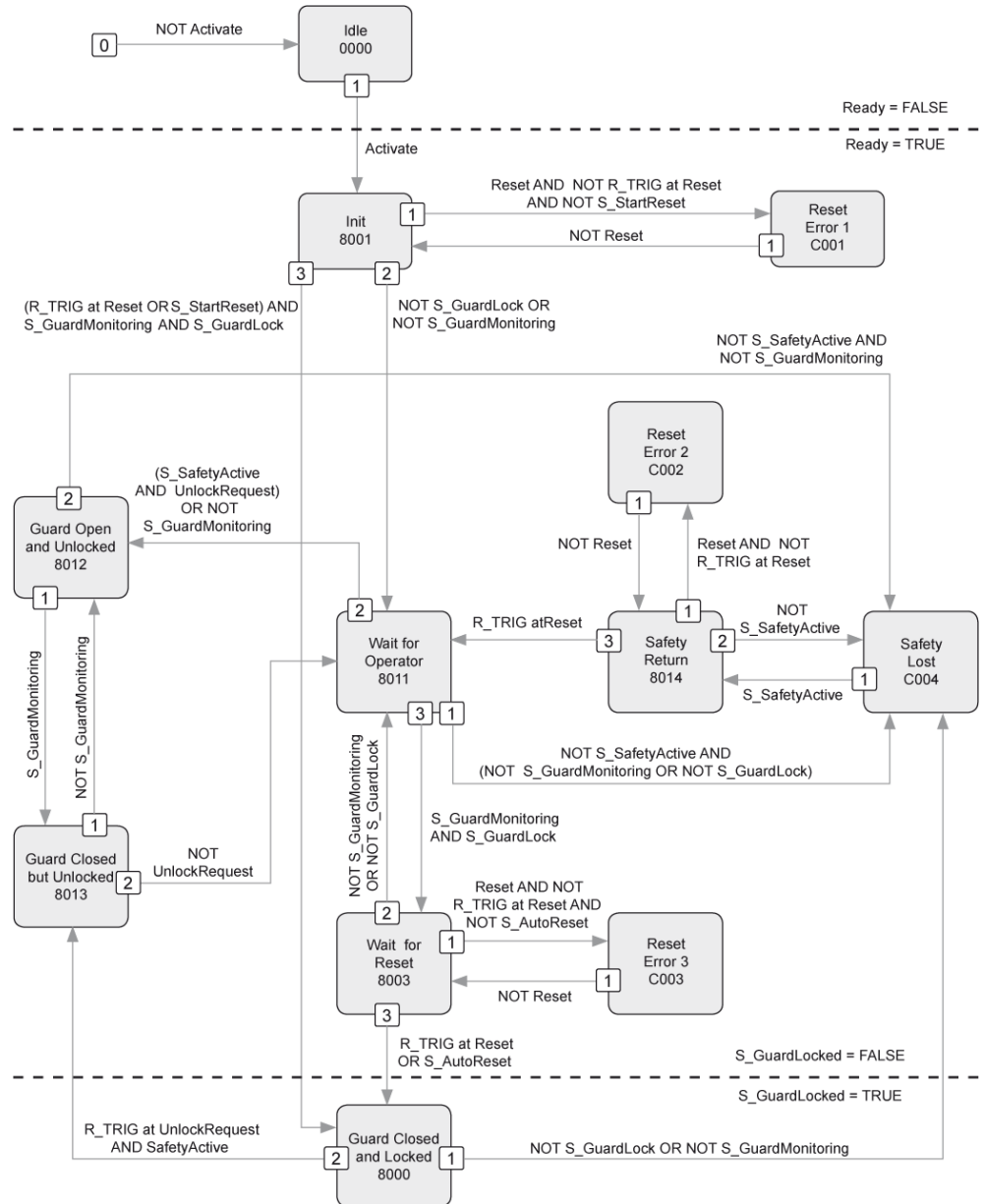
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8011	Wait for Operator	Wartet auf eine Entsperrungsanfrage oder ein Reset vom Operator. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = FALSE
8012	Guard Open and Unlocked	Verriegelung wird aufgehoben und die Schutztür ist geöffnet. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE Error = FALSE
8013	Guard Closed but Unlocked	Verriegelung wird aufgehoben, aber die Schutztür ist geschlossen. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = TRUE Error = FALSE
8014	Safety Return	Rückgabe von S_SafetyActive Signal, wartet nun auf die Bestätigung vom Operator. Ready = TRUE S_GuardLocked = FALSE S_UnlockGuard = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

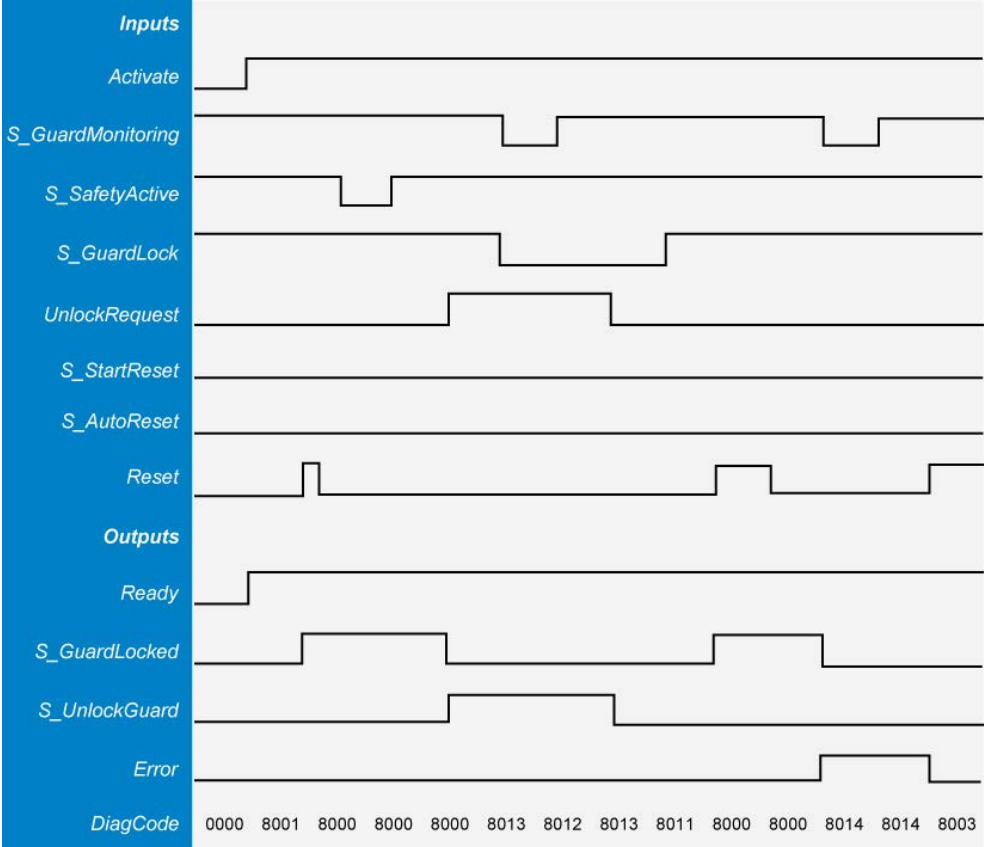
13.5.9.4 Status Diagramm SF_GuardLocking



13. Funktionsblöcke

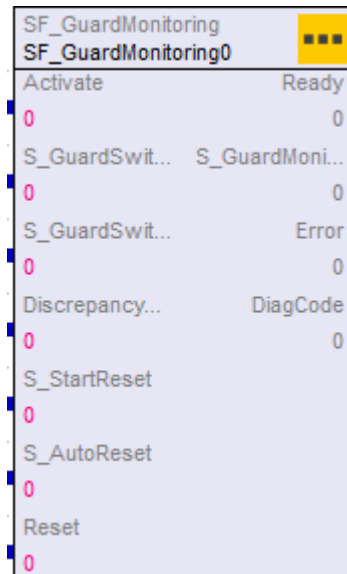
13.5 Komplexe Funktionsblöcke

13.5.9.5 Zeit-Diagramm SF_GuardLocking



13.5.10 SF_GuardMonitoring-Funktion

Dieser Funktionsblock überwacht eine Schutztür. Es gibt zwei unabhängige Eingänge für die beiden Schalter an der Schutztür, in Verbindung mit einer zeitlichen Differenz (MonitoringTime) für die Schließung der Schutzeinrichtung.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_GuardSwitch1	Schalter-Eingang 1 der Schutztür. (SAFEBOOL) FALSE: Schutztür offen. TRUE: Schutztür geschlossen.
S_GuardSwitch2	Schalter-Eingang 2 der Schutztür. (SAFEBOOL) FALSE: Schutztür offen. TRUE: Schutztür geschlossen.
DiscrepancyTime	Konfiguriert die Überwachungszeit, zwischen S_GuardSwitch1 und S_GuardSwitch2 (DINT)
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL).
S_AutoReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der Not-Halt-Schalter wieder freigegeben wird (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output Ergebnisse gültig sind. (BOOL)
S_GuardMonitoring	Der Ausgang zeigt den Status der Schutztür. (SAFEBOOL) FALSE: Schutztür ist nicht aktiv. TRUE: Beide S_GuardSwitches sind TRUE, keine Fehlermeldung. Schutztür ist aktiv.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Der Funktionsblock benötigt zwei Eingänge für die Schalter (nach EN1088) an der Schutztür, eine Diskrepanzzeit und einen Reset-Eingang. Wenn die Schutztür nur einen Schalter hat, können die S_GuardSwitch1- und S_GuardSwitch2-Eingänge gebrückt werden. Die Überwachungszeit ist die maximale Zeit, die benötigt wird, um beide Schalter beim Schließen der Schutztür auszulösen. Die Eingänge Reset, S_StartReset und S_AutoReset bestimmen, wie der Funktionsblock resetiert wird, nachdem die Schutztür geöffnet wurde.

Beim Öffnen der Schutztür sollten sowohl der Eingang S_GuardSwitch1 als auch der Eingang S_GuardSwitch2 auf FALSE schalten. Der S_GuardMonitoring-Ausgang schaltet auf FALSE, sobald einer der Schalter FALSE ist. Beim Schließen der Schutztür sollten sowohl der Eingang S_GuardSwitch1 als auch der S_GuardSwitch2-Eingang auf TRUE schalten.

Dieser FB überwacht die Symmetrie des Schaltverhaltens der beiden Schalter. Der Ausgang S_GuardMonitoring bleibt FALSE, wenn nur einer der Kontakte einen Öffnungs- / Schließvorgang abgeschlossen hat.

Das Verhalten des Ausgangs S_GuardMonitoring hängt von der Zeitdifferenz zwischen den Schaltereingängen ab. Die Diskrepanzzeit wird überwacht, sobald sich die beiden Eingänge S_GuardSwitch1/S_GuardSwitch2 unterscheiden. Ist die Diskrepanzzeit abgelaufen, aber die Eingänge sind noch unterschiedlich, bleibt der Ausgang S_GuardMonitoring FALSE. Schaltet der zweite S_GuardSwitch1/S_GuardSwitch2-Eingang innerhalb des für den DiscrepancyTime definierten Wertes auf TRUE, wird der Ausgang S_GuardMonitoring nach der Quittierung auf TRUE gesetzt.

Die S_StartReset- und S_AutoReset-Eingänge sollten nur aktiviert werden, wenn sichergestellt ist, dass, beim Starten des PES, keine gefährliche Situation auftreten kann.

13.5.10.1 Fehlererkennung

Externe Signale: SAFEBOOL-Eingänge stellen die Fehlererkennung zur Verfügung. Ein mechanischer Aufbau kombiniert einen Öffner und einen Schließer nach EN ISO 13849 (Schutztür mit zwei Schaltern). Die Überwachung der Diskrepanzzeit zur zeitlichen Verzögerung zwischen den beiden mechanischen Schaltern, nach EN ISO 13849 (als "Applikation Fehlererkennung", d.h. durch die Anwendung generiert zu betrachten).

Ein Fehler wird erkannt, wenn die Zeitspanne zwischen dem ersten S_GuardSwitch1/S_GuardSwitch2-Eingang und dem zweiten größer ist, als der Wert des DiscrepancyTime-Eingangs X. Der Fehlerausgang wird auf TRUE gesetzt.

Der Funktionsblock erkennt ein statisches TRUE-Signal am RESET-Eingang.

13.5.10.2 Fehler- und Reset-Verhalten

Der S_GuardMonitoring-Ausgang ist auf FALSE gesetzt. Sind die beiden S_GuardSwitch1- und S_Guardswitch2-Eingänge überbrückt, wird kein Fehler erkannt. Um den Reset-Fehlerzustand zu verlassen, muss der Reset-Eingang auf FALSE gesetzt werden. Um die Diskrepanzzeit-Fehler zu verlassen, müssen die Eingänge S_GuardSwitch1 und 2 auf FALSE gesetzt werden.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error	Statisches Reset-Signal wurde im Zustand 8003 erkannt. Ready = TRUE S_GuardMonitoring = FALSE Error = TRUE
C011	Discrepancytime Error 1	DiscrepancyTime ist im Zustand 8004 abgelaufen. Ready = TRUE S_GuardMonitoring = FALSE Error = TRUE
C012	Discrepancytime Error 2	DiscrepancyTime ist im Zustand 8014 abgelaufen. Ready = TRUE S_GuardMonitoring = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_GuardMonitoring = FALSE Error = FALSE
8000	Normal	Schutztür geschlossen und Safe-Zustand erkannt. Ready = TRUE S_GuardMonitoring = TRUE Error = FALSE
8001	Init	Funktionsblock wurde aktiviert. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE
8002	Open Guard Request	Komplette Schaltfolge erforderlich. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE
8003	Wait for Reset	Warten auf steigende Trigger beim Reset. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE
8012	Guard Opened	Guard vollständig geöffnet. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE

13. Funktionsblöcke

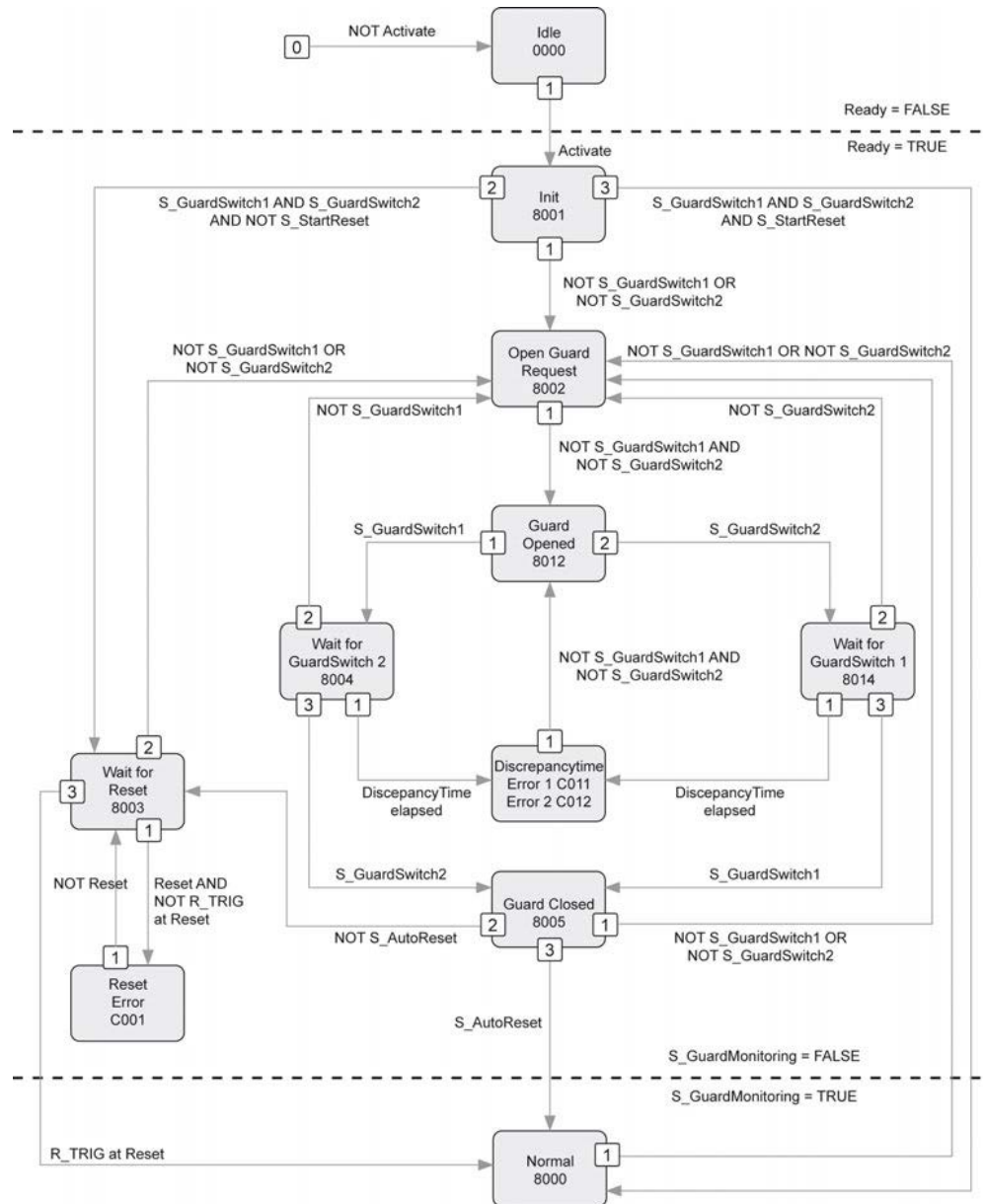
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8004	Wait for GuardSwitch2	S_GuardSwitch1 wurde auf TRUE geschaltet - warten auf S_GuardSwitch2; Diskrepanz-Timer wurde gestartet. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE
8014	Wait for GuardSwitch1	S_GuardSwitch2 wurde auf TRUE geschaltet - warten auf S_GuardSwitch1; Diskrepanz-Timer wurde gestartet. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE
8005	Guard Closed	Schutztür geschlossen. Warten auf Reset, wenn S_AutoReset = FALSE. Ready = TRUE S_GuardMonitoring = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

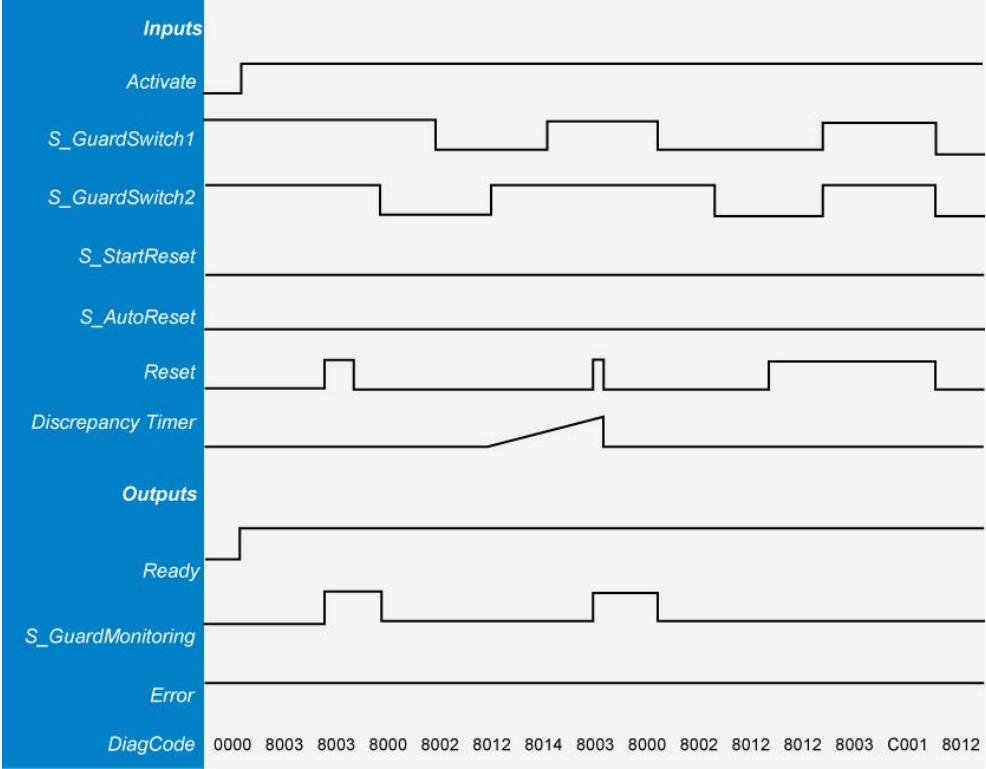
13.5.10.3 Status-Diagramm SF_GuardMonitoring



13. Funktionsblöcke

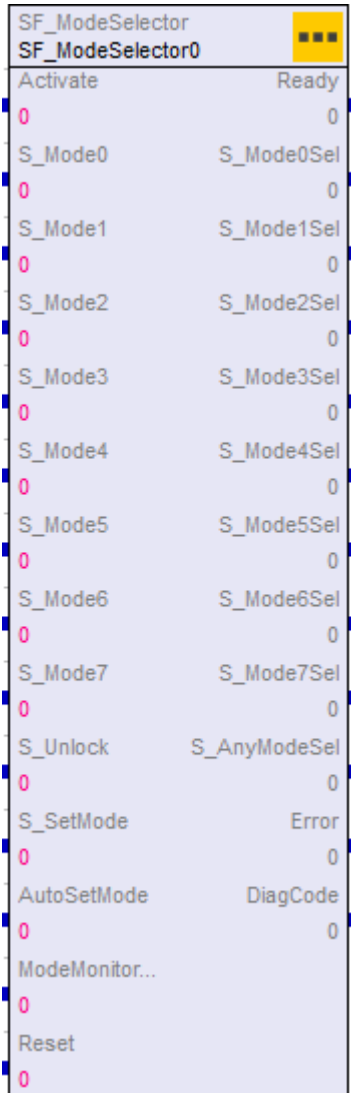
13.5 Komplexe Funktionsblöcke

13.5.10.4 Zeit-Diagramm SF_GuardMonitoring



13.5.11 SF_ModeSelector-Funktion

Dieser Funktionsblock wählt die Betriebsart des Systems z. B. Manuell, Automatik, Halbautomatik, usw.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_Mode0	Gilt für alle: Eingang X vom Betriebsarten-Schalter FALSE: Mode X nicht ausgewählt. TRUE: Mode X vom Operator ausgewählt.
S_Mode1	
S_Mode2	
S_Mode3	
S_Mode4	
S_Mode5	
S_Mode6	
S_Mode7	
S_Unlock	FALSE: Sperrt die Betriebsart. Eine Änderung an den S_ModeX-Eingängen führt zu kei-

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

	ner Änderung an den S_ModeXSel-Ausgängen.
S_SetMode	TRUE: Quittierung zum Setzen einer neuen Betriebsart bei gültigen S_ModeX-Eingängen.
AutoSetMode	TRUE: Automatische Quittierung zum Setzen einer neuen Betriebsart bei gültigen S_ModeX-Eingängen.
ModeMonitorTime	Maximale Überwachungszeit, in der alle S_ModeX-Eingang FALSE sind.
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige dass der Funktionsblock aktiviert ist und die Output Ergebnisse gültig sind (BOOL)
S_Mode0Sel	Gilt für alle. TRUE: Zeigt die aktuelle Betriebsart an.
S_Mode1Sel	
S_Mode2Sel	
S_Mode3Sel	
S_Mode4Sel	
S_Mode5Sel	
S_Mode6Sel	
S_Mode7Sel	
S_AnyModeSel	TRUE: Es ist eine Betriebsart ausgewählt.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Dieser Funktionsblock wählt die Betriebsart des Systems z.B. manuell, automatisch, halbautomatisch, usw. Beim Starten des FB sollte sichergestellt sein, dass sich die Maschine X im sicheren Zustand befindet. Beim Starten der Maschine muss der Übergang in den durch den Modusschalter eingestellten Modus durch einen Funktionsblock-Eingang ausgelöst werden (z.B. START-Taste der Maschine).

Der Standardzustand nach der Aktivierung des FBs ist der ModeChanged Zustand. Dies ist auch der sichere Zustand des FBs, wobei alle S_ModeXSel und S_AnyModeSel auf FALSE gesetzt sind.

Wenn der FB im ModeChanged Zustand ist:

- Der neue S_ModeX-Eingang muss durch eine steigende Flanke am S_SetMode-Eingang quittiert werden, was zu einem neuen S_ModeXSel Ausgang führt. (wenn AutoSetMode = FALSE).
- Der neue S_ModeX-Eingang führt automatisch zu einem neuen S_ModeXSel-Ausgang (wenn AutoSetMode = TRUE).
- Ein Übergang von 8005 auf 8000 ist nur gültig, wenn ein S_ModeX Eingang TRUE ist. Solange alle S_ModeX FALSE sind, bleibt der FB im Zustand 8005, auch wenn eine steigende Flanke an S_SetMode ausgelöst wird.

Der Übergang vom ModeChanged zum ModeSelected-Zustand, der mit S_SetMode durch den Operator quittiert werden muss, wird nicht durch einen Timer überwacht.

Ist der FB im ModeSelected-Zustand, führt das gleichzeitige Auftreten eines neuen S_ModeX-Eingangs (höhere Priorität) und des NOT S_Unlock-Signals (niedrigere Priorität) zum ModeChanged-Zustand.

Die S_ModeX-Eingabeparameter, die nicht zur Auswahl des Modus verwendet werden, sollten mit konstanten FALSE beschalten werden, um die Überprüfung des Programms zu vereinfachen.

Der AutoSetMode-Eingang sollte nur aktiviert werden, wenn sichergestellt ist, dass beim Starten der PES keine gefährliche Situation auftreten kann.

13.5.11.1 Fehlererkennung

Der FB erkennt, ob keiner der Eingangs-Modi ausgewählt ist. Diese ungültige Bedingung wird nach dem Ablauf der ModeMonitorTime erkannt:

- die mit jedem fallenden Trigger eines S_ModeX Schaltermodus-Eingangs neu startet
- die dann nach der Aktivierung des FB im ModeChanged Zustand gebracht wird

Im Gegensatz dazu, erkennt der FB, ob mehr als ein S_ModeX-Eingang zur selben Zeit ausgewählt ist.

Ein statischer Reset-Zustand wird erkannt, wenn der FB entweder im Fehlerstatus C001 oder C002 ist.

13.5.11.2 Fehlerverhalten

Im Falle eines Fehlers werden die S_ModeXSel- und S_AnyModeSel-Ausgänge in den sicheren Zustand = FALSE gesetzt. Der DiagCode-Ausgang zeigt den entsprechenden Fehler-Code an und der Errors-Ausgang wird auf TRUE gesetzt.

Ein Fehler muss mit einer steigenden Flanke am Reset-Eingangs quittiert werden. Der FB wechselt von einem Fehlerzustand in den Zustand ModeChanged.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Error Short-circuit	Der FB hat festgestellt, dass zwei oder mehr S_ModeX auf TRUE sind, z. B. Kabel-Kurzschluss. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE All S_ModeXSel = FALSE
C002	Error Open-circuit	Die FB hat festgestellt, dass alle S_ModeX auf FALSE sind: Die Zeit nach einer fallenden S_ModeX Flanke überschreitet die ModeMonitorTime, z. B. durch einen Kabelbruch. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE All S_ModeXSel = FALSE
C003	Reset Error 1	Statisches Reset-Signal wurde im Zustand C001 erkannt. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE All S_ModeXSel = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C004	Reset Error 2	Statisches Reset-Signal wurde im Zustand C002 erkannt. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE All S_ModeXSel = FALSE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE Error = FALSE S_AnyModeSel = FALSE All S_ModeXSel = FALSE
8005	ModeChanged	Zustand nach der Aktivierung oder wenn S_ModeX geändert wurde (wenn nicht gesperrt) oder nach dem Zurücksetzen eines Fehlerzustands. Ready = TRUE Error = FALSE S_AnyModeSel = FALSE All S_ModeXSel = FALSE
8000	ModeSelected	Gültige Modusauswahl, noch nicht gesperrt. Ready = TRUE Error = FALSE S_AnyModeSel = TRUE S_ModeXSel = Selected X ist TRUE, die anderen sind FALSE.
8004	ModeLocked	Gültige Modusauswahl ist gesperrt. Ready = TRUE Error = FALSE S_AnyModeSel = TRUE S_ModeXSel = Selected X ist TRUE, die anderen sind FALSE.

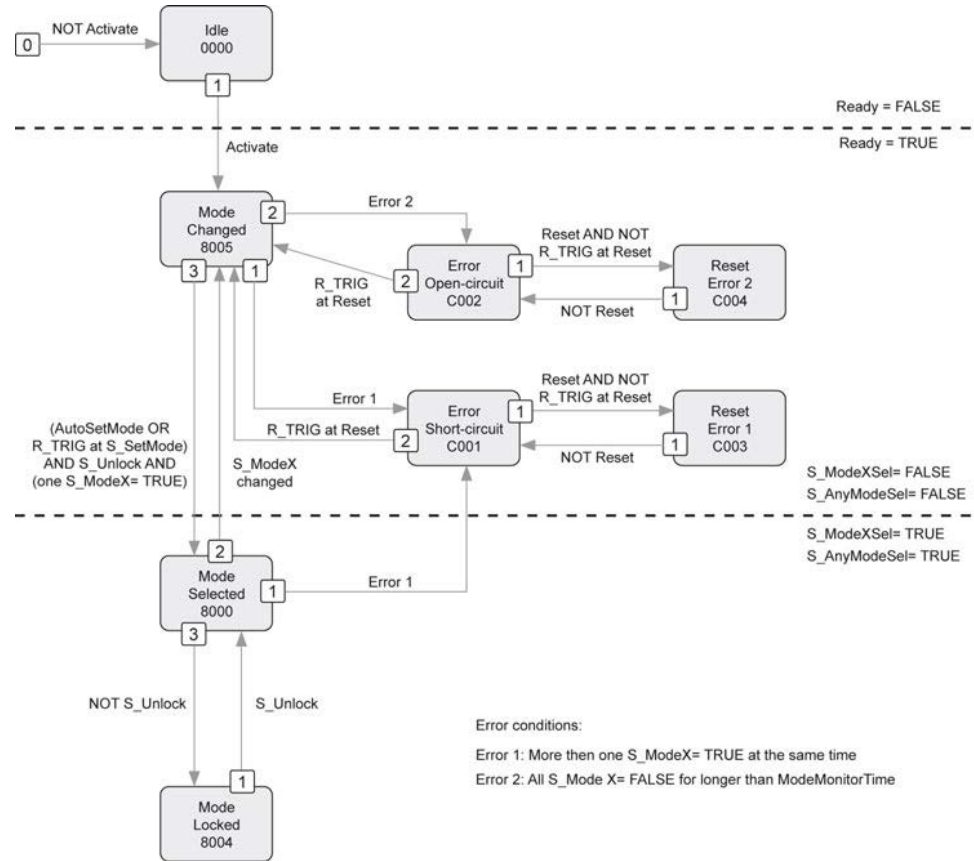
13.5.11.3 Besonderheiten

Die Überwachungszeit in State8005 wird erst gestartet, wenn alle S_ModeX Eingänge SAFE_FALSE sind und beim vorherigen Durchlauf ein Eingang gesetzt war.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

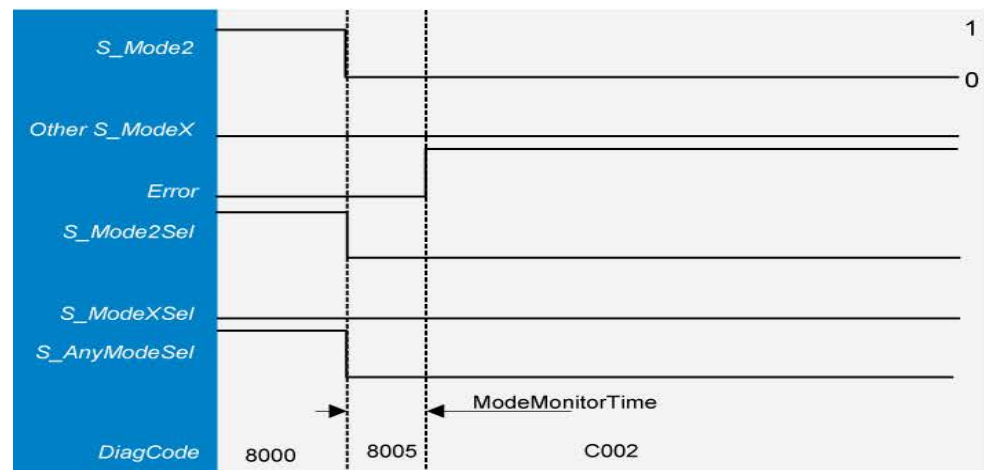
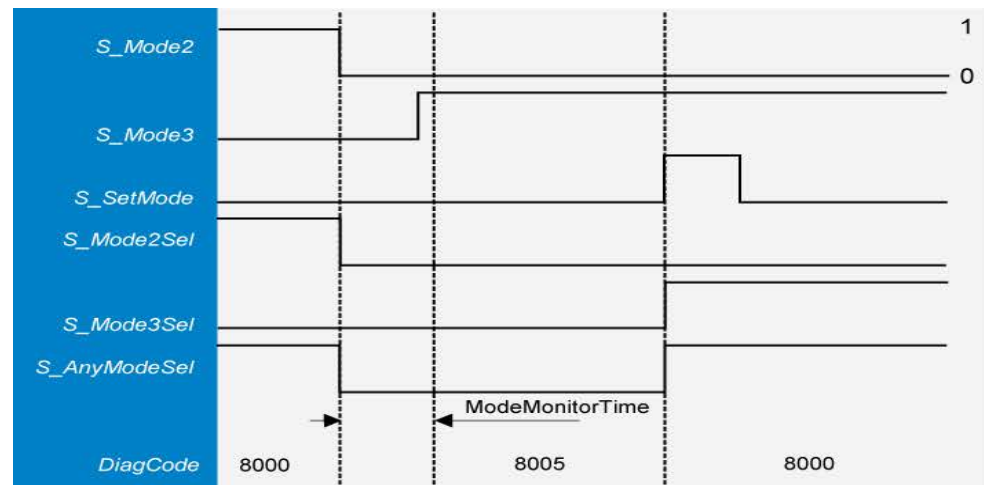
13.5.11.4 Status-Diagramm SF_ModeSelector



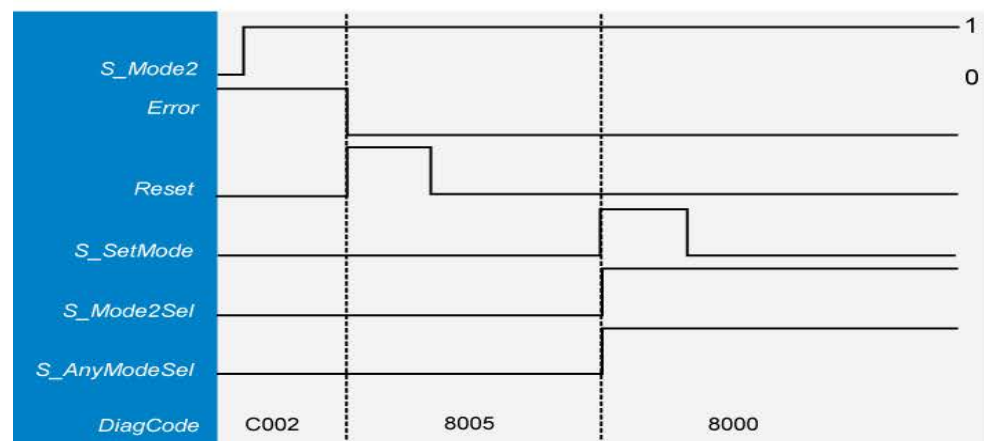
13.5.11.5 Zeit-Diagramm SF_ModeSelector

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke



Timing diagram for SF_ModeSelector, error condition 2 at Mode inputs



Timing diagram for SF_ModeSelector, reset of error condition

13.5.12 SF_MutingPar-Funktion

Muting ist die beabsichtigte Unterdrückung der Sicherheitsfunktion. In diesem Funktionsblock ist paralleles Muting mit vier Muting-Sensoren möglich.



Eingangsvariablen	
Activate	Aktiviert den Funktionsblock (BOOL).
S_AOPD_In	OSSD-Signal (Output Signal Switching Device) von AOPD (Active Opto-electronic Protective Device). FALSE: Schutzfeld unterbrochen. TRUE: Schutzfeld nicht unterbrochen.
MutingSwitch11 MutingSwitch12 MutingSwitch21 MutingSwitch22	Status des Muting-Sensors. Gilt für alle: FALSE: Muting-Sensor nicht aktiviert. TRUE: Werkstück aktiviert den Muting-Sensor. Es wird darauf hingewiesen, dass je nach den Sicherheitsanforderungen anstatt eines BOOL ein SAFEBOOL angeschlossen werden muss. (Safebool to Bool FB)
S_MutingLamp	Gibt den Betriebszustand der Muting-Lampe an. FALSE: Muting-Lampe defekt.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

	TRUE: Kein Muting-Lampen-Fehler
DiscTime11_12	Konstant 0 .. 4 s; Maximale Diskrepanzzeit für MutingSwitch11 und MutingSwitch12.
DiscTime21_22	Konstant 0 .. 4 s; Maximale Diskrepanzzeit für MutingSwitch21 und MutingSwitch22.
MaxMutingTime	Konstant 0 .. 10 min; Maximale Zeit für eine vollständige Muting-Sequenz, der Timer wird bei der Aktivierung des ersten Muting-Sensors gestartet.
MutingEnable	Befehl vom Steuerungssystem, das den Start der Muting-Funktion bei Bedarf des Maschinenzyklus aktiviert. Nach dem Start der Muting-Funktion kann das Signal ausgeschaltet werden. FALSE: Muting nicht aktiviert TRUE: Anfang der Muting-Funktion aktiviert.
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL).
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind (BOOL).
S_AOPD_Out	Der sicherheitsrelevante Ausgang zeigt den Zustand der Schutztür. FALSE: AOPD-Schutzfeld ist unterbrochen und Muting ist nicht aktiv. TRUE: AOPD Schutzfeld ist nicht unterbrochen oder Muting ist aktiv.
S_MutingActive	FALSE: Muting nicht aktiv. TRUE: Muting aktiv.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Muting ist die beabsichtigte Unterdrückung von Sicherheitsfunktionen. Dies ist z.B. notwendig, um Material in eine Gefahrenzone zu transportieren, ohne die Maschine zu stoppen. Muting wird durch Muting-Sensoren überwacht. Die Verwendung von zwei oder vier Muting-Sensoren und die korrekte Integration in den Fertigungsablauf müssen sicherstellen, dass keine Personen den Gefahrenbereich betreten, während der Lichtvorhang ausgeschaltet ist. Muting-Sensoren können Näherungsschalter, Lichtschranken, Endschalter, usw. sein, die nicht ausfallsicher sein müssen.

Der aktive Muting-Modus muss durch Anzeigelampen angezeigt werden.

Es gibt sequentielle und parallele Muting-Verfahren. In diesem FBX wird Parallel-Muting mit vier Muting-Sensoren angewendet. Der FB kann für Muting in beide Richtungen verwendet werden - vorwärts und rückwärts. Das Muting sollte durch die Prozesssteuerung mit dem MutingEnable Signal aktiviert werden, um Manipulationen zu vermeiden.

Die FB-Eingänge enthalten die Signale der vier Muting-Sensoren (MutingSwitch11 ... MutingSwitch22), das OSSD-Signal aus der "aktiven opto-elektronischen Schutzrichtung", S_AOPD_In, sowie drei parametrierbare Zeiten (DiscTime11_12, DiscTime21_22 und MaxMutingTime).

Der S_StartReset-Eingang sollte nur aktiviert werden, wenn sichergestellt ist, dass keine gefährliche Situation beim Starten der PES auftreten kann.

13.5.12.1 Fehlererkennung

Der FB erkennt folgende Fehlerbedingungen:

- DiscTime11_12 und DiscTime21_22 wurden auf Werte kleiner als T#0 s oder größer als T#4 s gestellt.
- MaxMutingTime wurde auf einen Wert gestellt, der kleiner als T#0s oder größer als T#10 min ist.
- Die Diskrepanzzeit von den MutingSwitch11/MutingSwitch12 oder MutingSwitch21/MutingSwitch22 Sensorpaaren wurde überschritten.
- Die Muting-Funktion (S_MutingActive = TRUE) überschreitet die maximale Muting-Zeit MaxMutingTime.
- Muting-Sensoren MutingSwitch11, MutingSwitch12, MutingSwitch21 und MutingSwitch22 wurden in der falschen Reihenfolge aktiviert.
- Die Muting-Sequenz beginnt, ohne von MutingEnable aktiviert zu werden.
- Eine fehlerhafte Muting-Lampe wird durch S_MutingLamp = FALSE angezeigt.
- Ein statisches Reset-Signal liegt im Zustand 8001 oder 8003 an.

13.5.12.2 Fehlerverhalten

Im Falle eines Fehlers werden die S_AOPD_Out- und S_MutingActive-Ausgänge auf FALSE gesetzt. Der DiagCode-Ausgang zeigt den entsprechenden Fehler-Code an und der Error Ausgang wird auf TRUE gesetzt. Ein Neustart wird verhindert, bis die Fehlerbedingungen behoben wurden und der sichere Zustand vom Operator mit Reset quittiert wurde.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal wurde nach FB-Aktivierung im Zustand 8001 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C002	Reset Error 2	Statische Reset-Signal wurde im Zustand 8003 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C003	Error Muting Lamp	Fehler in der Muting-Lampe gefunden. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
CYx4	Error Muting sequence	Falsche Muting-Sequenz im Status 8000, 8011, 8311, 8012, 8021, 8014, 8314, 8122, 8422, 8121, 8112, 8114 oder 8414. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE Y = Status in der Sequenz (6 Zustände für Vorwärts- und 6 Zustände für Rückwärts). C0x4 = Fehler im Zustand 8000 aufgetreten C1x4 = Fehler im Zustand Vorwärts 8011 aufgetreten C2x4 = Fehler im Zustand Vorwärts 8311 aufgetreten C3x4 = Fehler im Zustand Vorwärts 8012 aufgetreten C4x4 = Fehler im Zustand Vorwärts 8014 aufgetreten C5x4 = Fehler im Zustand Vorwärts 8314 aufgetreten C6x4 = Fehler im Zustand Vorwärts 8021 aufgetreten C7x4 = Fehler im Zustand Rückwärts 8012 aufgetreten C8x4 = Fehler im Zustand Rückwärts 8422 aufgetreten C9x4 = Fehler im Zustand Rückwärts 8121 aufgetreten CAx4 = Fehler im Zustand Rückwärts 8114 aufgetreten CBx4 = Fehler im Zustand Rückwärts 8414 aufgetreten CCx4 = Fehler im Zustand Rückwärts 8112 aufgetreten CFx4 = Muting Enable fehlt x = Status der Sensoren bei Auftreten des Fehlers. (4 Bits: LSB = MS_11; MS_12; MS_21; MSB = MS_22).
C005	Parameter Error	Der Wert von DiscTime11_12, DiscTime21_22 oder MaxMutingTime ist außerhalb des zulässigen Bereichs. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C006	Error Timer MaxMuting	Timing-Fehler: Aktive Muting-Zeit (wenn S_MutingActive = TRUE) überschreitet MaxMutingTime. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C007	Error Timer MS11_12	Timing-Fehler: Diskrepanzzeit von MutingSwitch11 und MutingSwitch12 > DiscTime11_12. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C008	Error Timer MS21_22	Timing-Fehler: Diskrepanzzeit von MutingSwitch21 und MutingSwitch22 > DiscTime21_22. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8000	AOPD Free	Muting nicht aktiv und keine Sicherheitsanforderung von AOPD. Die Timer werden gestoppt. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8001	Init	Funktionsblock wurde aktiviert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8002	Safety Demand AOPD	Sicherheitsanforderung von AOPD wurde erkannt, Muting ist nicht aktiv. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8003	Wait for Reset	Sicherheitsanforderungen oder Fehler wurden erkannt und werden jetzt gelöscht. Quittierung vom Operator wird benötigt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8005	Safe	Safety-Funktion ist aktiviert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8011	Muting Forward Start 1	Muting-Vorwärts Sequenz ist nach der steigenden Flanke von MutingSwitch 11 in der Startphase. Überwachung von DiscTime11_12 ist aktiviert. Überwachung von MaxMutingTime ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8311	Muting Forward Start 2	Muting-Vorwärts Sequenz ist nach der steigenden Flanke von MutingSwitch 12 in der Startphase. Überwachung von DiscTime11_12 ist aktiviert. Überwachung von MaxMutingTime ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8012	Muting Forward Active 1	Muting-Vorwärts Sequenz ist aktiv: - nach einer steigenden Flanke des zweiten MutingSwitch 12 oder 11. - wenn beide MutingSwitch 11 und 12 im gleichen Zyklus betätigt wurden. Überwachung der DiscTime11_12 wurde gestoppt. Überwachung der MaxMuting-Time wird aktiviert, wenn der Übergang direkt vom Zustand 8000 kam. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8014	Muting Forward Step 1	Muting-Vorwärts Sequenz ist aktiv. MutingSwitch21 ist der erste betätigte Ausgangsschalter. Überwachung der DiscTime21_22 wird gestartet. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8314	Muting Forward Step 2	Muting-Vorwärts-Sequenz ist aktiv. MutingSwitch22 ist der erste betätigte Ausgangsschalter. Überwachung der DiscTime21_22 wird gestartet. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8021	Muting Forward Active 2	Muting-Vorwärts-Sequenz ist noch aktiv. MutingSwitch21 und 22 sind betätigt, die Überwachung der DiscTime21_22 wird gestoppt. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8122	Muting Backward Start 1	Muting-Rückwärts-Sequenz ist nach der steigenden Flanke von MutingSwitch21 in der Startphase. Überwachung der DiscTime21_22 ist aktiviert. Überwachung von MaxMutingTime ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8422	Muting Backward Start 2	Muting-Rückwärts-Sequenz ist nach der steigenden Flanke von MutingSwitch22 in der Startphase. Überwachung der DiscTime21_22 ist aktiviert. Überwachung von MaxMutingTime ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE

13. Funktionsblöcke

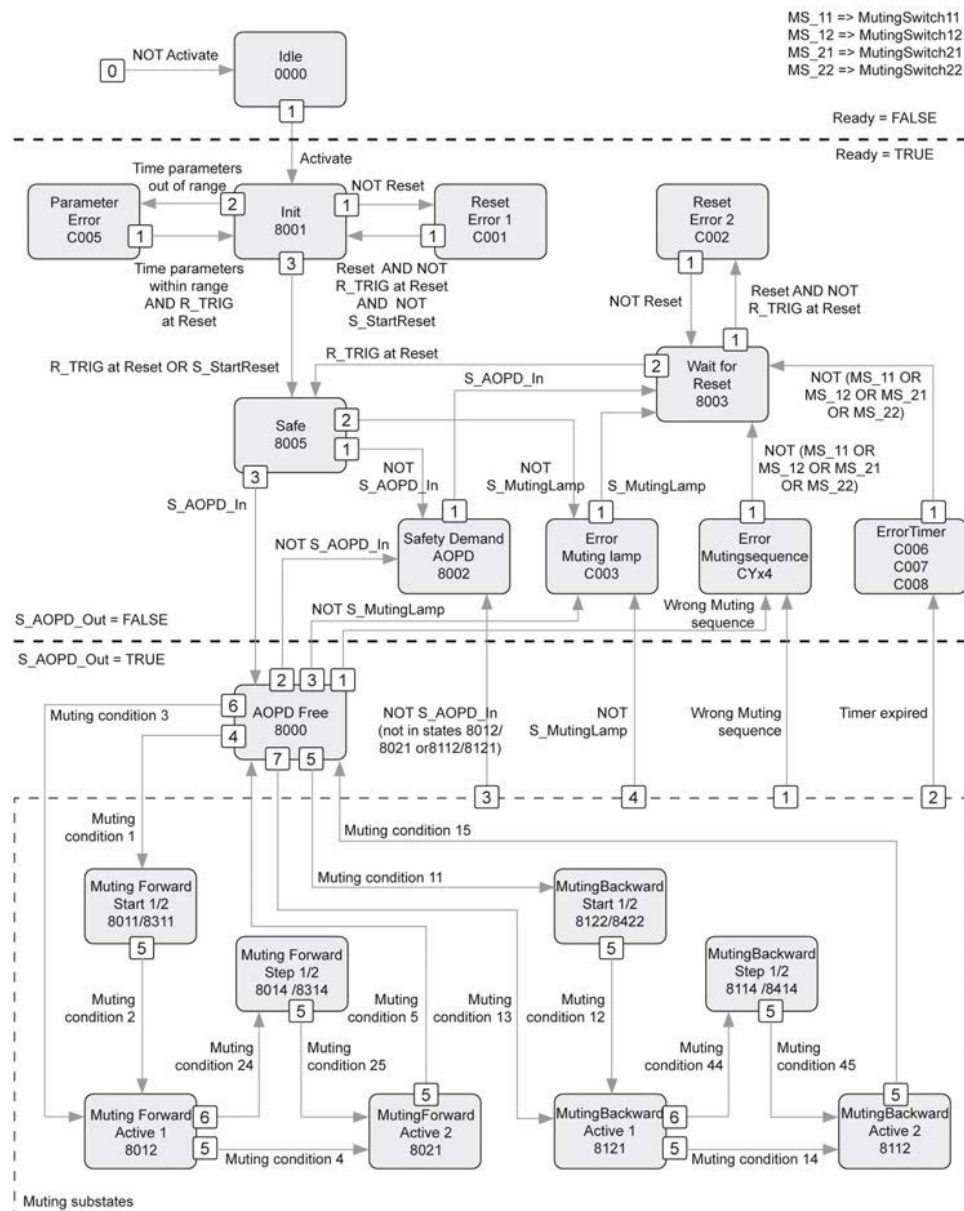
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8121	Muting Backward Active 1	Muting-Rückwärts Sequenz ist aktiv: - nach einer steigende Flanke des zweiten MutingSwitch 21 oder 22. - wenn MutingSwitch 21 und 22 im gleichen Zyklus betätigt wurden. Überwachung der DiscTime21_22 wird gestoppt. Überwachung der MaxMuting-Time wird aktiviert, wenn der Übergang direkt vom Zustand 8000 kam. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8114	Muting Backward Step 1	Muting-Rückwärts Sequenz ist aktiv. MutingSwitch11 ist der erste betätigte Ausgangsschalter. Überwachung der DiscTime11_12 wird gestartet. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8414	Muting Backward Step 2	Muting-Rückwärts Sequenz ist aktiv. MutingSwitch12 ist der erste betätigte Ausgangsschalter. Überwachung der DiscTime11_12 wird gestartet. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8112	Muting Backward Active 2	Muting-Rückwärts Sequenz ist noch aktiv. Die Exit-Schalter MutingSwitch11 und 12 sind betätigt, die Überwachung der DiscTime11_12 wird gestoppt. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE

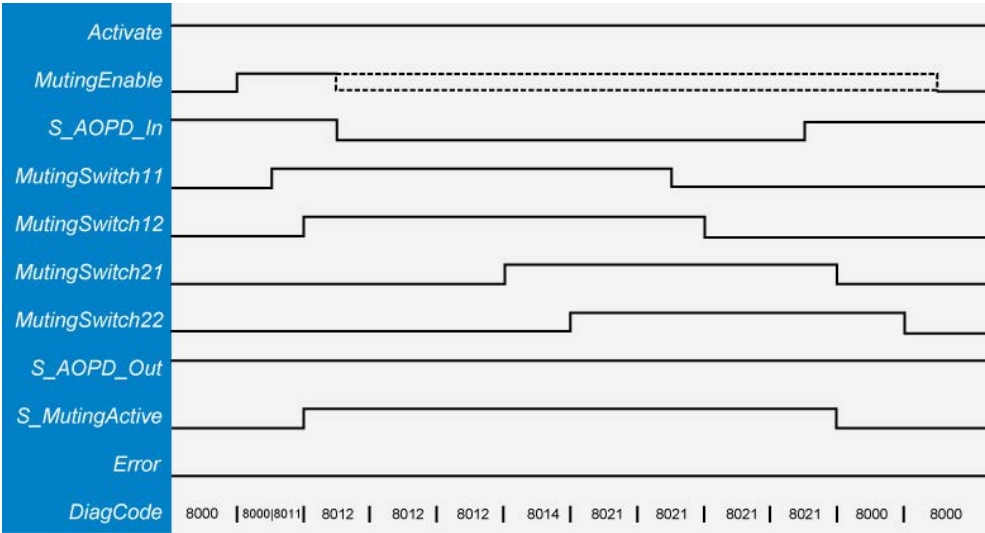
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.12.3 Status Diagramm SF_MutingPar



13.5.12.4 Zeit-Diagramm SF_MutingPar

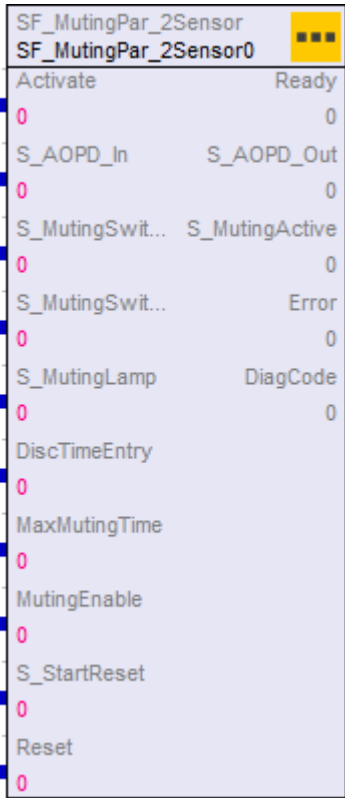


13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.13 SF_MutingPar_2Sensor-Funktion

Muting ist die beabsichtigte Unterdrückung der Sicherheitsfunktionen. In diesem Funktionsblock ist paralleles Muting mit zwei Muting-Sensoren möglich.



Eingangsvariablen	
Activate	Aktiviert den Funktionsblock (BOOL)
S_AOPD_In	OSSD-Signal (Output Signal Switching Device) von AOPD (Active Opto-electronic Protective Device). FALSE: Schutzfeld unterbrochen. TRUE: Schutzfeld nicht unterbrochen.
S_MutingSwitch11 S_MutingSwitch12	Status des Muting-Sensors. Gilt für alle: FALSE: Muting-Sensor nicht aktiviert. TRUE: Werkstück aktiviert den Muting-Sensor.
S_MutingLamp	Gibt den Betriebszustand der Muting-Lampe an. FALSE: Muting-Lampe defekt. TRUE: Kein Muting-Lampen-Fehler
DiscTimeEntry	Konstant 0-4 s; Max. Diskrepanzzeit zur Aktivierung von S_MutingSwitch11 und S_MutingSwitch12.
MaxMutingTime	Konstant 0-10 min; Maximale Zeit für eine vollständige Muting-Sequenz, der Timer wird bei der Aktivierung des ersten Muting-Sensors gestartet.
MutingEnable	Befehl vom Steuerungssystem, das bei Bedarf den Start der Muting-Funktion des Maschinenzyklus aktiviert. Nach dem Start der Muting-Funktion kann das Signal ausgeschaltet werden. FALSE: Muting nicht aktiviert, TRUE: Anfang der Muting-Funktion aktiviert.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wurde (SAFEBOOL)
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Ausgänge gültig sind. (BOOL)
S_AOPD_Out	Sicherheitsrelevanter Ausgang, zeigt den Status der Muting Schutztür. FALSE: AOPD Schutzfeld unterbrochen und Muting nicht aktiv. TRUE: AOPD Schutzfeld nicht unterbrochen oder muting ist aktiv.
S_MutingActive	Zeigt den Status des Muting-Prozesses. FALSE: Muting nicht aktiv. TRUE: Muting aktiv.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Muting ist die beabsichtigte Unterdrückung von Sicherheitsfunktionen. Dies ist z.B. notwendig um Material in eine Gefahrenzone zu transportieren, ohne die Maschine zu stoppen. Muting wird durch Muting-Sensoren überwacht. Die Verwendung von zwei oder vier Muting-Sensoren und die korrekte Integration in den Fertigungsablauf müssen sicherstellen, dass keine Personen den Gefahrenbereich betreten, während der Lichtvorhang ausgeschaltet ist.

Muting-Sensoren können Drucktasten, Näherungsschalter, Lichtschranken, Endschalter, usw. sein, die nicht ausfallsicher sein müssen.

Der aktive Muting-Modus muss durch Anzeigelampen angezeigt werden.

Es gibt sequentielle und parallele Muting-Verfahren. In diesem FB wird Parallel-Muting mit zwei Muting-Sensoren angewendet.

Der FB kann für Muting in beide Richtungen verwendet werden - vorwärts und rückwärts. Allerdings kann die tatsächliche Richtung nicht identifiziert werden.

Das Muting sollte durch die Prozesssteuerung mit dem MutingEnable Signal aktiviert werden, um Manipulationen zu vermeiden.

Die FB-Eingänge enthalten die Signale der beiden Muting-Sensoren (S_MutingSwitch11 und S_MutingSwitch12), das OSSD-Signal aus der "aktiven optoelektronischen Schutzeinrichtung", S_AOPD_In, sowie zwei parametrierbare Zeiten (Disc-TimeEntry und MaxMutingTime).

Der S_StartReset-Eingang sollte nur aktiviert werden, wenn sichergestellt ist, dass beim Starten der PES keine gefährliche Situation auftreten kann.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.13.1 Fehlererkennung

Der FB erkennt folgende Fehlerbedingungen:

- DiscTimeEntry wurde auf einen Wert kleiner als T#0s oder größer als T#4s gesetzt.
- MaxMutingTime wurde auf einen Wert gestellt, der kleiner als T#0 s oder größer als T#10 min ist.
- Die Diskrepanzzeit zum S_MutingSwitch11/S_MutingSwitch12 Fühlerpaar wurde überschritten.
- Die Muting-Funktion (S_MutingActive = TRUE) überschreitet die maximale Muting-Zeit MaxMutingTime.
- Muting-Sensoren S_MutingSwitch11, S_MutingSwitch12 wurden in der falschen Reihenfolge aktiviert.
- Muting-Sequenz beginnt, ohne von MutingEnable aktiviert zu werden.
- Statische Muting-Sensor Signale.
- Eine fehlerhafte Muting-Lampe wird durch S_MutingLamp = FALSE angezeigt.
- Ein statisches Reset-Signal liegt im Zustand 8001 oder 8003 an.

13.5.13.2 Fehlerverhalten

Im Falle eines Fehlers sind die S_AOPD_Out- und S_MutingActive-Ausgänge auf FALSE gesetzt. Der DiagCode-Ausgang zeigt den entsprechenden Fehler-Code an und der Error-Ausgang wird auf TRUE gesetzt. Ein Neustart wird verhindert, bis die Fehler-Bedingungen behoben sind und der sichere Zustand vom Operator mit Reset quittiert wurde.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal wurde nach Aktivierung im Zustand 8001 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C002	Reset Error 2	Statisches Reset-Signal wurde im Zustand 8003 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C003	Error Muting Lamp	Fehler in der Muting-Lampe gefunden. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
CYx4	Error Muting sequence	CYx4 Muting-Sequenz-Fehler wurde in Muting-Sequenz Zustand 8000, 8011, 8311 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE Y = Status in der Reihenfolge C0x4 = Fehler im Zustand 8000 aufgetreten C1x4 = Fehler im Zustand 8011 aufgetreten C2x4 = Fehler im Zustand 8311 aufgetreten CFx4 = Muting Enable fehlt x = Status der Sensoren beim Auftreten des Fehlers aufgetreten (4 Bits: LSB = MS_11, neben LSB = MS_12).
C005	Parameter Error	Der Wert in DiscTimeEntry oder MaxMutingTime ist außerhalb des zulässigen Bereichs. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C006	Error timer MaxMuting	Timing-Fehler: Aktive Muting-Zeit (wenn S_MutingActive = TRUE) MaxMutingTime überschreitet. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C007	Error timer Entry	Timing-Fehler: Diskrepanzzeit abgelaufen bei Wechsel S_MutingSwitch11 und S_MutingSwitch12 von FALSE auf TRUE. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8000	AOPD Free	Muting nicht aktiv und keine Sicherheitsanforderung von AOPD. Die Timer werden gestoppt. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8001	Init	Funktionsblock wurde aktiviert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE

13. Funktionsblöcke

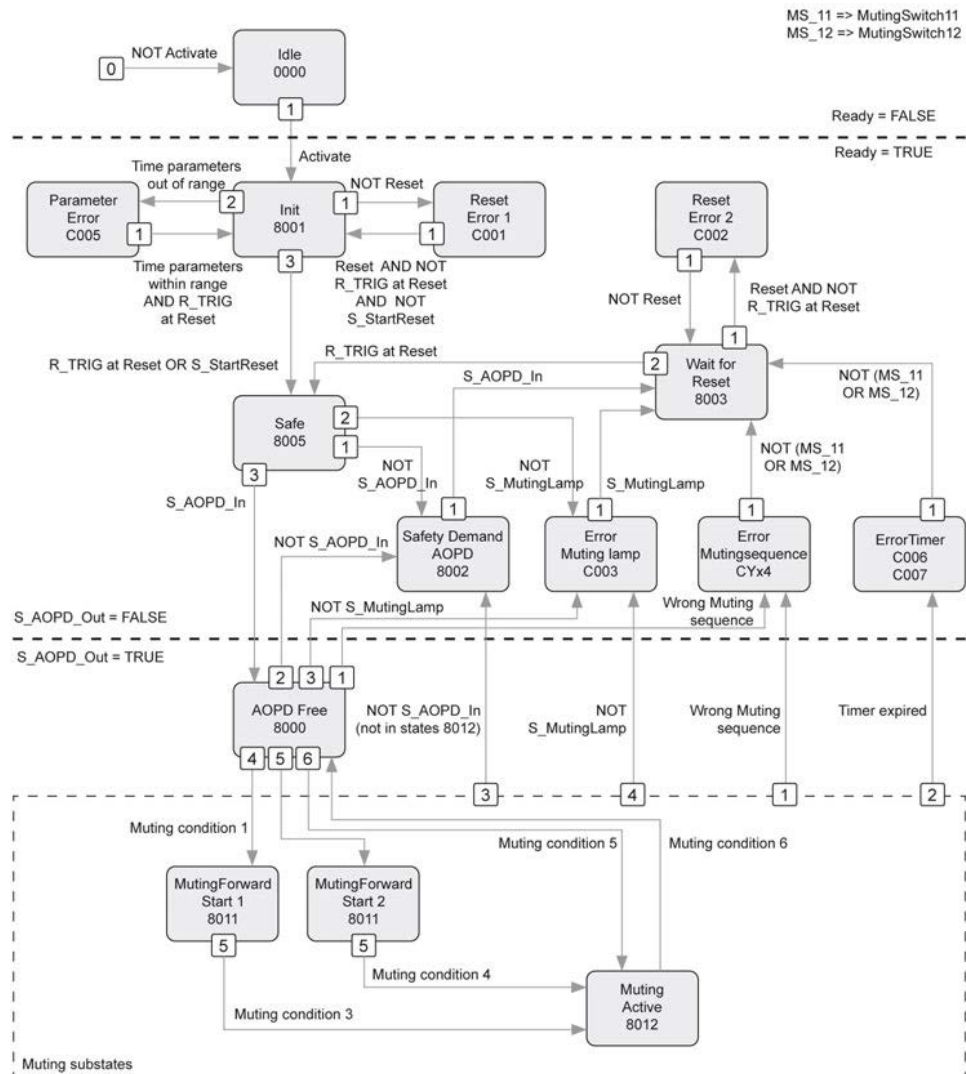
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8002	Safety Demand AOPD	Sicherheitsanforderung von AOPD wurde erkannt, Muting nicht aktiv. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8003	Wait for Reset	Sicherheitsanforderung oder Fehler wurde erkannt und ist jetzt quittiert. Quittierung vom Operator wird beim Reset benötigt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8005	Safe	Safety-Funktion aktiviert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8011	Muting Start 1	Muting-Sequenz ist nach steigender Flanke von S_MutingSwitch11 in der Startphase. Überwachung von DiscTimeEntry ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8311	Muting Start 2	Muting-Sequenz ist nach der steigenden Flanke von S_MutingSwitch12 in der Startphase. Überwachung von DiscTimeEntry ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8012	Muting Active	Muting-Sequenz ist aktiv. - nachdem eine steigende Flanke des zweiten S_MutingSwitch 12 oder 11 erkannt wurde. - wenn S_MutingSwitch 11 und 12 im gleichen Zyklus betätigt wurden. Überwachung der DiscTimeEntry wird gestoppt. Überwachung von MaxMutingTime ist aktiviert. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE

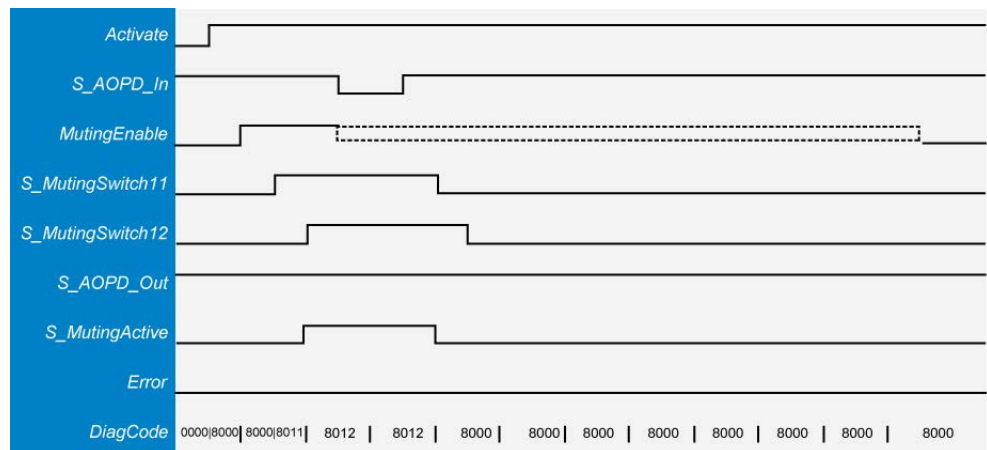
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.13.3 Status-Diagramm SF_MutingPar_2Sensor



13.5.13.4 Zeit-Diagramm SF_MutingPar_2Sensor



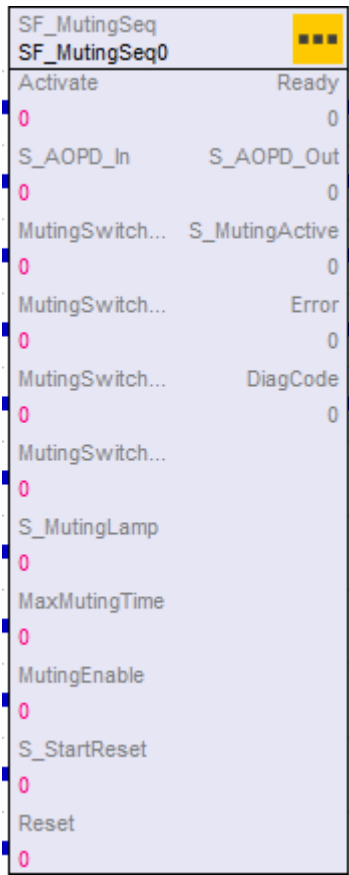
Timing diagram for SF_MutingPar_2Sensor (S_StartReset = TRUE, Reset = FALSE, S_MutingLamp = TRUE)

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.14 SF_MutingSeq-Funktion

Muting ist die beabsichtigte Unterdrückung der Sicherheitsfunktion. In diesem Funktionsblock ist sequentielles Muting mit vier Muting-Sensoren möglich (z.B. Lichtschranken).



Eingangsvariablen	
Activate	Aktiviert den Funktionsblock (BOOL)
S_AOPD_In	OSSD-Signal (Output Signal Switching Device) von AOPD (Active Opto-electronic Protective Device). FALSE: Schutzfeld unterbrochen. TRUE: Schutzfeld nicht unterbrochen.
MutingSwitch11 MutingSwitch12 MutingSwitch21 MutingSwitch22	Status der Muting-Sensoren. Gilt für alle: FALSE: Muting-Sensor nicht aktiviert. TRUE: Werkstück aktiviert den Muting-Sensor. Er wird darauf hingewiesen, dass je nach den Sicherheitsanforderungen, anstatt eines BOOL ein SAFEBOOL angeschlossen werden muss. (Safebool to Bool FB)
S_MutingLamp	Gibt den Betriebszustand der Muting-Lampe an. FALSE: Muting-Lampe defekt. TRUE: Kein Muting-Lampen-Fehler
MaxMutingTime	Konstant 0 .. 10 min; Maximale Zeit für eine vollständige Muting-Sequenz, der Timer wird bei der Aktivierung des ersten Muting-Sensor gestartet.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

MutingEnable	Befehl vom Steuerungssystem, das den Start der Muting-Funktion bei Bedarf des Maschinenzyklus aktiviert. Nach dem Start der Muting-Funktion kann das Signal ausgeschaltet werden. FALSE: Muting nicht aktiviert, TRUE: Anfang der Muting-Funktion aktiviert.
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL).
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind (BOOL).
S_AOPD_Out	Sicherheitsrelevanter Ausgang zeigt den Status der Muting-Schutztür. FALSE: AOPD-Schutzfeld unterbrochen und Muting ist nicht aktiv. TRUE: AOPD-Schutzfeld nicht unterbrochen oder Muting ist aktiv.
S_MutingActive	Zeigt den Status des Muting-Prozesses. FALSE: Muting nicht aktiv, TRUE: Muting aktiv
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Muting ist die beabsichtigte Unterdrückung von Sicherheitsfunktionen. Dies ist z.B. notwendig um Material in eine Gefahrenzone zu transportieren, ohne die Maschine zu stoppen. Muting wird durch Muting-Sensoren überwacht. Die Verwendung von zwei oder vier Muting-Sensoren und die korrekte Integration in den Fertigungsablauf müssen sicherstellen, dass keine Personen den Gefahrenbereich betreten, während der Lichtvorhang ausgeschaltet ist. Muting-Sensoren können Näherungsschalter, Lichtschranken, Endschalter, usw. sein, die nicht ausfallsicher sein müssen.

Der aktive Muting-Modus muss durch Anzeigelampen angezeigt werden.

Es gibt sequentielle und parallele Muting-Verfahren. In diesem FB wird sequentielles Muting mit vier Muting-Sensoren angewendet. Der FB kann in beide Richtungen verwendet werden - vorwärts und rückwärts. Das Muting sollte durch die Prozesssteuerung mit dem MutingEnable-Signal aktiviert werden, um Manipulationen zu vermeiden. Wenn das MutingEnable-Signal nicht verfügbar ist, muss dieser Eingang auf TRUE gesetzt werden.

Die FB-Eingänge beinhalten die Signale der vier Muting-Sensoren (MutingSwitch11 ... MutingSwitch22), sowie das OSSD-Signal aus der "aktiven opto-elektronischen Schutzeinrichtung" S_AOPD_In.

Der S_StartReset Eingang sollte nur aktiviert werden, wenn sichergestellt ist, dass beim Starten der PES keine gefährliche Situation auftreten kann.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.14.1 Fehlererkennung

Der FB erkennt folgende Fehlerbedingungen:

- Muting-Sensoren MutingSwitch11, MutingSwitch12, MutingSwitch21 und MutingSwitch22 werden in der falschen Reihenfolge aktiviert.
- Muting-Sequenz beginnt, ohne von MutingEnable aktiviert zu werden.
- Eine fehlerhafte Muting-Lampe wird durch S_MutingLamp = FALSE angezeigt.
- Eine statische Reset Bedingung.
- MaxMutingTime wurde auf einen Wert gestellt, der kleiner als T#0 s oder größer als T#10 min ist.
- Die Muting-Funktion (S_MutingActive = TRUE) überschreitet die maximal Muting-Zeit MaxMutingTime.

13.5.14.2 Fehlerverhalten

Im Falle eines Fehlers werden die S_AOPD_Out- und S_MutingActive-Ausgänge auf FALSE gesetzt. Der DiagCode-Ausgang zeigt den entsprechenden Fehler-Code an und der Error Ausgang wird auf TRUE gesetzt. Ein Neustart wird verhindert, bis die Fehler-Bedingungen behoben sind und vom Operator mit Reset quittiert wird.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal wurde nach der FB-Aktivierung erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C002	Reset Error 2	Statisches Reset-Signal wurde im Zustand 8003 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C003	Error Muting lamp	Fehler in Muting-Lampe gefunden. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
CYx4	Wrong mutig sequence	Falsche Muting-Sequenz in den States, 8000, 8011, 8012, 8112 oder 8122 erkannt. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE Y = Status in der Sequenz (2 Zustände für die Vorwärts-Richtung und 2 für die Rückwärts-Richtung). C0x4 = Fehler im Zustand 8000 aufgetreten C1x4 = Fehler im Zustand Vorward 8011 aufgetreten C2x4 = Fehler im Zustand Vorward 8012 aufgetreten C3x4 = Fehler im Zustand Rückward 8122 aufgetreten C4x4 = Fehler im Zustand Rückward 8112 aufgetreten CFx4 = Muting Enable fehlt x = Status der Sensoren beim Auftreten des Fehlers aufgetreten (4 Bits: LSB = MS_11; MS_12; MS_21; MSB = MS_22).
C005	Parameter Error	MaxMutingTime liegt außerhalb des zulässigen Bereichs. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
C006	Error Timer	Timing-Fehler: Aktive Muting-Zeit (wenn S_MutingActive = TRUE) überschreitet MaxMutingTime. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8000	AOPD Free	Muting nicht aktiv und keine Sicherheitsanforderung von AOPD. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8001	Init	Funktionsblock wurde aktiviert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8002	Safety Demand AOPD	Sicherheitsanforderungen von AOPD wurden erkannt, Muting ist nicht aktiv. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE

13. Funktionsblöcke

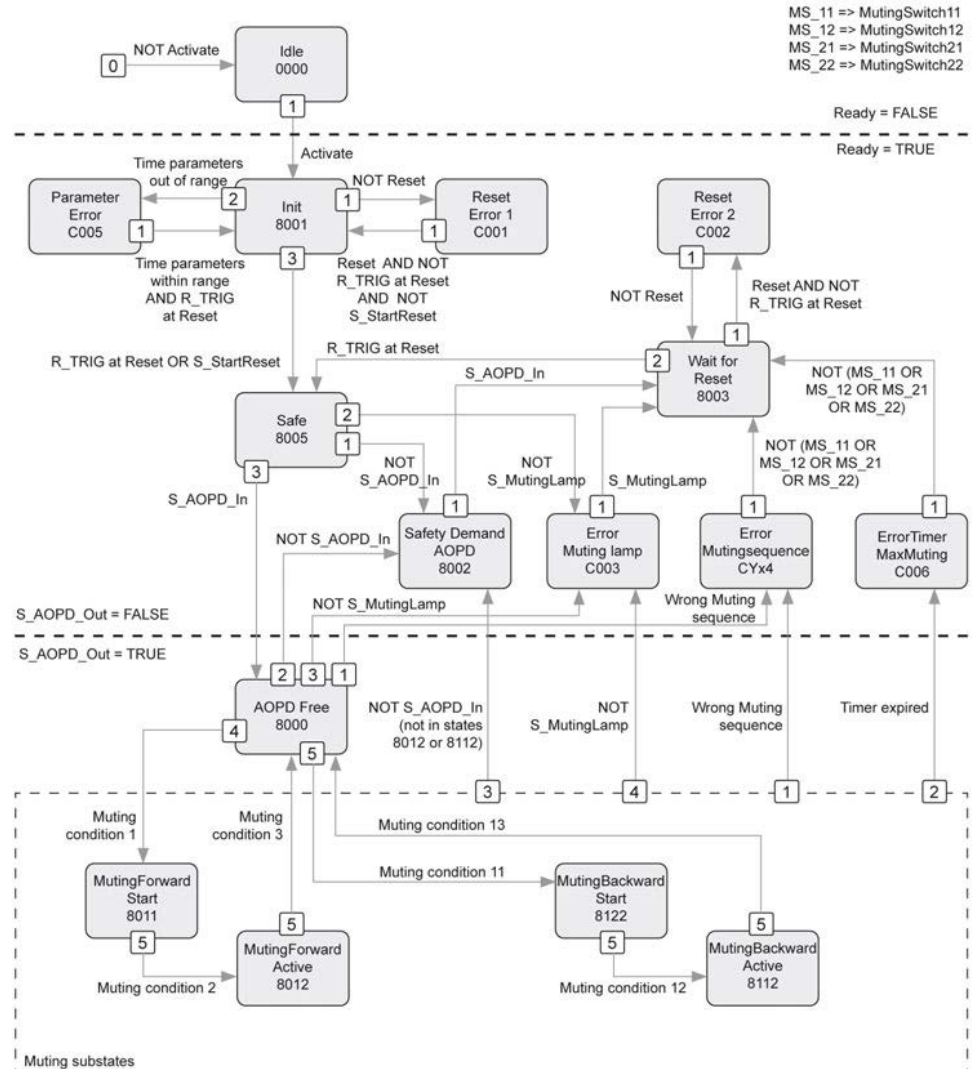
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8003	Wait for Reset	Sicherheitsanforderung oder Fehler wurde erkannt und ist jetzt durch den Operator quittiert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8005	Safe	Safety-Funktion aktiviert. Ready = TRUE S_AOPD_Out = FALSE S_MutingActive = FALSE Error = FALSE
8011	Muting Forward Start	Muting-Vorwärts, Sequenz ist in der Startphase und keine Sicherheitsanforderung. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE
8012	Muting Forward Active	Muting-Vorwärts, Sequenz ist aktiv. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8112	Muting Backward Active	Muting-Rückwärts, Sequenz aktiv. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = TRUE Error = FALSE
8122	Muting Backward Start	Muting-Rückwärts, Sequenz ist in der Startphase und keine Sicherheitsanforderung. Ready = TRUE S_AOPD_Out = TRUE S_MutingActive = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

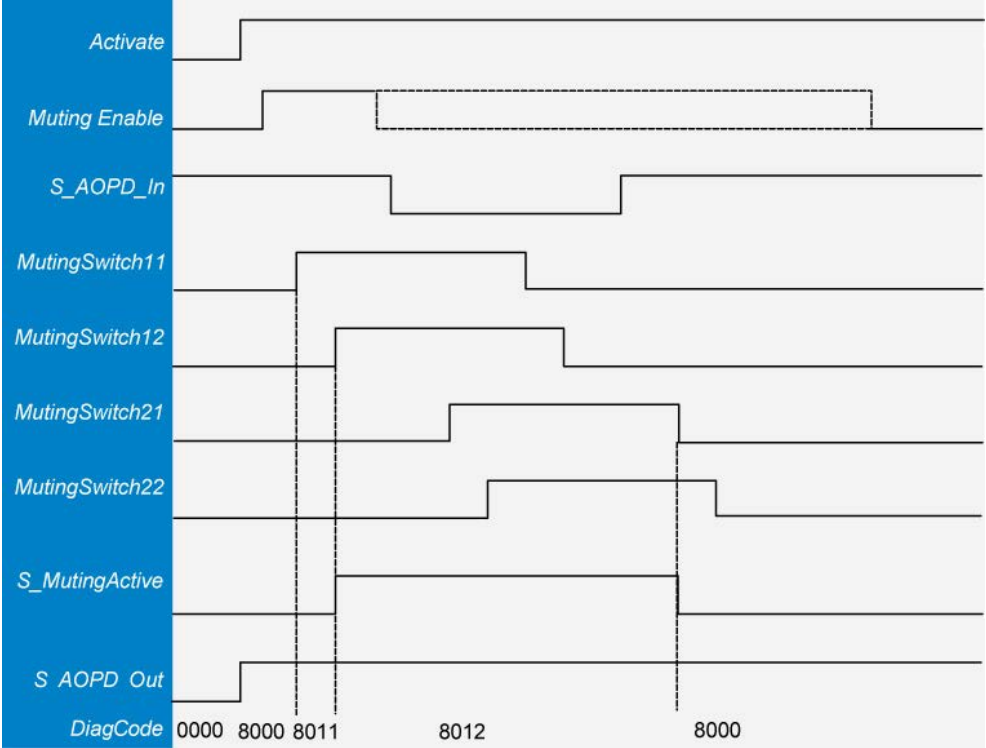
13.5.14.3 Status-Diagramm SF_MutingSeq



13. Funktionsblöcke

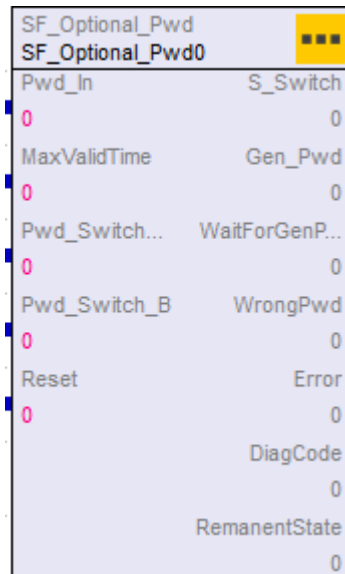
13.5 Komplexe Funktionsblöcke

13.5.14.4 Zeit-Diagramm SF_MutingSeq



13.5.15 SF_Optional_Pwd-Funktion

Dient zur Kennwortabfrage und Kennwortüberprüfung bei optionalen Modulen. Dieser Funktionsblock ist nur zusammen mit dem Funktionsblock → "SF_Optional_Switch-Funktion", Seite 303 zu verwenden.



Eingangsvariablen

Pwd_In	Eingegebenes Kennwort (DINT), eine Änderung löst den Umschaltvorgang aus.
MaxValidTime	Maximale Zeit (DINT), in der das generierte Bestätigungspasswort gültig ist. 0 ist unbegrenzte Zeit. Der Wert muss durch eine Konstante beschrieben werden.
Pwd_Switch_A	Kennwort (DINT) zum Aktivieren des optionalen Moduls. Der Wert muss durch eine Konstante beschrieben werden.
Pwd_Switch_B	Kennwort (DINT) zum Deaktivieren des optionalen Moduls. Der Wert muss durch eine Konstante beschrieben werden.
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

S_Switch	Schaltet den angeschlossenen Funktionsblock SF_Optional_Switch . (SAFEBOOL) FALSE: Optionales Modul eingeschaltet. TRUE: Optionales Modul ausgeschaltet. In der Zustandsübergangszeit (während auf das Bestätigungspasswort gewartet wird) ist dieser Ausgang auf ERROR gesetzt.
Gen_Pwd	Generiertes Bestätigungspasswort (DINT).
WaitForGenPwd	Angabe (BOOL), ob auf die Eingabe eines generierten Passworts von Gen_Pwd gewartet wird. FALSE: Es liegt kein gültiges Bestätigungspasswort an. TRUE: Das angegebene Bestätigungspasswort kann eingegeben werden.
WrongPwd	Information (BOOL), ob am Eingang Pwd_In ein erwartetes Kennwort anliegt. FALSE: Default Zustand TRUE: Es wurde ein falsches Passwort eingegeben.
Error	Fehleranzeige (BOOL)

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusanzeige (HDINT)
RemanentState	Remanenter Wert (DINT) wird im Flash des Moduls gespeichert, um nach einem Reset des Systems wieder in den alten Zustand zu kommen.

Der SF_Optional_Pwd-FB ist für die Prüfung der Passwörter optionaler Module zuständig und kontrolliert den angeschlossenen Funktionsblock → "SF_Optional_Switch-Funktion", Seite 303. Diese Kombination wird dazu verwendet, um Blöcke, die unter bestimmten Umständen Fehler auslösen würden, vom restlichen System zu trennen. Solche Blöcke können Hardwaremodule sein, die temporär entfernt oder auf Grund von Wartungs- und Service-Arbeiten abgestellt werden sollen. Damit diese nicht die gesamte Anlage blockieren, kann auf einen Alternativwert/Alternativblock umgeschaltet werden. Der abgeschaltete/unterdrückte Block kann keinen weiteren Fehler auslösen.

Besonderheit von diesem Funktionsblock ist, dass bei jedem Zustandsübergang ein Restart der Safety-CPU ausgelöst wird. Um weiters einen gesicherten Zustandsübergang zu erhalten, wird der Ausgang S_Switch während der Kennwortbestätigung auf ERROR gesetzt. Eine weitere Besonderheit ist, dass der Zustand des Funktionsblocks remanent gespeichert wird und dieser nach Aus- und wieder Einschalten der Safety-CPU wieder automatisch in den gespeicherten Status zurückkehrt.

13.5.15.1 Fehlererkennung

Der Funktionsblock erkennt folgende Fehlerzustände:

- Eingabe des falschen Kennwortes zur Bestätigung des Umschaltvorgangs
- Zeitüberschreitung bei Eingabe des Bestätigungskennworts

Wird im Zustand 8000 oder 8010 ein falsches Kennwort eingegeben, löst dies keinen Fehler aus, sondern wird nur durch ein TRUE auf den Ausgang WrongPwd signalisiert.

13.5.15.2 Fehlerverhalten

Der Ausgang DiagCode zeigt den entsprechenden Fehler-Code an und der Error-Ausgang wird auf TRUE gesetzt. Um den Fehlerzustand zu verlassen, muss dieser mit einer steigenden Flanke am Reset-Eingang quittiert werden. Dabei sollte der Eingang Pwd_In auf „0“ gesetzt sein.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Timeout for Pwd	Zeit im Zustand 8001 ist abgelaufen. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE RemanentState = 1

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

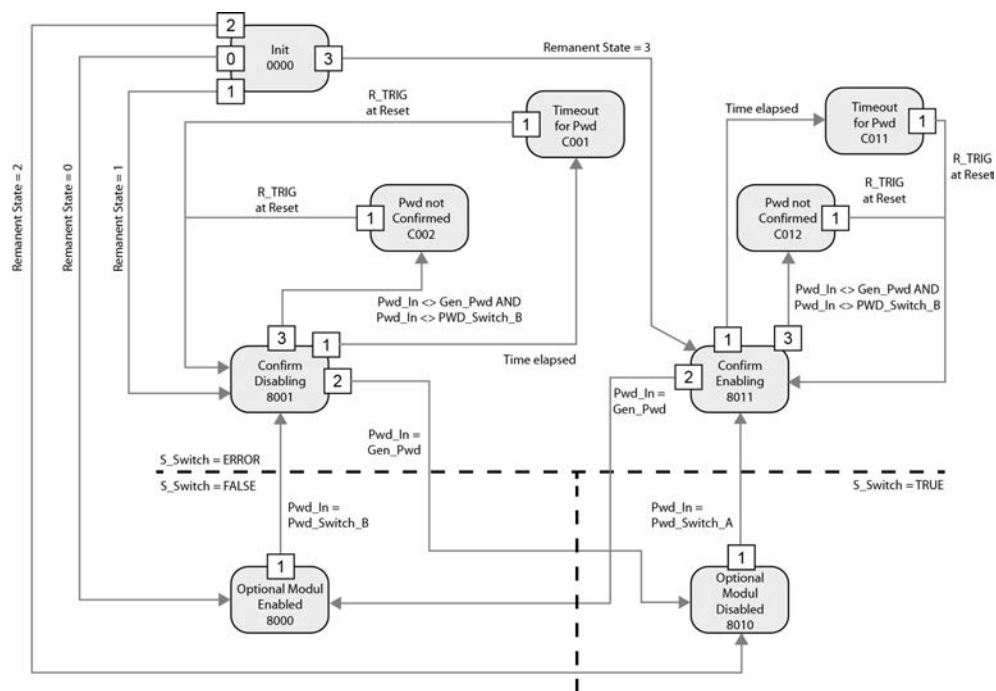
DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C011	Timeout for Pwd	Zeit im Zustand 8011 ist abgelaufen. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE RemanentState = 3
C002	Pwd not Confirmed	Pwd_In wurde im Zustand 8001 nicht auf generiertes Passwort verändert. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = TRUE Error = TRUE RemanentState = 1
C012	Pwd not Confirmed	Pwd_In wurde im Zustand 8011 nicht auf generiertes Passwort verändert. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = TRUE Error = TRUE RemanentState = 3
FB-spezifische Status-Codes (kein Fehler)		
0000	Init	Es wird automatisch auf den am remanenten Server gespeicherten Zustand weitergeschaltet (Initialwert des remanenten Servers ist 0). S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE
8000	Optional Modul Enabled	Optional Modul Unterdrückung abgeschaltet. Der nachfolgende Switch-Block wird angewiesen, S_InA auf den Ausgang durchzuschalten. Wird in diesem Zustand auf dem Eingang Pwd_In der Wert von Pwd_Switch_B angelegt, so schaltet der FUB in den Zustand 8001, um den Umschaltvorgang bestätigt zu bekommen. S_Switch = FALSE WaitForGenPwd = FALSE WrongPwd = Pwd_In != Pwd_Switch_A Error = FALSE RemanentState = 0
8010	Optional Modul Disabled	Optional Modul Unterdrückung aktiviert. Der nachfolgende Switch-Block wird angewiesen, S_InB auf den Ausgang durchzuschalten. Wird in diesem Zustand auf dem Eingang Pwd_In der Wert von Pwd_Switch_A angelegt, so schaltet der FUB in den Zustand 8011, um den Umschaltvorgang bestätigt zu bekommen. S_Switch = TRUE WaitForGenPwd = FALSE WrongPwd = Pwd_In != Pwd_Switch_B Error = FALSE RemanentState = 2

13. Funktionsblöcke

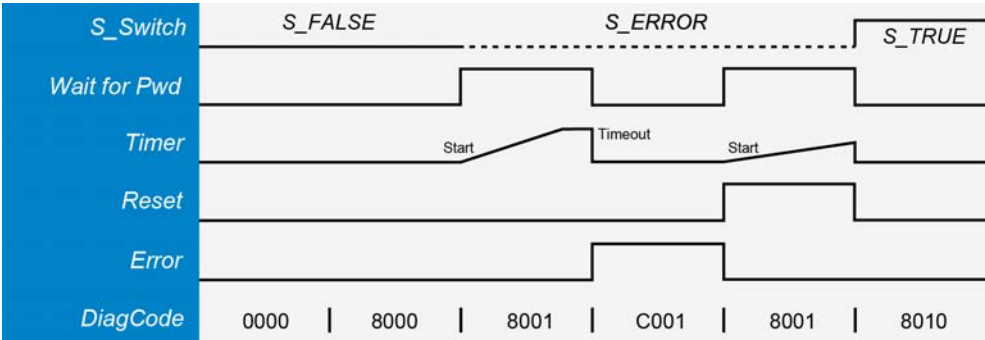
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8001	Confirm Disabling	Wartet auf die Bestätigung durch die Eingabe des Gen_Pwd auf Pwd_In, um mit dem Zustand 8010 den Abschaltvorgang abzuschließen. S_Switch = ERROR WaitForGenPwd = TRUE WrongPwd = FALSE Error = FALSE RemanentState = 1
8011	Confirm Enabling	Wartet auf die Bestätigung durch die Eingabe des Gen_Pwd auf Pwd_In, um mit dem Zustand 8000 den Einschaltvorgang abzuschließen. S_Switch = ERROR WaitForGenPwd = TRUE WrongPwd = FALSE Error = FALSE RemanentState = 3

13.5.15.3 Status-Diagramm SF_Optional_Pwd



13.5.15.4 Zeit-Diagramm SF_Optional_Pwd



13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.16 SF_Optional_PwdII-Funktion

Dient zur Kennwortabfrage und Kennwortüberprüfung bei optionalen Modulen. Dieser Funktionsblock ist nur zusammen mit dem Funktionsblock → "SF_Optional_Switch-Funktion", Seite 303 zu verwenden.

SF_Optional_PwdII	■ ■ ■
SF_Optional_PwdII0	
Pwd_In	S_Switch
0	0
MaxValidTime	Gen_Pwd
0	0
Pwd_Switch...	WaitForGenP...
0	0
Pwd_Switch_B	WrongPwd
0	0
MaxRetries	Error
0	0
Reset	DiagCode
0	0
	RemanentState
	0

Eingangsvariablen

Pwd_In	Eingegebenes Kennwort (DINT), eine Änderung löst den Umschaltvorgang aus.
MaxValidTime	Maximale Zeit (DINT), um einen Umschaltvorgang komplett abzuschließen (von 8000 nach 8010 bzw. von 8010 nach 8000). 0 bis maximal 8 Stunden. Der Wert muss durch eine Konstante beschrieben werden.
Pwd_Switch_A	Kennwort (DINT) zum Aktivieren des optionalen Moduls. Der Wert muss durch eine Konstante beschrieben werden.
Pwd_Switch_B	Kennwort (DINT) zum Deaktivieren des optionalen Moduls. Der Wert muss durch eine Konstante beschrieben werden.
MaxRetries	Maximale Anzahl (DINT) von Fehlversuchen für das Bestätigungspasswort.
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

S_Switch	Schaltet den angeschlossenen Funktionsblock → "SF_Optional_Switch-Funktion", Seite 303. (SAFEBOOL) FALSE: Optionales Modul eingeschaltet. TRUE: Optionales Modul ausgeschaltet.
Gen_Pwd	Generiertes Bestätigungspasswort (DINT).
WaitForGenPwd	Angabe (BOOL), ob auf die Eingabe eines generierten Passworts von Gen_Pwd gewartet wird. FALSE: Es liegt kein gültiges Bestätigungspasswort an. TRUE: Das angegebene Bestätigungspasswort kann eingegeben werden.
WrongPwd	Information (BOOL), ob am Eingang Pwd_In ein erwartetes Kennwort anliegt. FALSE: Default Zustand TRUE: Es wurde ein falsches Passwort eingegeben.

Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)
RemanentState	Remanenter Wert (DINT) wird im Flash des Moduls gespeichert, um nach einem Reset des Systems wieder in den alten Zustand zu kommen.

Der SF_Optional_PwdII-FB ist für die Prüfung der Passwörter optionaler Module zuständig und kontrolliert den angeschlossenen Funktionsblock → "SF_Optional_Switch-Funktion", Seite 303. Diese Kombination wird dazu verwendet, um Blöcke, die unter bestimmten Umständen Fehler auslösen würden, vom restlichen System zu trennen. Solche Blöcke können Hardwaremodule sein, die temporär entfernt oder auf Grund von Wartungs- und Service-Arbeiten abgestellt werden sollen. Damit diese nicht die gesamte Anlage blockieren, kann auf einen Alternativwert / Alternativblock umgeschaltet werden. Der abgeschaltete/unterdrückte Block kann keinen weiteren Fehler auslösen.

Besonderheit von diesem Funktionsblock ist, dass der Ausgang S_Switch während eines Fehlerzustands auf ERROR gesetzt, um einen gesicherten Zustandsübergang zu erhalten. Eine weitere Besonderheit ist, dass der Zustand des Funktionsblocks remanent gespeichert wird und dieser nach Aus- und wieder Einschalten der Safety-CPU wieder automatisch in den gespeicherten Status zurückkehrt.

Unterschiede zwischen Funktionsblock → "SF_Optional_Pwd-Funktion", Seite 292 und → "SF_Optional_PwdII-Funktion", Seite 297 ..

- Während der Phase der Rückbestätigung behält der Ausgang S_Switch beim Funktionsblock SF_Optional_PwdII den alten Wert bei und geht nicht auf SAFEBOOL_ERROR, wie beim Funktionsblock SF_Optional_Pwd.
- Beim Funktionsblock SF_Optional_PwdII wird beim remanenten Speichern kein Reboot der Safety-CPU ausgelöst.

13.5.16.1 Fehlererkennung

Der Funktionsblock erkennt folgende Fehlerzustände:

- Eingabe des falschen Kennwortes zur Bestätigung des Umschaltvorgangs
- Zeitüberschreitung bei Eingabe des Bestätigungskennworts

Wird im Zustand 8000 oder 8010 ein falsches Kennwort eingegeben, löst dies keinen Fehler aus, sondern wird nur durch ein TRUE auf den Ausgang WrongPwd signalisiert.

13.5.16.2 Fehlerverhalten

Der Ausgang DiagCode zeigt den entsprechenden Fehler-Code an und der Error-Ausgang wird auf TRUE gesetzt. Um den Fehlerzustand zu verlassen, muss dieser mit einer steigenden Flanke am Reset-Eingang quittiert werden. Dabei sollte der Eingang Pwd_In auf „0“ gesetzt sein.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Timeout for Pwd	Zeit im Zustand 8001 oder 8002 ist abgelaufen. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE RemanentState = 1
C011	Timeout for Pwd	Zeit im Zustand 8011 oder 8012 ist abgelaufen. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE RemanentState = 3
C002	MaxRetries	Anzahl der Versuche wurde im Zustand 8001 überschritten. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = TRUE Error = TRUE RemanentState = 1
C012	MaxRetries	Anzahl der Versuche wurde im Zustand 8011 überschritten. S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = TRUE Error = TRUE RemanentState = 3
FB-spezifische Status-Codes (kein Fehler)		
0000	Init	Es wird automatisch auf den am remanenten Server gespeicherten Zustand weitergeschaltet (Initialwert des remanenten Servers ist 0). S_Switch = ERROR WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE
8000	Optional Modul Enabled	Optional Modul Unterdrückung abgeschaltet. Der nachfolgende Switch-Block wird angewiesen, S_InA auf den Ausgang durchzuschalten. Wird in diesem Zustand auf dem Eingang Pwd_In der Wert von Pwd_Switch_B angelegt, so schaltet der FUB in den Zustand 8001, um den Umschaltvorgang bestätigt zu bekommen. S_Switch = FALSE WaitForGenPwd = FALSE WrongPwd = Pwd_In != Pwd_Switch_A Error = FALSE RemanentState = 0

13. Funktionsblöcke

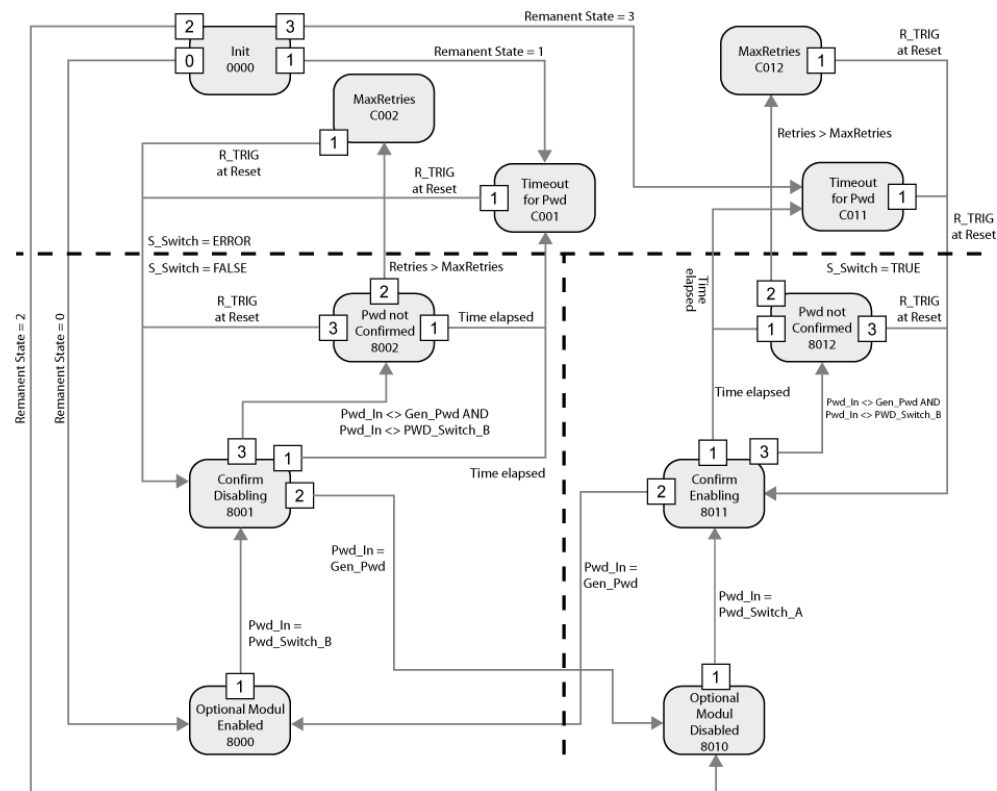
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8010	Optional Modul Disabled	Optional Modul Unterdrückung aktiviert. Der nachfolgende Switch-Block wird angewiesen, S_InB auf den Ausgang durchzuschalten. Wird in diesem Zustand auf dem Eingang Pwd_In der Wert von Pwd_Switch_A angelegt, so schaltet der FUB in den Zustand 8011, um den Umschaltvorgang bestätigt zu bekommen. S_Switch = TRUE WaitForGenPwd = FALSE WrongPwd = Pwd_In != Pwd_Switch_B Error = FALSE RemanentState = 2
8001	Confirm Disabling	Wartet auf die Bestätigung durch die Eingabe des Gen_Pwd auf Pwd_In, um mit dem Zustand 8010 den Abschaltvorgang abzuschließen. S_Switch = FALSE WaitForGenPwd = TRUE WrongPwd = FALSE Error = FALSE RemanentState = 1
8011	Confirm Enabling	Wartet auf die Bestätigung durch die Eingabe des Gen_Pwd auf Pwd_In, um mit dem Zustand 8000 den Einschaltvorgang abzuschließen. S_Switch = TRUE WaitForGenPwd = TRUE WrongPwd = FALSE Error = FALSE RemanentState = 3
8002	Pwd not Confirmed	Pwd_In wurde im Zustand 8001 nicht auf generiertes Passwort verändert. S_Switch = FALSE WaitForGenPwd = FALSE WrongPwd = TRUE Error = FALSE RemanentState = 1
8012	Pwd not Confirmed	Pwd_In wurde im Zustand 8011 nicht auf generiertes Passwort verändert. S_Switch = TRUE WaitForGenPwd = FALSE WrongPwd = TRUE Error = FALSE RemanentState = 3

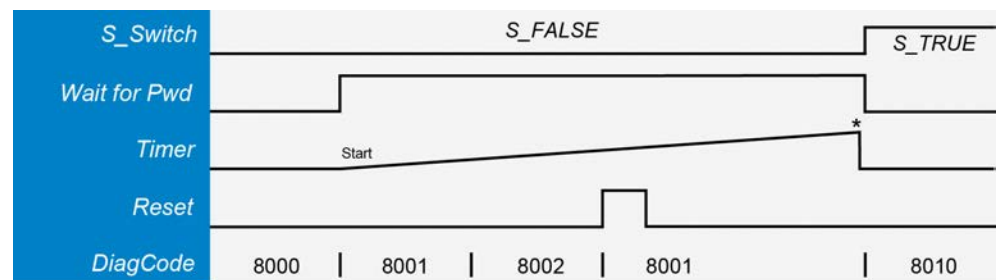
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.16.3 Status-Diagramm SF_Optional_PwdII



13.5.16.4 Zeit-Diagramm SF_Optional_PwdII



* nicht bis MaxValue!

13.5.16.5 Sicherheitsmaßnahmen und -hinweise

Durch die Verwendung des Funktionsblocks „SF_OPTIONAL_PWDII“ werden grundsätzlich alle Sicherheitsfunktionen betroffen; daher sind unbedingt die nachfolgenden Maßnahmen und Gefahrenhinweise zu berücksichtigen. Die Nichtbeachtung dieser Warnhinweise kann zu schweren Körperverletzungen oder zum Tod führen.

GEFAHR



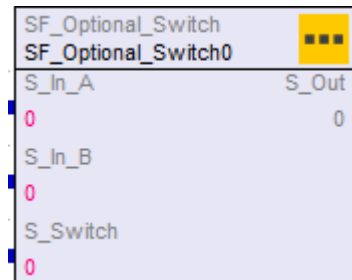
- Der Anwender der Maschine muss im Hinblick auf die möglichen Gefahren, die von der Verwendung des Funktionsblock „SF_OPTIONAL_PWDII“ ausgehen können, und unter Berücksichtigung der für die spezifische Anwendung geltenden normativen Anforderungen (beispielsweise hydraulische Gesenkbiegepressen) eine Risikoanalyse erstellen und die erforderlichen Maßnahmen der Risikominderung treffen.
- Die Gültigkeitsdauer des von der Safety-CPU errechneten Codes (Anmeldezeit) muss in jedem Fall so kurz wie möglich sein; insbesondere muss sichergestellt werden, dass sie nicht schichtübergreifend wirksam ist. Es muss gewährleistet sein, dass Passworteingabe und Bestätigung des automatisch errechneten Codes von ein und dem selben autorisierten Einrichter vorgenommen wird.
- Die Maschine muss während der Gültigkeitsdauer des automatisch errechneten Codes permanent im Blickfeld des Einrichters der Maschine sein.
- Außer dem Einrichter der Maschine dürfen sich innerhalb der Grenzen der Maschine gemäß EN 12100 keine weiteren Personen aufhalten. Insbesondere muss der freie Zugang zur Maschine durch geeignete Maßnahmen verhindert werden.
- Befehlsgeräte ohne NOT-HALT-Funktion sind so zu installieren, dass von jeder Position aus in Handreichweite jederzeit ein NOT-HALT ausgelöst werden kann.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.17 SF_Optional_Switch-Funktion

Über einen Schaltereingang kann selektiert werden, welcher Eingang auf den Ausgang durchgeschaltet werden soll.



Eingangsvariablen

S_In_A	Eingang A (SAFEBOOL)
S_In_B	Eingang B (SAFEBOOL)
S_Switch	Schaltereingang zur Selektion welcher Eingang (S_In_A oder S_In_B) auf den Ausgang durchgeschaltet werden soll. Sollte dieser Eingang einen ungültigen SAFEBOOL-Wert annehmen oder ERROR sein, dann wird auch der Ausgang auf ERROR geschaltet (SAFEBOOL).

Ausgangsvariablen

S_Out	Wert des durchgeschalteten Eingangs oder ERROR, wenn am Schaltereingang S_Switch ein nicht gültiger Wert anliegt (SAFEBOOL).
-------	--

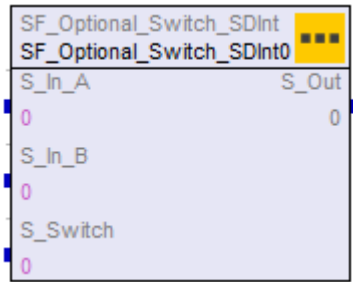
Besonderheit ist, dass eine Prüfung auf Fehler nicht bei den Eingängen, sondern auf den Ausgang durchgeführt wird. Dadurch kann auf den nicht durchgeschalteten Eingang ein nicht definierter Wert anliegen, ohne dass der Ausgang auf ERROR geht.

13.5.17.1 Wertetabelle

S_Switch	S_Out
FALSE	S_In_A
TRUE	S_In_B
ERROR	ERROR
undefiniert	ERROR

13.5.18 SF_Optional_Switch_SDInt

Über einen Schaltereingang kann selektiert werden, welcher Eingang auf den Ausgang durchgeschaltet werden soll.



Eingangsvariablen	
S_In_A	Eingang A (SAFEDINT)
S_In_B	Eingang B (SAFEDINT)
S_Switch	Schaltereingang zur Selektion welcher Eingang (S_In_A oder S_In_B) auf den Ausgang durchgeschaltet werden soll. Sollte dieser Eingang SAFEBOOL_ERROR annehmen, dann wird auch der Ausgang auf SAFEDINT_ERROR geschaltet. (SAFEBOOL)
Ausgangsvariablen	
S_Out	Wert des durchgeschalteten Eingangs oder ERROR, wenn am Schaltereingang S_Switch ein ERROR anliegt. (SAFEDINT).

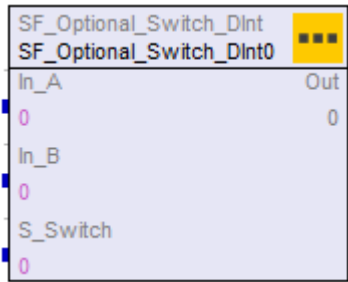
Besonderheit ist, dass eine Prüfung auf Fehler nicht bei den Eingängen, sondern auf dem Ausgang durchgeführt wird. Dadurch kann auf den nicht durchgeschalteten Eingang ein nicht definierter Wert anliegen, ohne dass der Ausgang auf ERROR geht.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.19 SF_Optional_Switch_Dint

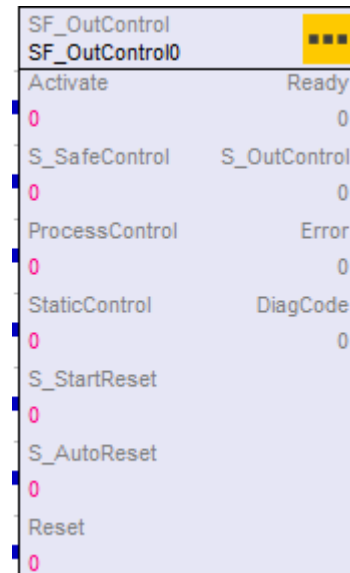
Über einen Schaltereingang kann selektiert werden, welcher Eingang auf den Ausgang durchgeschaltet werden soll.



Eingangsvariablen	
In_A	Eingang A (DINT)
In_B	Eingang B (DINT)
S_Switch	Schaltereingang zur Selektion welcher Eingang (In_A oder In_B) auf den Ausgang durchgeschaltet werden soll. Sollte dieser Eingang SAFEBOOL_ERROR annehmen, dann wird auf den Ausgang IN_A geschaltet. (SAFEBOOL)
Ausgangsvariablen	
Out	Wert des durchgeschalteten Eingangs oder In_A, wenn am Schaltereingang S_Switch ein ERROR anliegt. (DINT)

13.5.20 SF_OutControl-Funktion

Kontrolliert einen Safety-Ausgang mit einem Signal der funktionalen Applikation und einem Safety-Signal mit optionaler Starthemmung.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_SafeControl	Control-Signal eines vorhergehenden Sicherheits-FBs. (z.B. SF_Estop, SF_GuardMonitoring, SF_TwoHandControlTypell bzw. andere). FALSE: Die vorhergehenden Sicherheits-FBs sind in sicherem Zustand. TRUE: Die vorhergehenden Sicherheits-FBs aktivieren die Sicherheitssteuerung.
ProcessControl	Steuersignal der funktionalen Anwendung. (BOOL) FALSE: Anfrage, um S_OutControl auf FALSE zu setzen. TRUE: Anfrage, um S_OutControl auf TRUE zu setzen.
StaticControl	Optionale Voraussetzungen für die Prozess-Steuerung. (BOOL) FALSE: Dynamische Veränderung bei ProcessControl (FALSE => TRUE) wird nach der Block-Aktivierung oder einer ausgelösten Sicherheitsfunktion benötigt. Zusätzlicher Funktionsstart benötigt. TRUE: Es wird keine dynamische Veränderung auf ProcessControl (FALSE => TRUE) nach der Block-Aktivierung oder einer ausgelösten Sicherheitsfunktion benötigt.
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL).
S_AutoReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der Not-Halt-Schalter wieder freigegeben wird (SAFEBOOL)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Ausgänge gültig sind. (BOOL)
S_OutControl	Kontrolliert den angeschlossenen Aktuator. (SAFEBOOL) FALSE: Disable der angeschlossenen Aktuatoren TRUE: Enable der angeschlossenen Aktuatoren

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

13.5.20.1 Allgemeines

Der SF_OutControl-FB ist ein Ausgangstreiber für einen Sicherheitsausgang. Der Sicherheitsausgang wird unter Verwendung eines Signal von der funktionalen Anwendung (ProcessControl/BOOL, um den Prozess zu steuern) und eines Signals von der Sicherheitsanwendung (S_SafeControl / SAFEBOOL, um die Sicherheitsfunktion zu steuern) über S_OutControl gesteuert.

13.5.20.2 Optionale Bedingungen zur Prozesskontrolle (ProcessControl)

- Ein zusätzlicher Funktionsstart (ProcessControl FALSE => TRUE) ist nach der Aktivierung des Blocks oder Feedbacks vom sicheren Signal (S_SafeControl) erforderlich. Ein statisches TRUE-Signal bei ProcessControl setzt S_OutControl nicht auf TRUE.
- Ein zusätzlicher Funktionsstart (ProcessControl FALSE => TRUE) ist nach der Blockaktivierung oder eines Feedbacks des sicheren Signals (S_SafeControl) nicht erforderlich. Ein statisches TRUE-Signal bei ProcessControl setzt S_OutControl auf TRUE, wenn die übrigen Voraussetzungen erfüllt sind.

13.5.20.3 Optional startup inhibits

- Start-Verhinderung nach Aktivierung des Funktionsblocks.
- Start-Verhinderung nach einer Unterbrechung der Sicherungseinrichtung.

Die StaticControl-, S_StartReset- und S_AutoReset-Eingänge sollten nur aktiviert werden, wenn sichergestellt ist, dass beim Starten des PES keine gefährliche Situation auftreten kann.

13.5.20.4 Fehlererkennung

Die folgenden Bedingungen erzwingen eine Umstellung auf den Fehlerstatus:

- Ungültiges statisches Reset-Signal im Prozess
- Ungültiges statisches ProcessControl Signal.
- ProcessControl und Reset sind durch einen Programmierfehler falsch verbunden.

13.5.20.5 Fehlerverhalten

Im Falle eines Fehlers wird der Ausgang S_OutControl auf FALSE gesetzt und bleibt im sicheren Zustand. Um die Reset-, Init- oder Lock-Fehlerzustände zu verlassen, muss der Reset-Eingang auf FALSE gesetzt werden. Um den Control-Fehlerzustand zu verlassen, muss der ProcessControl-Eingang auf FALSE gesetzt werden.

Nach dem Übergang von S_SafeControl auf TRUE kann die optionale Startup-Inhibit durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden. Nach der Blockaktivierung kann die optionale Start-Verhinderung durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden.

13. Funktionsblöcke

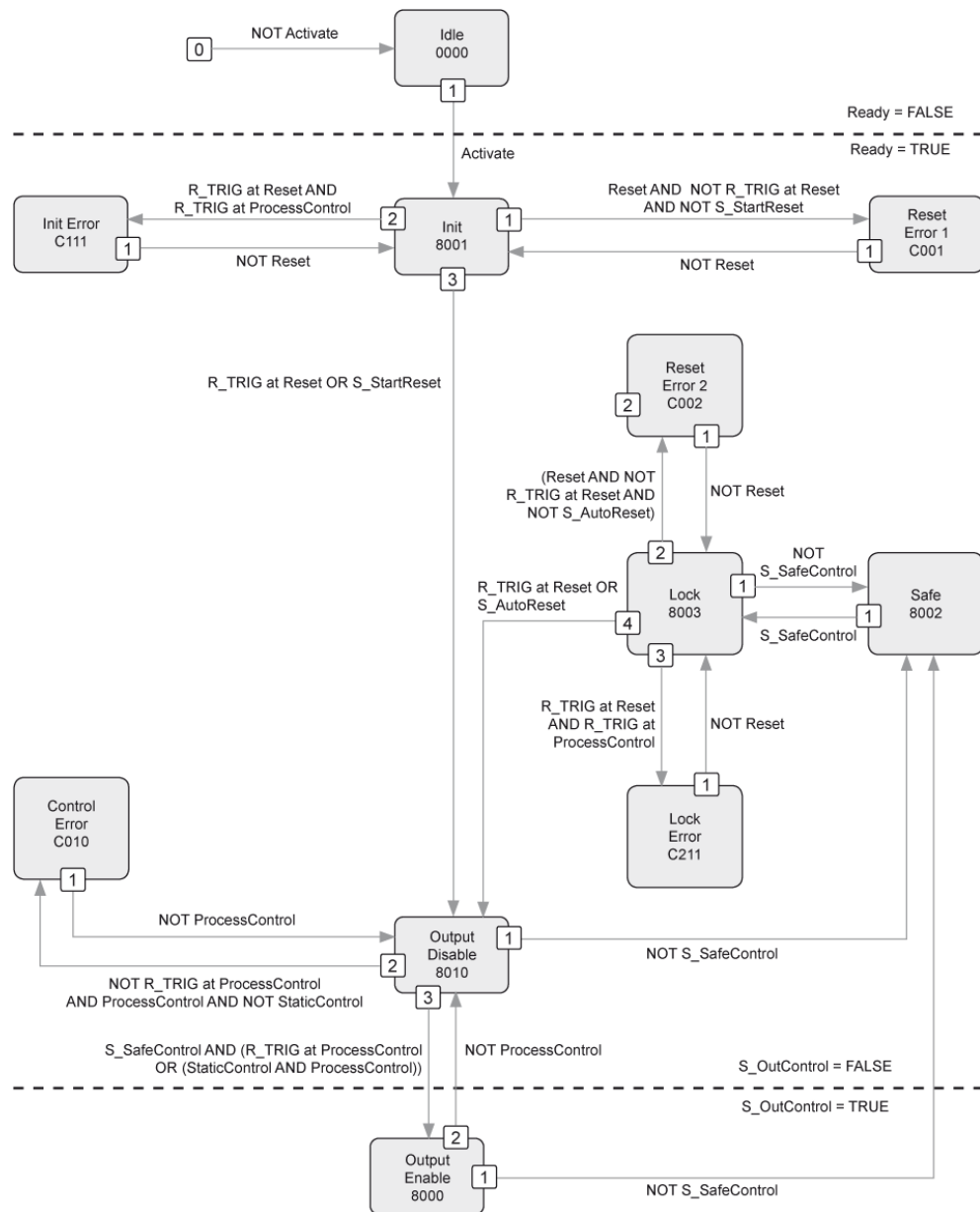
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal im Zustand 8001. Ready = TRUE S_OutControl = FALSE Error = TRUE
C002	Reset Error 2	Statisches Reset-Signal im Zustand 8003. Ready = TRUE S_OutControl = FALSE Error = TRUE
C010	Control Error	Statisches Signal bei ProcessControl im Zustand 8010. Ready = TRUE S_OutControl = FALSE Error = TRUE
C111	Init Error	Gleichzeitig steigende Flanke bei Reset und ProcessControl im Zustand 8001. Ready = TRUE S_OutControl = FALSE Error = TRUE
C211	Lock Error	Gleichzeitig steigende Flanke bei Reset und ProcessControl im Zustand 8003. Ready = TRUE S_OutControl = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_OutControl = FALSE Error = FALSE
8001	Init	Blockaktivierung Start Verhinderung ist aktiv. Reset erforderlich. Ready = TRUE S_OutControl = FALSE Error = FALSE
8002	Safe	Ausgelöste Sicherheitsfunktion. Ready = TRUE S_OutControl = FALSE Error = FALSE
8003	Lock	Start-Verhinderung der Sicherheitsfunktion ist aktiv. Reset erforderlich. Ready = TRUE S_OutControl = FALSE Error = FALSE
8010	Output Disable	Prozess-Steuerung ist nicht aktiv. Ready = TRUE S_OutControl = FALSE Error = FALSE
8000	Output	Aktivierung der Prozess-Steuerung ist aktiv und die Sicherheit ist gegeben. Ready = TRUE S_OutControl = TRUE Error = FALSE

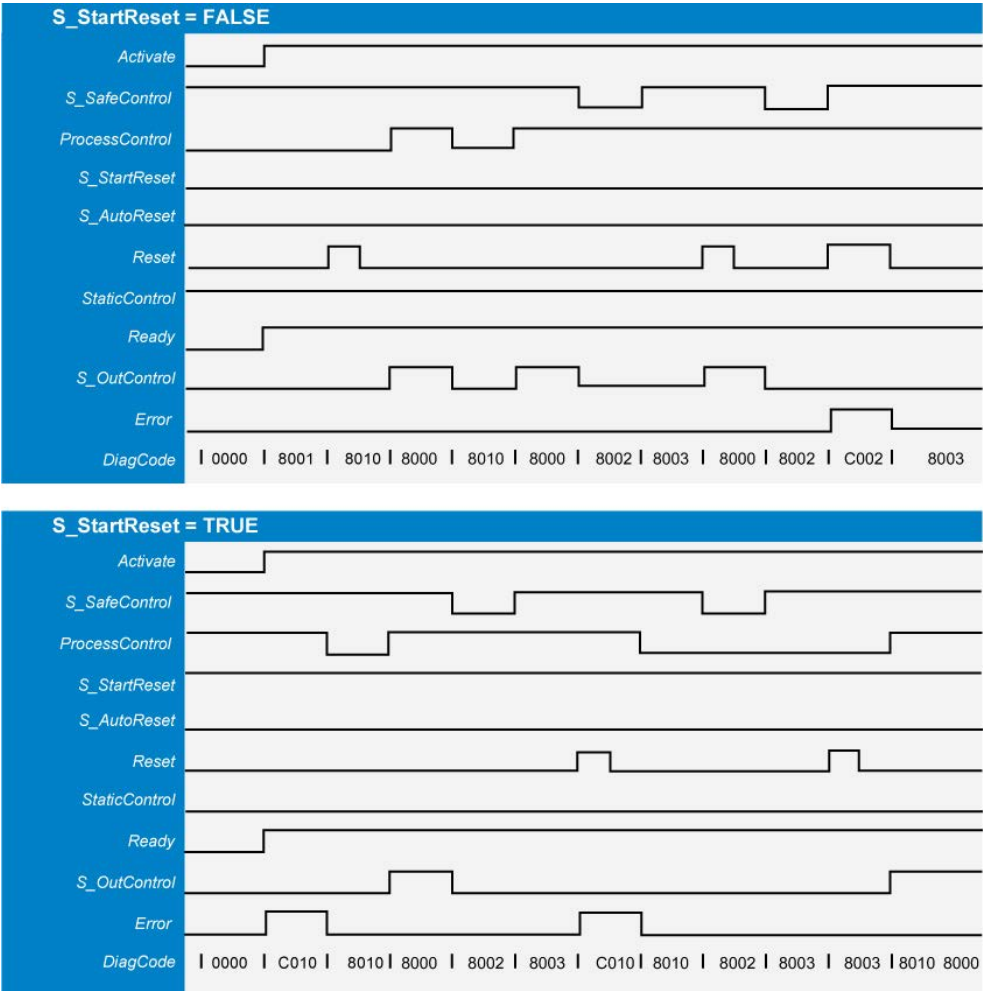
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.20.6 Status-Diagramm SF_OutControl



13.5.20.7 Zeit-Diagramm SF_OutControl



13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.21 SF_SafetyRequest-Funktion

Dieser Funktionsblock dient als Schnittstelle zu einem generischen Aktuator, z.B. einem Safety Drive oder einem Sicherheitsventil, um den Aktuator in einen sicheren Zustand zu versetzen.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_OpMode	Angeforderter Safe Modus eines Aktuators (SAFEBOOL). FALSE: Safe-Modus wird angefordert TRUE: Betriebsmodus wird angefordert
S_Acknowledge	Bestätigung des Aktuators, wenn sich der Aktuator in einem sicheren Zustand befindet. (SAFEBOOL) FALSE: Betriebsmodus (non-safe) TRUE: Safe Modus
MonitoringTime	Überwachung der Reaktionszeit zwischen dem angeforderten Safe Modus (S_OpMode auf FALSE) und der Bestätigung des Aktuators (S_Acknowledge auf TRUE). (DINT)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	TRUE: Wenn Funktionsblock aktiviert ist und die Ausgangs-Ergebnisse gültig sind. (BOOL)
S_SafetyActive	Bestätigung des Safe Modus. FALSE: Non-safe Modus TRUE: Safe Modus (SAFEBOOL)
S_SafetyRequest	Angeforderter Modus eines Aktuators FALSE: Safe-Modus wird angefordert TRUE: Non-safe Modus (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Dieser FB stellt die Schnittstelle zwischen dem sicherheitsbezogenen System eines generischen Aktuators zur Verfügung. Dies bedeutet, dass die sicherheitsrelevanten Funktionen des Aktuators im Anwendungsprogramm zur Verfügung stehen. Allerdings gibt es nur zwei binäre Signale zum Steuern des sicheren Zustands des generischen Aktuators - einen zum Abfragen und einen zum Erhalten der Bestätigung.

Die Sicherheitsfunktion wird durch den Aktuator zur Verfügung gestellt. Deshalb initiiert der FB nur die Anfrage, überwacht diese und setzt den Ausgang, wenn der Aktuator den sicheren Zustand erkennt. Dies wird durch den "S_SafetyActive" Ausgang angegeben.

Dieser FB definiert keine generischen aktuator-spezifischen Parameter. Diese sollten im generischen Aktuator direkt festgelegt werden. Es schaltet den generischen Aktuator vom Betriebsmodus in einen sicheren Zustand.

13.5.21.1 Fehlererkennung

Der FB erkennt, ob der Aktuator innerhalb der Überwachungszeit nicht im sicheren Zustand ist.

Die FB erkennt, ob das Quittierungssignal verloren geht, während die Anfrage immer noch aufrecht ist.

Der FB erkennt ein statisches Reset-Signal.

Externe FB-Fehler:

Es gibt keine externen Fehler, da es kein Fehler Bit / Informationen gibt, die vom generischen Aktuator zur Verfügung gestellt werden.

13.5.21.2 Fehlerverhalten

Im Falle eines Fehlers wird der S_SafetyActive-Ausgang auf FALSE gesetzt. Ein Fehler muss durch eine steigende Flanke am Reset-Eingang quittiert werden. Um den Funktionsblock nach einem Reset fortzusetzen, muss die S_OpMode-Anfrage auf TRUE gesetzt werden.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C002	Acknowledge Lost	Keine Reaktion auf Quittierung im sicheren Zustand. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = TRUE
C003	MonitoringTime Elapsed	Keine Reaktion auf S_OpMode Anfrage innerhalb der Überwachungszeit. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = TRUE
C004	Reset Error 2	Statisches Reset wurde im Zustand C002 erkannt. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = TRUE

13. Funktionsblöcke

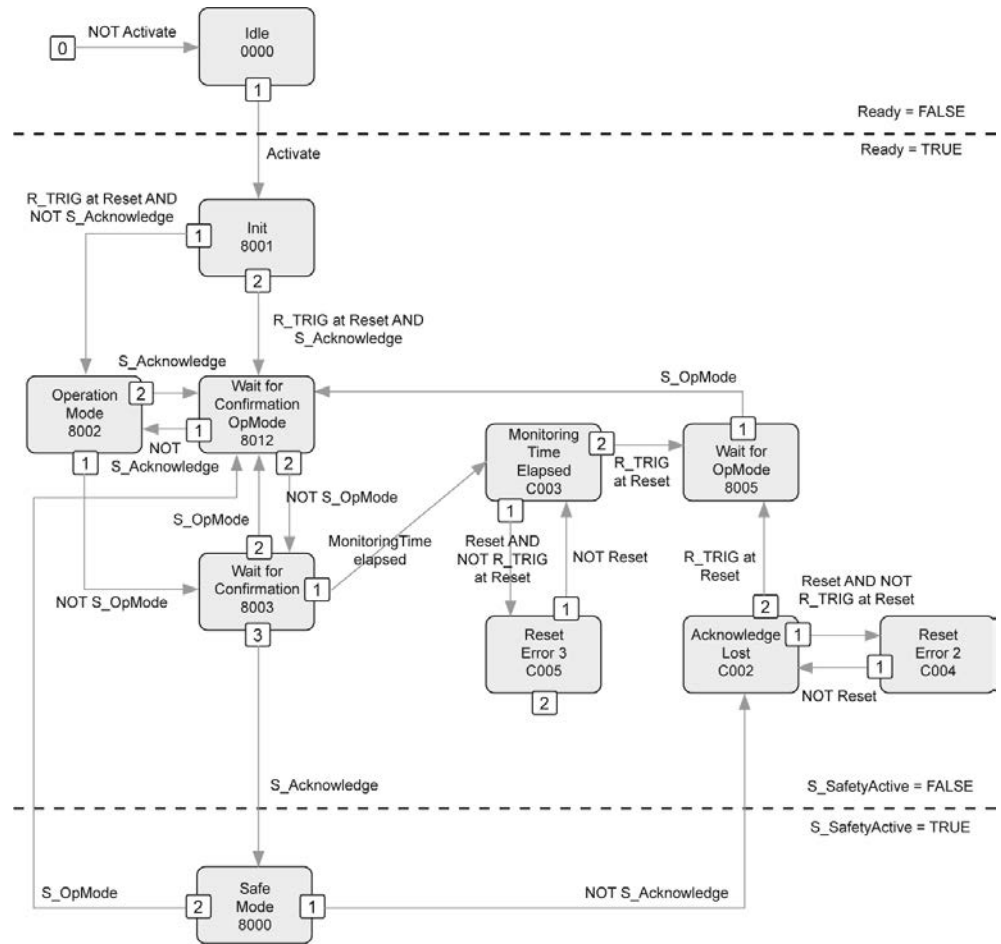
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C005	Reset Error 3	Statisches Reset wurde im Zustand C003 erkannt (MonitoringTime abgelaufen). Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = FALSE
8000	Safe Mode	Aktuator ist im sicheren Zustand. Ready = TRUE S_SafetyActive = TRUE S_SafetyRequest = FALSE Error = FALSE
8001	Init	Zustand nach dem Aktivieren. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = FALSE
8002	Operation Mode	Betriebsart ohne Quittierung des sicheren Zustands. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = TRUE Error = FALSE
8012	Wait for Confirmation OpMode	Betriebsmodus mit Quittierung des sicheren Zustands. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = TRUE Error = FALSE
8003	Wait for Confirmation	Warten auf die Bestätigung des Aktuators. (System Interface). Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = FALSE
8005	Wait for OpMode	Fehler wurde quittiert. Warten bis S_OpMode auf TRUE gesetzt wird. Ready = TRUE S_SafetyActive = FALSE S_SafetyRequest = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

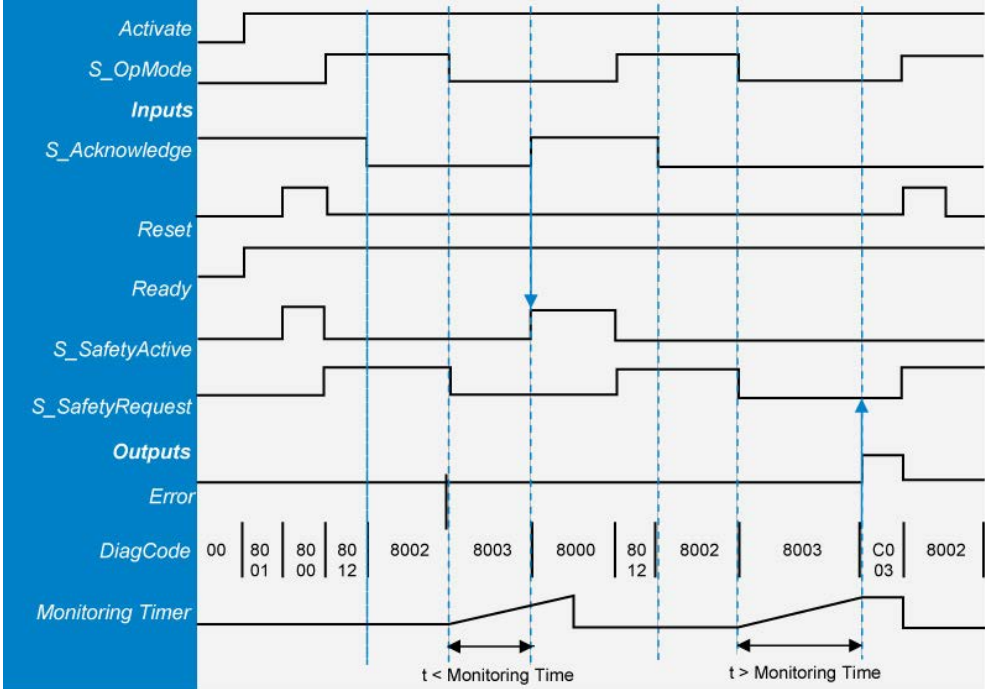
13.5.21.3 Status-Diagramm SF_SafetyRequest



13. Funktionsblöcke

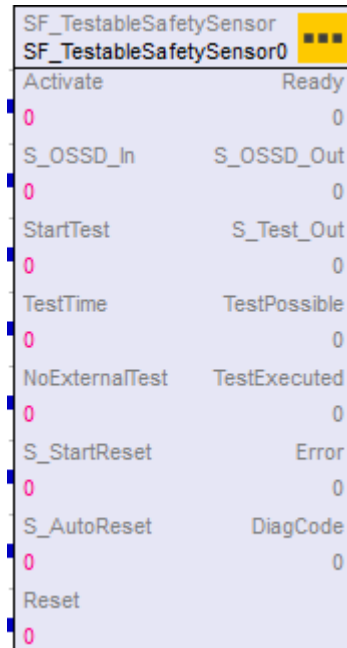
13.5 Komplexe Funktionsblöcke

13.5.21.4 Zeit-Diagramm SF_SafetyRequest



13.5.22 SF_TestableSafetySensor-Funktion

Dieser Funktionsblock erkennt zum Beispiel den Verlust einer Sensor-Erkennung, eine längere Reaktionszeit als angegeben und statische ON Signale in Einzel-Kanal-Sensor-Systemen. Er kann für extern testbare Sicherheits-Sensoren (ESPE) verwendet werden.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL).
S_OSSD_In	OSSD = Output Signal Switching Device Sensorstatus (z. B. Lichtvorhang) (SAFEBOOL) FALSE: Safety-Sensor im Teststaus oder im Zustand einer sicherheitsrelevanten Reaktion TRUE: Safety-Sensor im Status normaler Betriebsbedingungen.
StartTest	Eingang um den Sensortest zu starten. Setzt „S_TestOut“ und beginnt mit der internen Überwachung der Zeit-Funktion. (BOOL) FALSE: Kein Test erforderlich. TRUE: Test angefordert.
TestTime	Konstante. Bereich: 0 - 150 ms. Testzeit des Safety-Sensors. (DINT)
NoExternalTest	Konstante. Zeigt an, wenn ein externer manueller Sensortest unterstützt wird. (BOOL) FALSE: Ein externer manueller Sensortest wird unterstützt. Erst nach einer kompletten manuellen Sensor-Schaltsequenz ist eine automatische Prüfung möglich. TRUE: Ein externer manueller Sensortest wird nicht unterstützt. Eine automatische Prüfung ist möglich, ohne eine manuelle Sensor-Schaltsequenz nach fehlerhaftem automatischen Sensortest.
S_StartReset	FALSE: Manueller Reset TRUE: Automatischer Reset nachdem die CPU gestartet wird (SAFEBOOL).
S_AutoReset	FALSE: Manueller Reset

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

	TRUE: Automatischer Reset nachdem der Not-Halt-Taster wieder freigegeben wird (SAFEBOOL).
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige dass der Funktionsblock aktiviert ist und die Ausgangs-Ergebnisse gültig sind (BOOL).
S_OSSD_Out	Sicherheitsrelevanter Ausgang über den Zustand der ESPE. (SAFEBOOL) FALSE: Sicherheitsrelevante Anfrage an den Sensor oder Testfehler. TRUE: Keine sicherheitsgerichtete Anfrage an den Sensor und kein Testfehler.
S_TestOut	Wird mit dem Test-Eingang des Sensors verbunden. Obwohl dieser Ausgang als SAFEBOOL ausgeführt ist, wird dieses Signal in der Praxis oft mit einem BOOL ausgegeben. FALSE: Test angefordert TRUE: Test nicht angefordert
TestPossible	Feedback-Signal an den Prozess. (BOOL) FALSE: Ein automatischer Sensortest ist nicht möglich. TRUE: Ein automatischer Sensortest ist möglich.
TestExecuted	Eine positive Signal Flanke zeigt die erfolgreiche Ausführung des automatischen Sensortests. (BOOL) FALSE: - Ein automatischer Sensortest wurde noch nicht ausgeführt. - Ein automatischer Sensortest ist aktiv. - Ein automatischer Sensortest war fehlerhaft. TRUE: - Ein Sensortest wurde erfolgreich durchgeführt.
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Typ 2 ESPE muss ein Mittel zur regelmäßigen Prüfung enthalten, um einen gefährlichen Fehler zu erkennen (z.B. Verlust der Sensor-Erkennungsfähigkeit, Reaktionszeit überschreitet den angegebenen Wert). Das Prüfsignal muss die Betätigung der Abtasteinrichtung simulieren und die Dauer der Prüfung darf 150 ms nicht überschreiten. Die Prüfung muss sicherstellen, dass im Falle eines Lichtschutzgitters jeder Lichtstrahl nach den Lieferanten-Spezifikationen funktioniert. Sollte die Prüfung von einem externen sicherheitsrelevanten Steuerungssystem eingeleitet werden (z.B. einer Maschine), muss der ESPE mit geeigneten Eingangsmöglichkeiten vorgesehen werden (z.B. Terminals). Die ESPE muss im Bezug auf die Produktnormen EN IEC 61496-1, -2 und -3 und die erforderlichen Kategorien nach EN ISO 13849 ausgewählt werden. Diese muss durch eine separate Funktionalität überwacht werden, sodass der Test in angemessenen Abständen eingeleitet wird. Die S_StartReset- und S_AutoReset-Eingänge sollten nur aktiviert werden, wenn sichergestellt ist, dass beim Starten der PES keine gefährliche Situation auftreten kann.

13.5.22.1 Test-Modus

1. StartTest = TRUE: S_TestOut = FALSE. Überwachungszeit starten
2. S_TestOut Signal stoppt den Sender (Überwachung der TestTime zum ersten Mal gestartet)
3. S_OSSD_In ändert sich von TRUE auf FALSE (Überwachung der TestTime zum zweiten Mal gestartet)
4. S_TestOut ändert sich von FALSE auf TRUE
5. Sender starten
6. Sensor S_OSSD_In ändert sich von FALSE auf TRUE
7. Überwachungszeit stoppen
8. S_OSSD_Out während der Prüfung auf TRUE gesetzt

13.5.22.2 Optionale Startup inhibits

- Start-Verhinderung nach Aktivierung des Funktionsblocks.
- Start-Verhinderung nach einer Unterbrechung der Sicherungseinrichtung.

13.5.22.3 Fehlererkennung

Die folgenden Bedingungen erzwingen eine Umstellung auf den Fehlerstatus:

- Test-Zeit-Überschreitung ohne verzögertes Sensor-Feedback.
- Test ohne Sensorsignal-Feedback.
- Ungültiges statisches Reset-Signal im Prozess.
- Plausibilitätscheck der Zeiteinstellung der Überwachung.

13.5.22.4 Fehlerverhalten

Im Falle eines Fehlers wird der S_OSSD_Out-Ausgang auf FALSE gesetzt und dieser bleibt im sicheren Zustand. Sobald der Fehler beseitigt wurde und der Sensor auf (S_OSSD_In = TRUE) ist, quittiert ein Reset den Fehlerzustand und setzt den S_OSSD_Out-Ausgang auf TRUE. Wenn S_AutoReset = FALSE, ist ein steigende Flanke am Reset-Eingang erforderlich. Nach dem Übergang der S_OSSD_In auf TRUE kann die optionale Start-Verhinderung durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden. Nach der Blockaktivierung kann die optionale Start-Verhinderung durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C000	Parameter Error	Ungültiger Wert am TestTime-Eingang. Werte zwischen 0 ms und 150 ms sind möglich. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C001	Reset Error 1	Statisches Reset-Signal wurde nach der FB Aktivierung erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C002	Reset Error 2	Statisches Reset-Signal wurde im Zustand 8003 erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C003	Reset Error 3	Statisches Reset-Signal wurde im Zustand C010 erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C004	Reset Error 4	Statisches Reset-Signal wurde im Zustand C020 erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C005	Reset Error 5	Statisches Reset-Signal wurde im Zustand 8006 erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C006	Reset Error 6	Statisches Reset-Signal wurde im Zustand C000 erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C007	Reset Error 7	Statisches Reset-Signal wurde im Zustand 8013 erkannt. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = TRUE Error = TRUE
C010	Test Error 1	Prüfzeit im Zustand 8020 ist abgelaufen. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
C020	Test Error 2	Prüfzeit im Zustand 8030 ist abgelaufen. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8001	Init	FB wurde aktiviert. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8002	ESPE Interrupted 1	Die FB hat eine Sicherheitsnachfrage erkannt. Der Sensor wurde noch nicht automatisch getestet. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8003	Wait for Reset 1	Wartet auf eine steigende Reset-Flanke nach Zustand 8002. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8004	External Function Test	Der automatische Sensor-Test war fehlerhaft. Ein externer manueller Sensor-Test ist notwendig. Die Unterstützung für den notwendigen externen manuellen Sensor-Test wurde am FB (NoExternalTest = FALSE) aktiviert. Eine negative Flanke am Sensor ist erforderlich. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8005	ESPE Interrupted External Test	Der automatische Sensor-Test war fehlerhaft. Ein externer manueller Sensor-Test ist notwendig. Die Unterstützung für den notwendigen externen manuellen Sensor-Test wurde am FB (NoExternalTest = FALSE) aktiviert. Ein TRUE-Signal am Sensor ist erforderlich. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8006	End External Test	Der automatische Sensor-Test war fehlerhaft. Ein externer manueller Sensor-Test ist notwendig. Die Unterstützung für den notwendigen externen manuellen Sensor-Test wurde am FB (NoExternalTest = FALSE) aktiviert. Der externe manuelle Test ist abgeschlossen. Der FB hat einen kompletten Sensor-Schaltzyklus (extern gesteuert). Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8010	ESPE Free No Test	Der FB hat eine Sicherheitsanfrage nicht erkannt. Der Sensor wurde nicht automatisch getestet. Ready = TRUE S_OSSD_Out = TRUE S_TestOut = TRUE TestPossible = TRUE TestExecuted = FALSE Error = FALSE

13. Funktionsblöcke

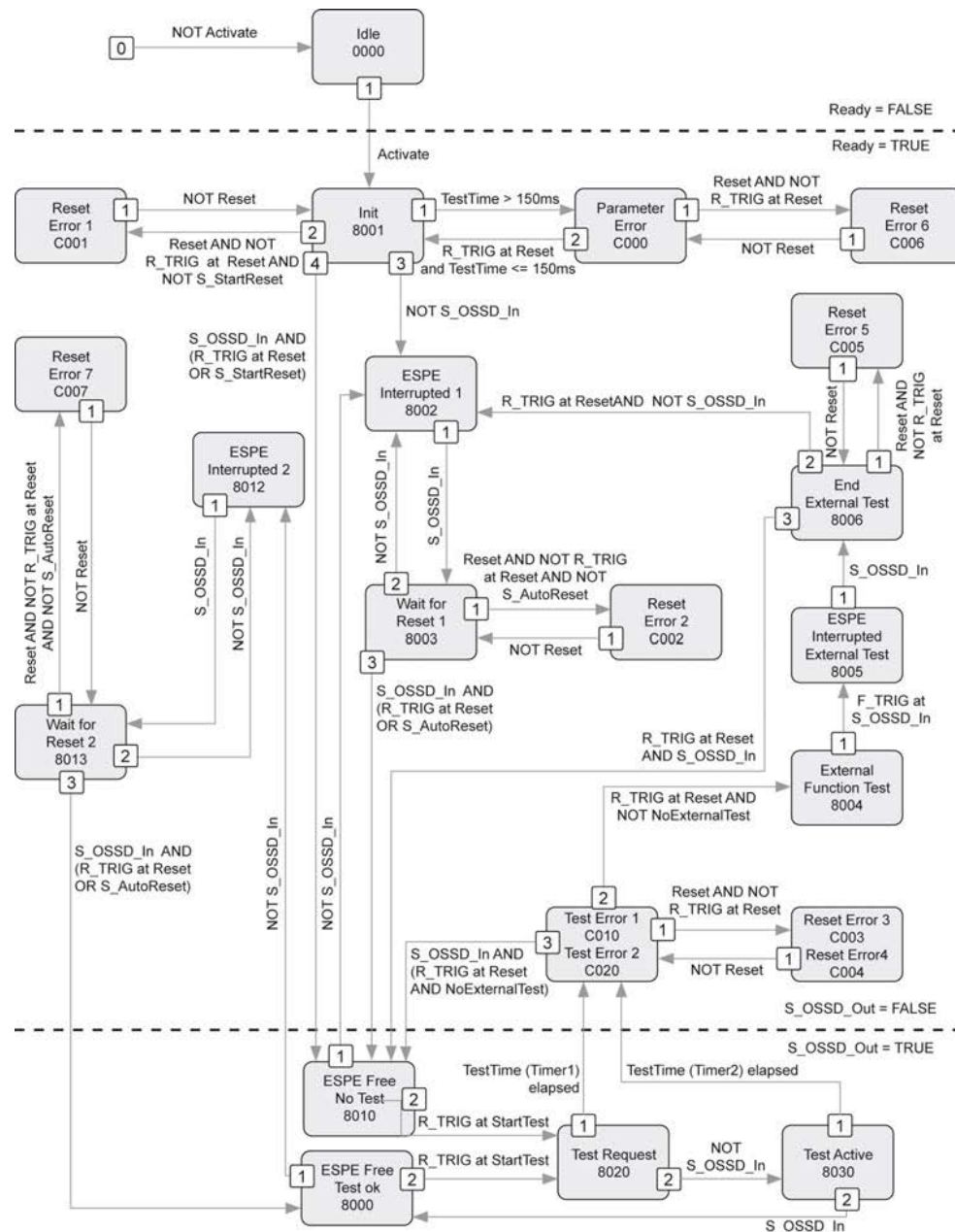
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8020	Test Request	Der automatische Sensor-Test ist aktiv. Der Test-Timer wird zum ersten Mal gestartet. Das Sender Signal des Sensors wird vom FB ausgeschaltet. Das Signal des Empfängers muss dem Signal des Senders folgen. Ready = TRUE S_OSSD_Out = TRUE S_TestOut = FALSE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8030	Test Active	Der automatische Sensor-Test ist aktiv. Test Timer wird zum zweiten Mal gestartet. Der Sender des Sensors wird vom FB eingeschaltet. Das Signal des Empfänger muss dem Signal des Senders folgen. Ready = TRUE S_OSSD_Out = TRUE S_TestOut = TRUE TestPossible = FALSE TestExecuted = FALSE Error = FALSE
8000	ESPE Free Test ok	Der FB hat eine Sicherheitsanfrage nicht erkannt. Der Sensor wurde automatisch getestet. Ready = TRUE S_OSSD_Out = TRUE S_TestOut = TRUE TestPossible = TRUE TestExecuted = TRUE Error = FALSE
8012	ESPE Interrupted 2	Der FB hat eine Sicherheitsnachfrage erkannt. Der Schalter wurde automatisch getestet. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = TRUE Error = FALSE
8013	Wait for Reset 2	Wartet auf eine steigende Flanke am Reset-Eingang nach dem Zustand 8012. Ready = TRUE S_OSSD_Out = FALSE S_TestOut = TRUE TestPossible = FALSE TestExecuted = TRUE Error = FALSE

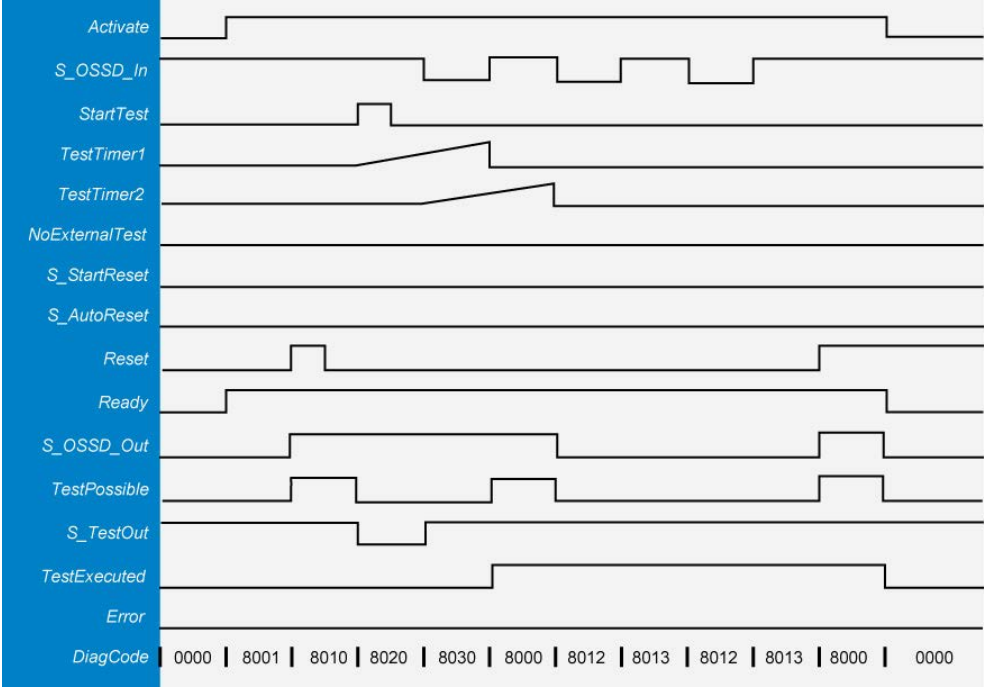
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.22.5 Status-Diagramm SF_TestableSafetySensor



13.5.22.6 Zeit-Diagramm SF_TestableSafetySensor

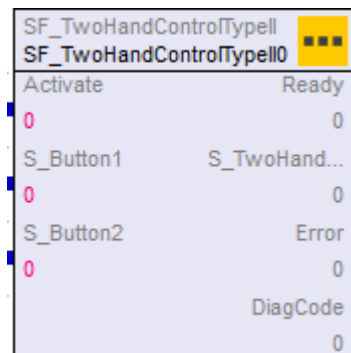


13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.23 SF_TwoHandControlTypII-Funktion

Dieser Funktionsblock bietet eine Zwei-Hand-Control-Funktionalität (siehe EN 574, Abschnitt 4 Typ II).



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_Button1	Eingang 1 (SAFEBOOL)
S_Button2	Eingang 2 (SAFEBOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Funktionsblock ist aktiviert und die Ausgänge sind gültig. (BOOL)
S_TwoHandOut	Sicherheitsrelevantes Ausgangssignal (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Dieser Funktionsbaustein bietet die Zwei-Hand-Control-Funktionalität nach EN 574, Abschnitt 4 Typ II an. Werden S_Button1 und S_Button2 beide auf TRUE gesetzt, wird der Ausgang S_TwoHandOut auch auf TRUE gesetzt. Wird einer der S_Button anschließend auf FALSE gesetzt, wird S_TwoHandOut auch auf FALSE gesetzt. Bevor S_TwoHandOut wieder auf TRUE gesetzt werden kann, müssen vorher beide S_Button1 und S_Button2 auf FALSE sein. Wird einer der S_Button von TRUE auf FALSE gesetzt und gleichzeitig der andere von FALSE auf TRUE wird ein Fehler ausgelöst.

13.5.23.1 Fehlererkennung

Sind bei der Aktivierung des FBs ein oder beide Taster aktiviert, führt dies zu einem Fehlerzustand des FB.

13.5.23.2 Fehlerverhalten

Im Falle eines Fehlers wird der S_TwoHandOut-Ausgang auf FALSE gesetzt und bleibt im sicheren Zustand. Der Fehler-Zustand wird verlassen, wenn beide Taster losgelassen werden (auf FALSE gesetzt).

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Error B1	S_Button1 ist bei FB-Aktivierung auf TRUE gesetzt. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C002	Error B2	S_Button2 ist bei FB-Aktivierung auf TRUE gesetzt. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C003	Error B1&B2	Die Signale von S_Button1 und S_Button2 sind bei FB-Aktivierung auf TRUE gesetzt. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE Error = FALSE S_TwoHandOut = FALSE
8000	Buttons Actuated	Beide Taster wurden richtig betätigt. Der sicherheitsrelevante Ausgang wird aktiviert. Ready = TRUE Error = FALSE S_TwoHandOut = TRUE
8001	Init	Funktionsblock ist aktiv, jedoch im Init-Zustand. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8004	Buttons Released	Kein Taster wird betätigt. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8005	Button 1 Actuated	Taster 1 ist gedrückt. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8006	Button 2 Actuated	Taster 2 ist gedrückt. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8007	Button 2 Released	Der sicherheitsrelevante Ausgang wurde aktiviert und ist wieder deaktiviert. In diesem Zustand ist S_Button1 TRUE und S_Button2 FALSE. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE

13. Funktionsblöcke

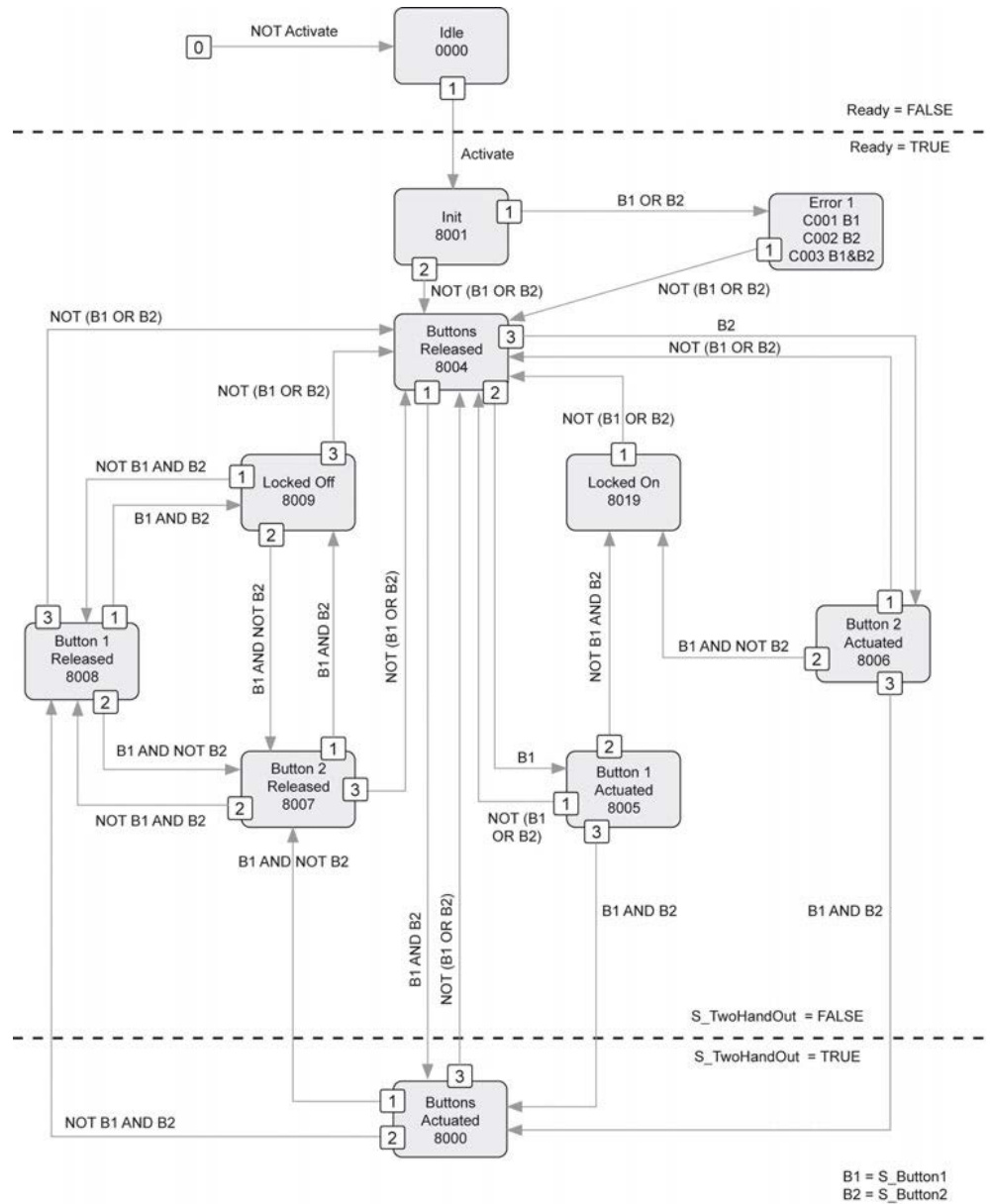
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8008	Button 1 Released	Der sicherheitsrelevante Ausgang wurde aktiviert und ist wieder deaktiviert. In diesem Zustand ist S_Button1 FALSE und S_Button2 TRUE. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8009	Locked Off	Der sicherheitsrelevante Ausgang wurde aktiviert und ist wieder deaktiviert. In diesem Zustand ist S_Button1 TRUE und S_Button2 wird nach dem Deaktivieren des sicherheitsrelevanten Ausgang wieder TRUE. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8019	Locked On	Falsche Betätigung der Tasten. Warten, dass die beiden Tasten losgelassen werden. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

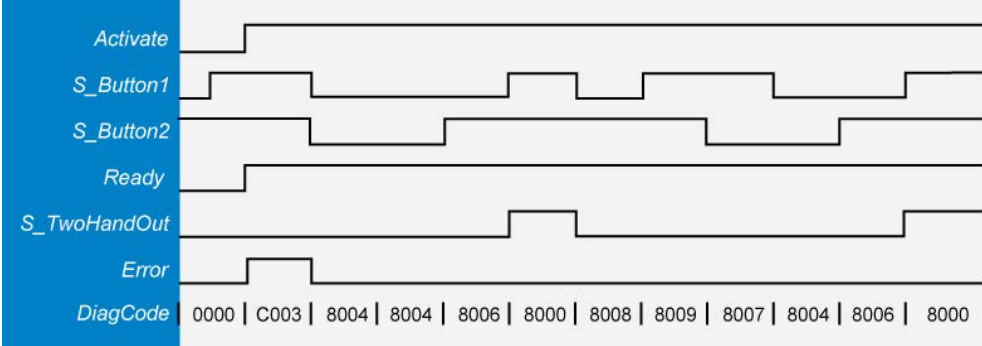
13.5.23.3 Status-Diagramm SF_TwoHandControlTypell



13. Funktionsblöcke

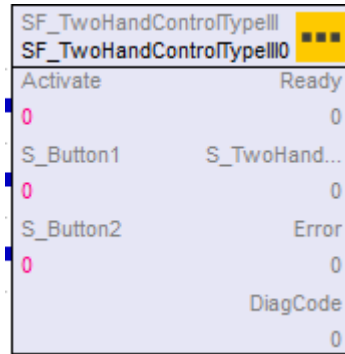
13.5 Komplexe Funktionsblöcke

13.5.23.4 Zeit-Diagramm SF_TwoHandControlTypell



13.5.24 SF_TwoHandControlTypeIII-Funktion

Dieser Funktionsblock bietet eine Zwei-Hand-Control-Funktionalität (siehe EN 574, Abschnitt 4 Typ III). Eine fix definierte Zeitdifferenz von 500 ms ist vorgegeben.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
S_Button1	Eingang 1 (SAFEBOOL)
S_Button2	Eingang 2 (SAFEBOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Der Funktionsblock ist aktiviert und die Ausgänge sind gültig. (BOOL)
S_TwoHandOut	Sicherheitsrelevantes Ausgangssignal (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Dieser Funktionsbaustein bietet die Zwei-Hand-Control-Funktionalität nach EN 574, Abschnitt 4 Typ II an. Werden S_Button1 und S_Button2 innerhalb von 500 ms auf TRUE gesetzt, wird der Ausgang S_TwoHandOut auch auf TRUE gesetzt. Wird einer der S_Button anschließend auf FALSE gesetzt, wird S_TwoHandOut auch auf FALSE gesetzt. Bevor S_TwoHandOut wieder auf TRUE gesetzt werden kann, müssen vorher beide S_Button1 und S_Button2 auf FALSE sein. Wird einer der S_Button von TRUE auf FALSE gesetzt und gleichzeitig der andere von FALSE auf TRUE wird ein Fehler ausgelöst.

13.5.24.1 Fehlererkennung

Sind bei Aktivierung des FBs ein oder beide Taster aktiviert, führt dies zu einem Fehlerzustand des FBs. Der FB erkennt, wenn die Abweichung der Eingangssignale 500 ms überschreitet.

13.5.24.2 Fehlerverhalten

Im Falle eines Fehlers wird der S_TwoHandOut-Ausgang auf FALSE gesetzt und bleibt im sicheren Zustand. Der Fehler-Zustand wird verlassen, wenn beide Taster losgelassen (auf FALSE gesetzt) werden.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Error 1 B1	S_Button1 ist bei FB-Aktivierung auf TRUE gesetzt. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C002	Error 1 B2	S_Button2 ist bei der FB-Aktivierung auf TRUE gesetzt. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C003	Error 1 B1&B2	Die Signale S_Button1 und S_Button2 sind bei FB-Aktivierung auf TRUE. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C004	Error 2 B1	S_Button1 = FALSE, S_Button 2 = TRUE und 500 ms abgelaufen. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C005	Error 2 B2	S_Button1 = TRUE, S_Button 2 = FALSE und 500 ms abgelaufen. Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
C006	Error 2 B1&B2	S_Button1 = TRUE, S_Button 2 = TRUE und 500 ms abgelaufen Ready = TRUE Error = TRUE S_TwoHandOut = FALSE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE Error = FALSE S_TwoHandOut = FALSE
8000	Buttons Actuated	Beide Tasten wurden richtig betätigt. Der sicherheitsrelevante Ausgang wird aktiviert. Ready = TRUE Error = FALSE S_TwoHandOut = TRUE
8001	Init	Funktionsblock ist aktiv, jedoch im Init-Zustand. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8004	Buttons Released	Kein Taste betätigt. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8005	Button 1 Actuated	Taster 1 ist betätigt. Überwachungszeit starten. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8006	Button 2 Actuated	Taster ist 2 betätigt. Überwachungszeit starten. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8007	Button 2 Released	Der sicherheitsrelevante Ausgang wurde aktiviert und ist wieder deaktiviert. In diesem Zustand ist S_Button1 TRUE und S_Button2 FALSE. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8008	Button 1 Released	Der sicherheitsrelevante Ausgang wurde aktiviert und ist wieder deaktiviert. In diesem Zustand ist S_Button1 FALSE und S_Button2 TRUE. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8009	Locked Off	Der sicherheitsrelevanten Ausgang wurde aktiviert und ist wieder deaktiviert. In diesem Zustand ist S_Button1 TRUE und S_Button2 wird nach dem Deaktivieren des sicherheitsrelevanten Ausgang wieder TRUE. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE
8019	Locked On	Falsche Betätigung der Taster. Warten, dass die beiden Taster losgelassen werden. Ready = TRUE Error = FALSE S_TwoHandOut = FALSE

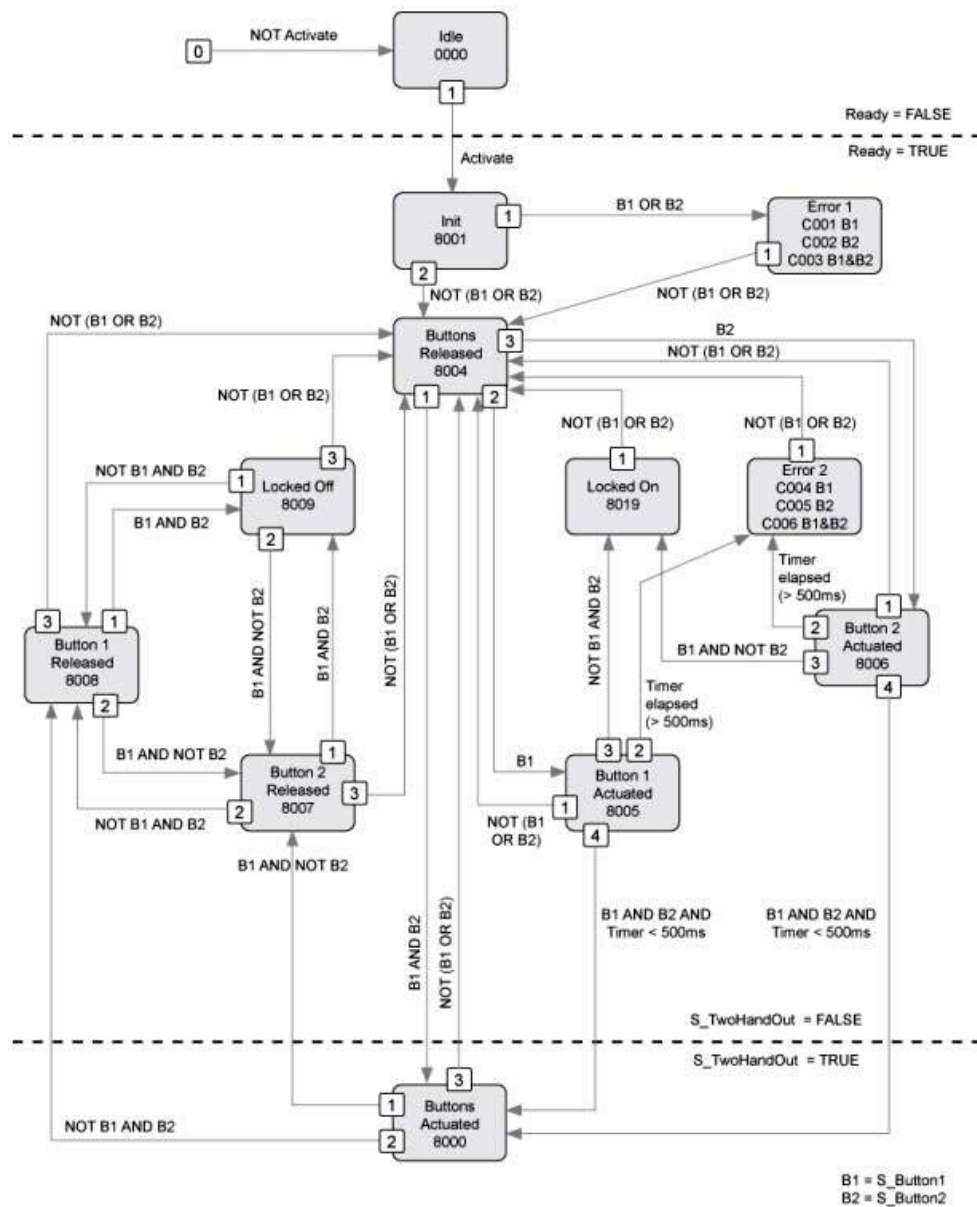
13.5.24.3 Besonderheiten

Im Status 8005 und 8006 wird als erstes geprüft, ob beide S_Button-Eingänge SAFE_FALSE sind, ist dies der Fall, wird im Status 8004 verzweigt. Der Grund dafür ist, dass kein Error State definiert ist, wenn die Überwachungszeit abgelaufen ist und beide S_Button-Eingänge SAFE_FALSE sind.

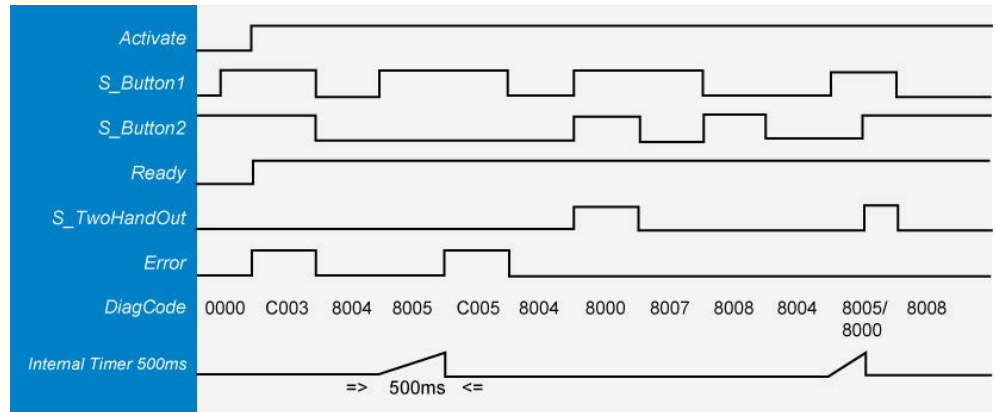
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.24.4 Status Diagramm SF_TwoHandControlTypIII



13.5.24.5 Zeit-Diagramm SF_TwoHandControlTypIII



13. Funktionsblöcke

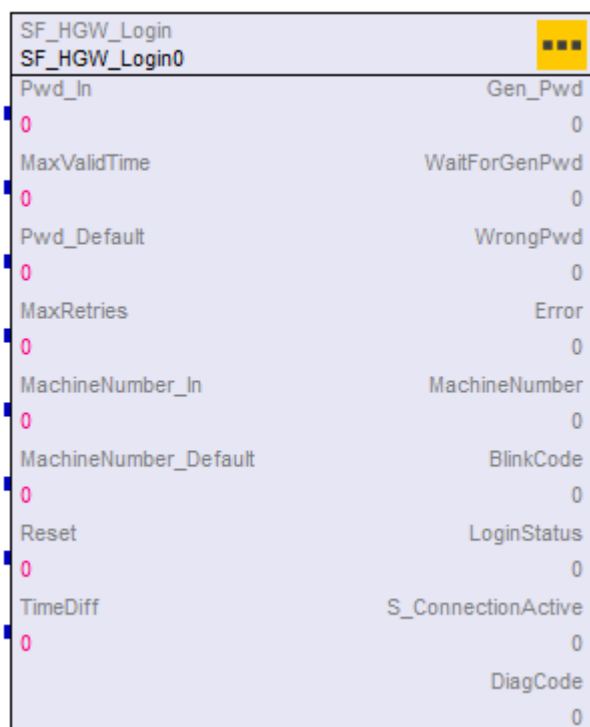
13.5 Komplexe Funktionsblöcke

13.5.25 SF_HGW_Login

Dieser Funktionsblock dient zur Anbindung eines HGW-Moduls an eine Safety-CPU. Der Funktionsblock kann nur zusammen mit einem HGW-Modul eingesetzt werden. Der Funktionsblock beinhaltet zwei getrennte Mechanismen zur Anbindung des Moduls.

Im ersten Bereich kann eine eindeutige Maschinennummer für das System vergeben werden. Die Maschinennummer kann entweder über eine Konstante gesetzt werden oder über einen passwortgeschützten Anmeldemechanismus aus der Visualisierung eingestellt werden.

Der zweite Bereich dient zum Aufbau einer sicheren Verbindung zwischen einem HGW-Modul und einer Safety-CPU.



Eingangsvariablen

Pwd_In	Eingegebenes Kennwort (DINT), eine Änderung löst den Umschaltvorgang zum Ändern der Maschinennummer aus.
MaxValidTime	Maximale Zeitdifferenz (DINT), um den Anmeldevorgang zum Ändern der Maschinennummer komplett abzuschließen (von 8001 nach 8004). 0 bis maximal 8 Stunden. 0 bedeutet die Zeitprüfung ist deaktiviert. Der Wert muss durch eine Konstante beschrieben werden.
Pwd_Default	Vorgegebenes Kennwort (DINT) zum Starten des Anmeldevorgangs zum Ändern der Maschinennummer. Der Wert muss durch eine Konstante beschrieben werden.
MaxRetries	Maximale Anzahl (DINT) von Fehlversuchen für das Bestätigungspasswort.
MachineNumber_In	Eingegebene Maschinennummer (DINT), der Wert wird verwendet um die Maschi-

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

	nennnummer zu ändern. Voraussetzung dafür ist der State 8004
MachineNumber_Default	Defaultwert für die Maschinennummer 1 bis maximal 99 Der Wert muss durch eine Konstante beschrieben werden.
Reset	Zurücksetzen (BOOL) des Umschaltvorgangs der Maschinennummer
Time_Diff	Maximale Zeit (DINT), um den Anmeldevorgang zwischen HGW-Modul und Safety-CPU komplett abzuschließen (von 8001 nach 8012). > 0 ms bis maximal 8 Stunden. Der Wert muss durch eine Konstante beschrieben werden.
Ausgangsvariablen	
Gen_Pwd	Generiertes Bestätigungspasswort (DINT).
WaitForGenPwd	Angabe (BOOL), ob auf die Eingabe eines generierten Passworts von Gen_Pwd gewartet wird bzw. ob auf die Eingabe des Blinkcodes bei der Anmeldung des HGW-Moduls gewartet wird. FALSE: Es liegt kein gültiges Bestätigungspasswort oder kein gültiger Blinkcode an. TRUE: Das angegebene Bestätigungspasswort oder der Blinkcode kann eingegeben werden.
Wrong_Pwd	Information (BOOL), ob am Eingang Pwd_In ein erwartetes Kennwort anliegt. FALSE: Default Zustand TRUE: Es wurde ein falsches Passwort eingegeben.
Error	Fehleranzeige (BOOL)
MachineNumber	(remanent) Eingestellte Maschinennummer (DINT)
BlinkCode	Generierter Blinkcode (DINT), welcher für die Bestätigung der Anmeldung des HGW-Moduls erforderlich ist.
LoginState	Login Status (DINT) zeigt den aktuellen Login Status des HGW-Moduls bei der Safety-CPU an. 0 = nicht angemeldet, 1 = warte auf Blinkcode Bestätigung, 2 = angemeldet, noch keine FSoE Verbindung aufrecht, 3 = Error, 4 = FSoE Verbindung läuft, 5 = Defaultanmeldung erfolgreich
S_Connec-tionActive	Statusflag (SAFEBOOL) zeigt an, ob eine FSoE Verbindung hergestellt ist
DiagCode	Statusanzeige (HDINT)

Der Funktionsblock ist nötig um der Safety-CPU eine eindeutige Maschinennummer zuzuweisen. Der Anwender ist für die Eindeutigkeit der Maschinennummer im System verantwortlich. Im HGW-Modul kann über diese Maschinennummer ausgewählt werden, zu welcher Safety-CPU eine Verbindung aufgebaut werden soll.



Die Maschinennummer muss im System eindeutig vergeben werden.

Beim ersten Hochfahren der Safety-CPU wird der Eingang MachineNumber_Default am remanenten Ausgang MachineNumber verspeichert. Die verspeicherte Maschinennummer kann zur Laufzeit verändert werden. Dazu muss der Maschineneinsteller über den Anmeldemechanismus der Maschinennummer die Änderung vornehmen. Die Anmeldung wird mit der Eingabe des vorgegebenen Passworts gestartet. Der Funktionsblock generiert am Ausgang Gen_Pwd ein Passwort, welches vom Anwender bestätigt werden muss. Nach erfolgreicher Bestätigung kann über den Eingang MachineNumber_In die Maschinennummer verändert werden.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

➔ Die Maschinenummer 0 am Eingang MachineNumber_In wird vom Funktionsblock nicht übernommen, da es ein ungültiger Wert ist.

Wird eine Maschinenummer vom HGW-Modul zur Safety-CPU gesendet und vom Funktionsblock als richtig erkannt, wird der Aufbau einer FSoE-Verbindung zwischen HGW-Modul und Safety-CPU vorbereitet. Der Funktionsblock generiert einen Blinkcode an dem vorgesehenen Ausgang. Der Blinkcode muss am HGW-Modul bestätigt werden. Sobald der Blinkcode bestätigt ist, wird eine FSoE Verbindung zwischen HGW-Modul und Safety-CPU hergestellt. Der gültige Aufbau einer FSoE-Verbindung, wird am Ausgang ConnectionActive angezeigt.

➔ Die Eingänge des HGW Moduls werden nur bei einer gültigen FSoE-Verbindung zur Safety-CPU übertragen. Die Eingänge des HGW-Moduls müssen mittels eines SF_Optional_Switch Funktionsblocks verschalten werden. Die Umschaltung dieses Funktionsblocks muss über den Ausgang ConnectionActive des SF_HGW_Login Funktionsblocks erfolgen.

13.5.25.1 Fehlererkennung

Der Funktionsblock erkennt folgende Fehlerzustände:

- Eingabe des falschen Kennwortes zur Bestätigung des Umschaltvorgangs
- Zeitüberschreitung bei Eingabe des Bestätigungskennworts
- Eingabe einer falschen Maschinenummer bei der Änderung der Nummer
- Anmeldung des HGW-Moduls mit einer falschen Maschinenummer
- Eingabe eines falschen Blinkcodes
- Zeitüberschreitung bei der Anmeldung des HGW-Moduls
- Abbruch der FSoE-Verbindung

Wird im Zustand 8000 oder 8010 ein falsches Kennwort eingeben, löst dies keinen Fehler aus, sondern wird nur durch ein TRUE auf den Ausgang WrongPwd signalisiert.

13.5.25.2 Fehlerverhalten

Der Ausgang DiagCode zeigt den entsprechenden Fehler-Code an und der Error-Ausgang wird auf TRUE gesetzt. Um den Fehlerzustand zu verlassen, muss dieser mit einer steigenden Flanke am Reset-Eingang quittiert werden. Dabei sollte der Eingang Pwd_In auf „0“ gesetzt sein.

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C001	MaxValidTime_exceed	Es wurde ein Zeitüberlauf von MaxValidTime erkannt. Kann aus State 8002 & 8003 folgen. WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
C002	Too_many_retries	Das generierte Passwort wurde zu oft falsch eingegeben. Kann aus State 8002 & 8003 folgen. WaitForGenPwd = FALSE WrongPwd = TRUE Error = TRUE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
C003	Invalid_MachineNumber_entered	Es wurde eine ungültige Maschinenummer an MachineNumber_In angelegt. Kann aus State 8004 folgen. WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
C010	Wrong_MachNum_for_login	Die Maschinenummer für den Login und die gesetzte Maschinenummer stimmen nicht überein. Kann aus State 8001 folgen. WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE LoginState = 3 ConnectionActive = SAFEBOOL_FALSE
C011	TimeDiff_exceed	Es wurde ein Zeitüberlauf von TimeDiff erkannt. Kann aus State 8010 & 8011 folgen. WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE LoginState = 3 ConnectionActive = SAFEBOOL_FALSE
C012	Wrong_blinkcode_entered	Es wurde ein falscher Blinkcode eingegeben. Kann aus State 8010 folgen. WaitForGenPwd = FALSE WrongPwd = TRUE Error = TRUE LoginState = 3 ConnectionActive = SAFEBOOL_FALSE
C013	FSoE_connection_broken	Die FSoE-Verbindung wurde unterbrochen. Kann aus State 8012 folgen. WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE LoginState = 3 ConnectionActive = SAFEBOOL_FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C020	Wrong_MachineNumber_Detected	Es wurde eine ungültige Maschinennummer am Ausgang erkannt. WaitForGenPwd = FALSE WrongPwd = FALSE Error = TRUE LoginState = 3 ConnectionActive = SAFEBOOL_FALSE
FB-spezifische Status-Codes (kein Fehler):		
0000	Init	Der Funktionsblock wurde aktiviert (Ausgangszustand). WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
8000	Init	Der Funktionsblock wurde aktiviert. Überprüfen ob eine Maschinennummer gesetzt ist. Wenn nicht, wird in diesem State der Eingang MachineNumber_Default auf den Ausgang MachineNumber gesetzt. WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
8001	Wait_for_Operation	Prüfen, ob ein Passwort gleich dem Pwd_Default an Pwd_In anliegt, wenn Reset auf FALSE steht. Prüfen ob eine FSoE Verbindung aufgebaut werden soll. WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
8002	Login_Request_Default	Es wurde das Default Passwort an Pwd_In angelegt. Der FUB generiert ein Bestätigungs-Passwort welchen an Gen_Pwd angelegt wird. Der Anwender muss zum Bestätigen des Anmeldevorgangs dieses Passwort an Pwd_In anlegen. WaitForGenPwd = TRUE WrongPwd = FALSE Error = FALSE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE
8003	Login_Request_Default_Retry	Das Bestätigungspasswort wurde falsch eingegeben. Der Anwender hat die Möglichkeit das Passwort so oft wie an Max_Retry angelegten Wert zu wiederholen. WaitForGenPwd = TRUE WrongPwd = TRUE Error = FALSE LoginState = 0 ConnectionActive = SAFEBOOL_FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8004	Logged_In_Default	Das generierte Passwort wurde bestätigt. Der Anwender hat nun die Möglichkeit über MachineNumber_In den Ausgang MachineNumber zu setzen. WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE LoginState = 5 ConnectionActive = SAFEBOOL_FALSE
8010	Login_Request_Machine	Es wurde die eingegebene Maschinenummer am HGW erkannt und der Vergleich dem Ausgang MachineNumber war erfolgreich. Es wird ein Blinkcode generiert, welcher vom Anwender bestätigt werden muss WaitForGenPwd = TRUE WrongPwd = FALSE Error = FALSE LoginState = 1 ConnectionActive = SAFEBOOL_FALSE
8011	Login_Request_Blinkcode_ACK	Der Blinkcode wurde vom Anwender bestätigt. Es wurde erfolgreich angemeldet. Das HGW baut eine FSoE-Verbindung zur Safety-CPU auf. WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE LoginState = 2 ConnectionActive = SAFEBOOL_FALSE
8012	Logged_In	Die FSoE Verbindung wurde erfolgreich aufgebaut. Die sicheren Daten werden nun von HGW zu Safety-CPU ausgetauscht. WaitForGenPwd = FALSE WrongPwd = FALSE Error = FALSE LoginState = 4 ConnectionActive = SAFEBOOL_TRUE

Die Zustände 8002, 8003 und 8004 lassen sich mit dem Reset auf TRUE abbrechen. Der Funktionsblock wechselt in den State 8001.

Die Fehlerzustände C001, C002 und C003 lassen sich mit dem Reset auf TRUE und dem Pwd_In auf 0 quittieren. Der Funktionsblock wechselt in den State 8001.

Die Zustände 8010, 8011 und 8012, sowie die Fehlerzustände C010, C011, C012 und C013 lassen sich über den Logout_Request des HGW Modul abbrechen. Der Funktionsblock wechselt in den State 8001.

Der Fehlerzustand C020 kann aus jedem Zustand ausgenommen der Initzustände generiert werden. Der Fehler lässt sich mit dem Reset und Pwd_In = 0 quittieren.

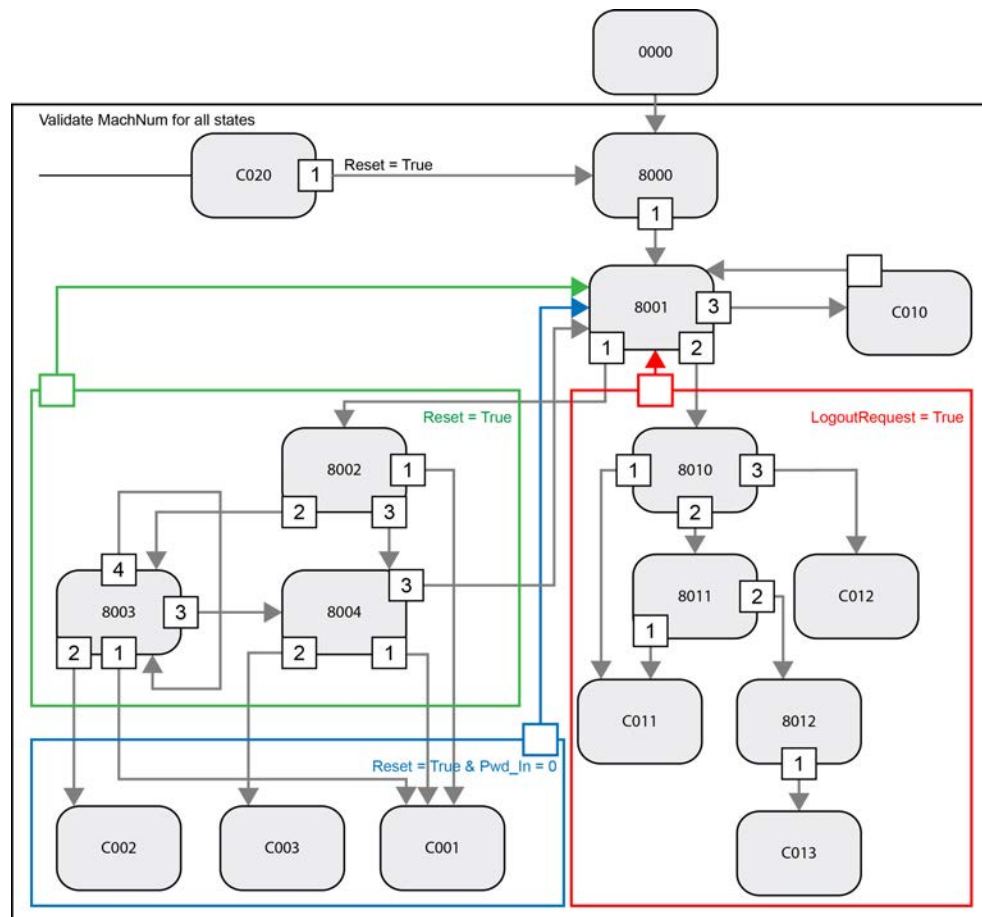


Das Quittieren des Errors zieht eine Maschinenummer-Änderung nach sich, welche auf den Wert von MachineNumber_Default gesetzt wird!

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.25.3 SF_HGW_Login Status Diagram



13.5.26 Komplexe numerische Funktionsblöcke

- "SF_LimitComparator", Seite 341
- "SF_Numeric_Selector", Seite 346
- "SF_RampMonitor", Seite 350
- "SF_DirectionMonitor", Seite 354
- "SF_SkewMonitor", Seite 358

Siehe auch

- "Numerische Standard-Funktionsblöcke", Seite 172

13.5.26.1 SF_LimitComparator

Überprüft das Eingangssignal auf oberes und unteres Limit. Das obere Limit muss größer als das untere Limit sein.

Die Überwachungsfunktion des Funktionsblocks kann mittels S_MonitoringOn aktiviert oder deaktiviert werden. Wenn S_MonitoringOn FALSE ist, wird der Ausgang S_

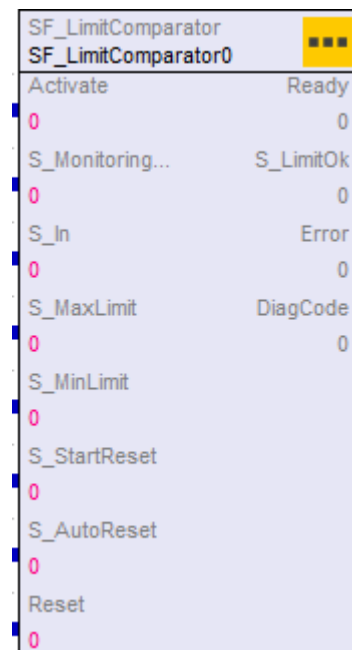
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

LimitOk auf TRUE gesetzt. S_LimitOk wird FALSE, wenn S_MonitoringOn TRUE und S_In höher als S_MaxLimit oder niedriger als S_MinLimit ist.

Nach einer Grenzwertverletzung kann nur dann wieder in den Überwachungszustand geschaltet werden, wenn ein Reset ausgeführt wird (Bedingung: Eingang S_In innerhalb der Grenzwerte). Das Reset-Verhalten hängt von den Eingängen S_AutoReset und Reset ab.

Ist S_AutoReset TRUE, dann erfolgt die Quittierung automatisch. Wenn S_AutoReset FALSE ist, muss eine steigende Flanke am Reset-Eingang anliegen, um die Aktivierung zu quittieren.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock. (BOOL)
S_MonitoringOn	Schaltet die Überwachung ein. FALSE: Keine Überwachung. Es werden keine Regelverletzungen ausgewertet. TRUE: Überwachungsfunktion ist eingeschaltet. (SAFEBOOL)
S_In	Eingangssignal das überwacht werden soll. (SAFEDINT)
S_MaxLimit	Oberer Grenzwert (SAFEDINT)
S_MinLimit	Unterer Grenzwert (SAFEDINT)
S_StartReset	Bestimmt Einschaltverhalten: FALSE: Manueller Reset TRUE: Automatischer Reset nachdem der FB gestartet wird. (SAFEBOOL)
S_AutoReset	Bestimmt Quittierungsverhalten: FALSE: Manueller Reset TRUE: Automatischer Reset (SAFEBOOL)

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind. (BOOL)
S_LimitOk	Ob bei eingeschalteter Überwachung der Eingangswert innerhalb der Grenzwerte ist. FALSE: Eingangswert liegt außerhalb der Grenzwerte. TRUE: Eingang innerhalb der Grenzwerte oder Überwachung ist ausgeschaltet (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

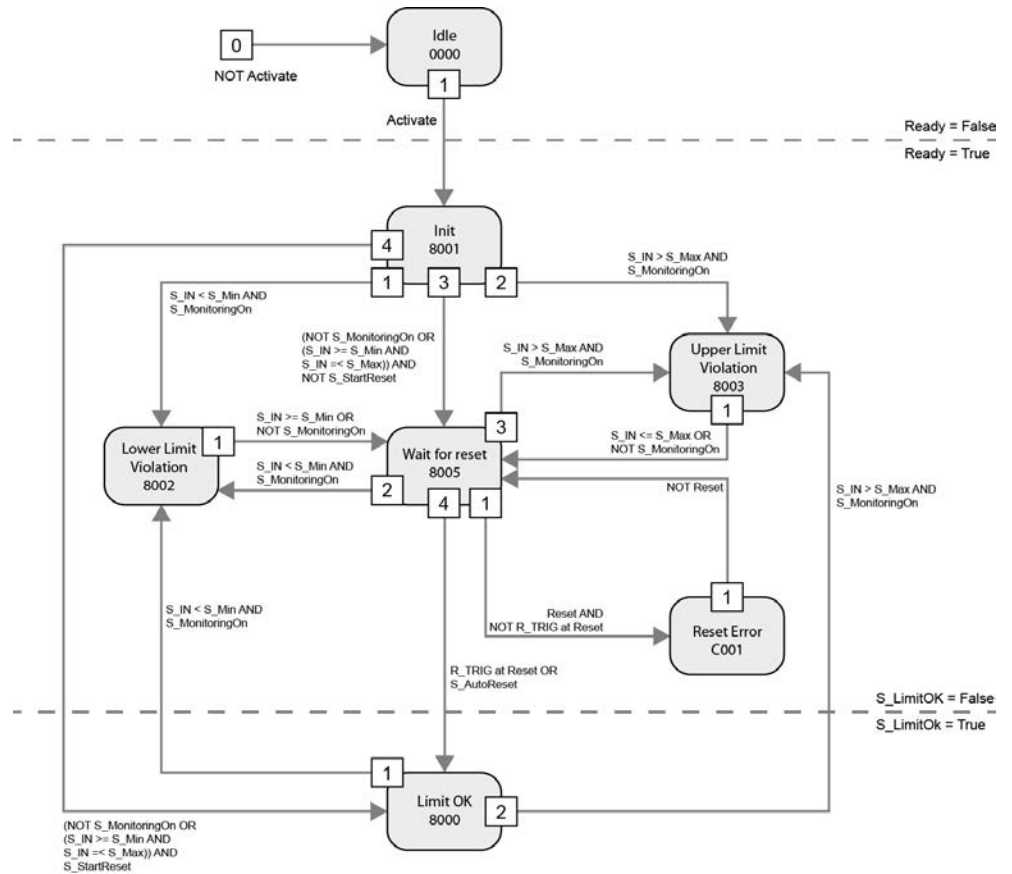
Fehlerverhalten

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal im Status 8005 Ready = TRUE S_LimitOk = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_LimitOk = FALSE Error = FALSE
8001	Init	Eine Aktivierung wurde vom FB erkannt und der Status ist nun aktiviert. Ready = TRUE S_LimitOk = FALSE Error = FALSE
8000	Limit Ok	Überwachungsstatus. Bei S_MonitoringOn TRUE wird der Eingang auf beide Grenzwerte hin überprüft. Ist S_MonitoringOn FALSE, dann findet keine Grenzwertüberprüfung statt. Ready = TRUE S_LimitOk = TRUE Error = FALSE
8002	Lower Limit Violation	Der Eingangswert hat den unteren Grenzwert unterschritten. Ready = TRUE S_LimitOk = FALSE Error = FALSE
8003	Upper Limit Violation	Der Eingangswert hat den oberen Grenzwert überschritten. Ready = TRUE S_LimitOk = FALSE Error = FALSE
8005	Wait for Reset	Die Aktivierung ist TRUE. Eingangswert innerhalb der beiden Grenzwerte. Prüfen, ob S_AutoReset TRUE ist oder auf eine steigende Flanke von Reset warten. Ready = TRUE S_LimitOk = FALSE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Status Diagramm SF_LimitComparator



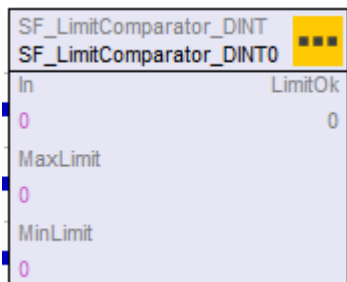
13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

13.5.26.2 SF_LimitComparator_Dint

Überprüft das Eingangssignal auf oberes und unteres Limit.

Ist der Eingang „In“ > „MaxLimit“ oder „In“ < „MinLimit“ wird der Ausgang „LimitOk“ = FALSE gesetzt.



Eingangsvariablen	
In	Eingang (DINT)
MaxLimit	Obere Grenze für den Eingangswert
MinLimit	Untere Grenze für den Eingangswert
Ausgangsvariablen	
LimitOk	TRUE, wenn Eingangswert innerhalb der vorgegebenen Grenzen liegt (BOOL)

13.5.26.3 SF_Numeric_Selector

Wie→ " SF_ModeSelector-Funktion", Seite 264, jedoch mit SafeDint-Werten. Mit den S_ModeX SafeBool-Eingängen kann ein SafeDint-Eingang ausgewählt werden, der auf dem Ausgang S_Out geschaltet wird.

Im Fehlerfall (kein oder mehrere Eingänge selektiert) wird der Ausgang S_Out immer auf 0 gesetzt. Hier muss zusätzlich der Ausgang S_AnyModeSel in die Benutzer-Logik einfließen, um zu ermitteln, ob S_Out gültig ist oder nicht.

SF_NumericSelector	
SF_NumericSelector0	
Activate	Ready
0	0
S_Mode0	S_Out
0	0
S_Mode1	S_AnyModeSel
0	0
S_Mode2	Error
0	0
S_Mode3	DiagCode
0	0
S_Mode4	
0	
S_NumIn0	
0	
S_NumIn1	
0	
S_NumIn2	
0	
S_NumIn3	
0	
S_NumIn4	
0	
S_Unlock	
0	
S_SetMode	
0	
AutoSetMode	
0	
ModeMonitor...	
0	
Reset	
0	

Eingangsvariablen

Activate	Aktiviert den Funktionsblock (BOOL)
----------	-------------------------------------

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

S_Mode0	Gilt für alle: Eingang X vom Betriebsarten-Schalter FALSE: Mode X nicht ausgewählt. TRUE: Mode X vom Operator ausgewählt. (SAFEBOOL)
S_Mode1	
S_Mode2	
S_Mode3	
S_Mode4	
S_NumIn0	Gilt für alle: Wenn S_NumInX aktiv ist, wird der Wert auf S_Out gesetzt. (SAFEDINT)
S_NumIn1	
S_NumIn2	
S_NumIn3	
S_NumIn4	
S_Unlock	FALSE: Sperrt die Betriebsart. Eine Änderung an den S_ModeX-Eingängen führt zu keiner Änderung an den S_ModeXSel-Ausgängen. (SAFEBOOL)
S_SetMode	TRUE: Quittierung zum Setzen einer neuen Betriebsart bei gültigen S_ModeX-Eingängen. (SAFEBOOL)
AutoSetMode	TRUE: Automatische Quittierung zum Setzen einer neuen Betriebsart bei gültigen S_ModeX-Eingängen. (BOOL)
ModeMonitorTime	Maximale Überwachungszeit, in der alle S_ModeX-Eingänge FALSE sind. (DINT)
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind. (BOOL)
S_Out	S_NumInX-Wert des gewählten Mode. (SAFEDINT)
S_AnyModeSel	TRUE: Es ist eine Betriebsart ausgewählt. (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

Fehlerverhalten

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Error Short-circuit	Der FB hat festgestellt, dass zwei oder mehr S_ModeX auf TRUE sind Ready = TRUE Error = TRUE S_AnyModeSel = FALSE S_Out = 0
C002	Error Open-circuit	Die FB hat festgestellt, dass alle S_ModeX auf FALSE sind: Die Zeit nach einer fallenden S_ModeX Flanke überschreitet die ModeMonitorTime. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE S_Out = 0
C003	Reset Error 1	Statisches Reset-Signal wurde im Zustand C001 erkannt. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE S_Out = 0

13. Funktionsblöcke

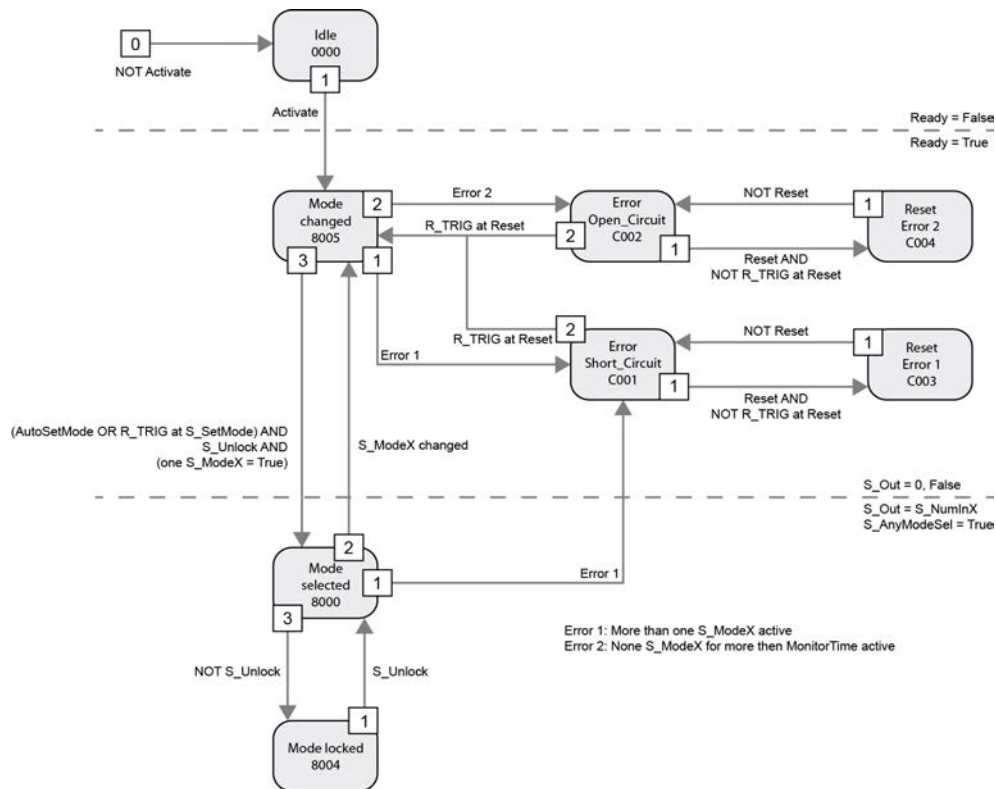
13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
C004	Reset Error 2	Statisches Reset-Signal wurde im Zustand C002 erkannt. Ready = TRUE Error = TRUE S_AnyModeSel = FALSE S_Out = 0
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE Error = FALSE S_AnyModeSel = FALSE S_Out = 0
8005	ModeChanged	Ready = TRUE Error = FALSE S_AnyModeSel = FALSE S_Out = 0
8000	ModeSelected	Gültige Modusauswahl noch nicht gesperrt. Ready = TRUE Error = FALSE S_AnyModeSel = TRUE S_Out = Selected Input Signal.
8004	ModeLocked	Gültige Modusauswahl ist gesperrt. Ready = TRUE Error = FALSE S_AnyModeSel = TRUE S_Out = Selected Input Signal.

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Status Diagramm SF_NumericSelector



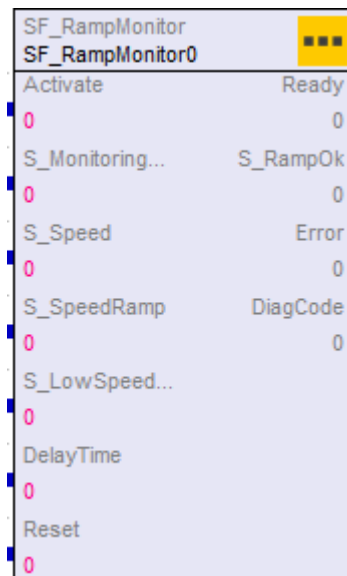
13.5.26.4 SF_RampMonitor

RampMonitor überwacht die Geschwindigkeitsreduktion im Falle eines NOT-Halts. Eine steigende Flanke bei S_MonitoringOn bewirkt das Übernehmen der aktuellen Geschwindigkeit als Ausgangsgeschwindigkeit für die Überwachungsfunktion.

Wenn S_MonitoringOn TRUE und die Verzögerungszeit abgelaufen ist, muss sich die Geschwindigkeit um den Wert S_SpeedRamp reduzieren (Geschwindigkeit muss sich unterhalb der berechneten Rampe befinden). Die Verzögerungszeit wird für die mechanischen Teile zum Auslösen des Bremsvorgangs benötigt.

Überwachung ist aktiviert wenn S_MonitoringOn TRUE ist und wird nicht beendet wenn die Geschwindigkeit unterhalb von S_LowSpeedThreshold liegt. S_LowSpeedThreshold wird als unterer Wert für die berechnete Rampe verwendet. Geschwindigkeiten unterhalb von S_LowSpeedThreshold bedeutet, dass die Maschine gestoppt hat.

Fehler können durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden.



Eingangsvariablen	
Activate	Aktiviert den Funktionsblock. (BOOL)
S_MonitoringOn	Aktivierung der Überwachung des S_Speed Eingangs. FALSE: Keine Überwachung. Es werden keine Regelverletzungen ausgewertet. TRUE: Überwachungsfunktion ist eingeschaltet. (SAFEBOOL)
S_Speed	Aktuelle Geschwindigkeit aus dem XS-322-2INC-Modul (SAFEDINT)
S_SpeedRamp	Faktor zur Berechnung des Kurvenverlaufs der berechneten Geschwindigkeit (Geschwindigkeit/[ms]). (SAFEDINT)
S_LowSpeedThreshold	Beschränkung der berechneten Geschwindigkeit. (SAFEDINT)
DelayTime	Verzögerungszeit bis zum Start der Überwachung nachdem S_MonitoringOn = TRUE. (DINT)

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind. (BOOL)
S_RampOk	Ergebnis der Überwachungsfunktion: FALSE: wenn S_MonitoringOn TRUE und S_Speed höher als der errechnete Wert vx ist. TRUE: wenn die Geschwindigkeit unterhalb des errechneten Wertes oder S_MonitoringOn FALSE ist (Überwachung deaktiviert). S_RampOk wird ungültig, wenn einer der Safety-Eingänge nicht gültig ist. (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

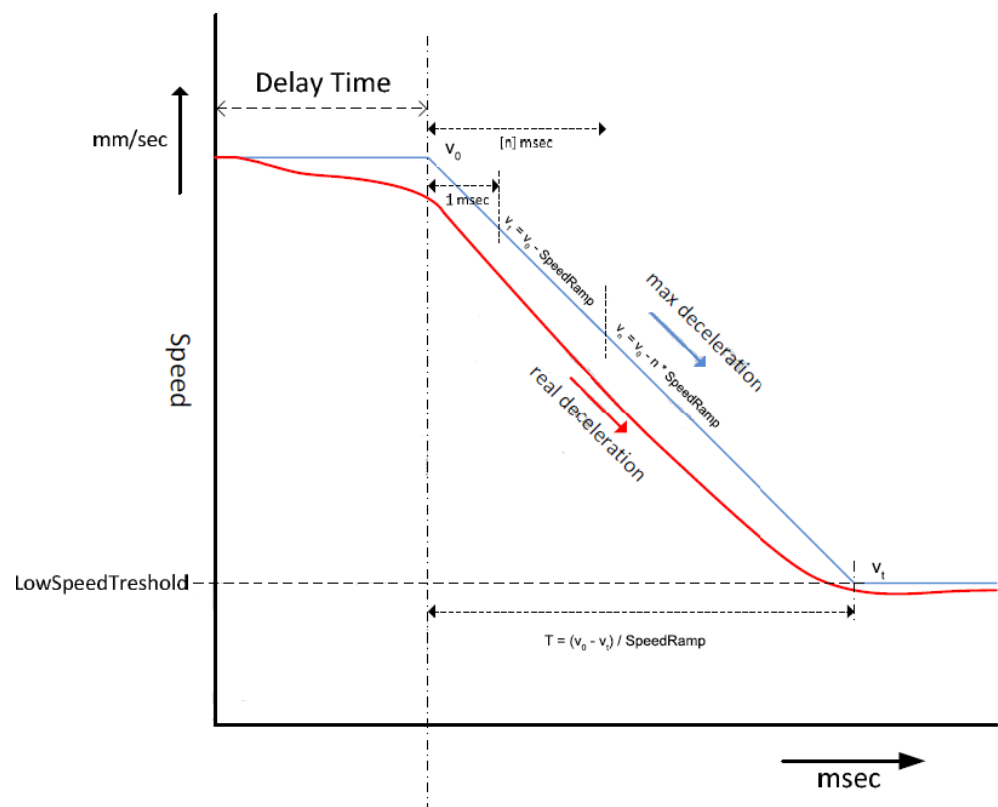
Formel zur Berechnung der Geschwindigkeitskurve

$$v_x = v_0 - \text{SpeedRamp} * (x - T_d)$$

$$T_d = T_0 + \text{DelayTime}$$

$$x > \text{DelayTime}$$

$$v_x > \text{S_LowSpeedThreshold}$$



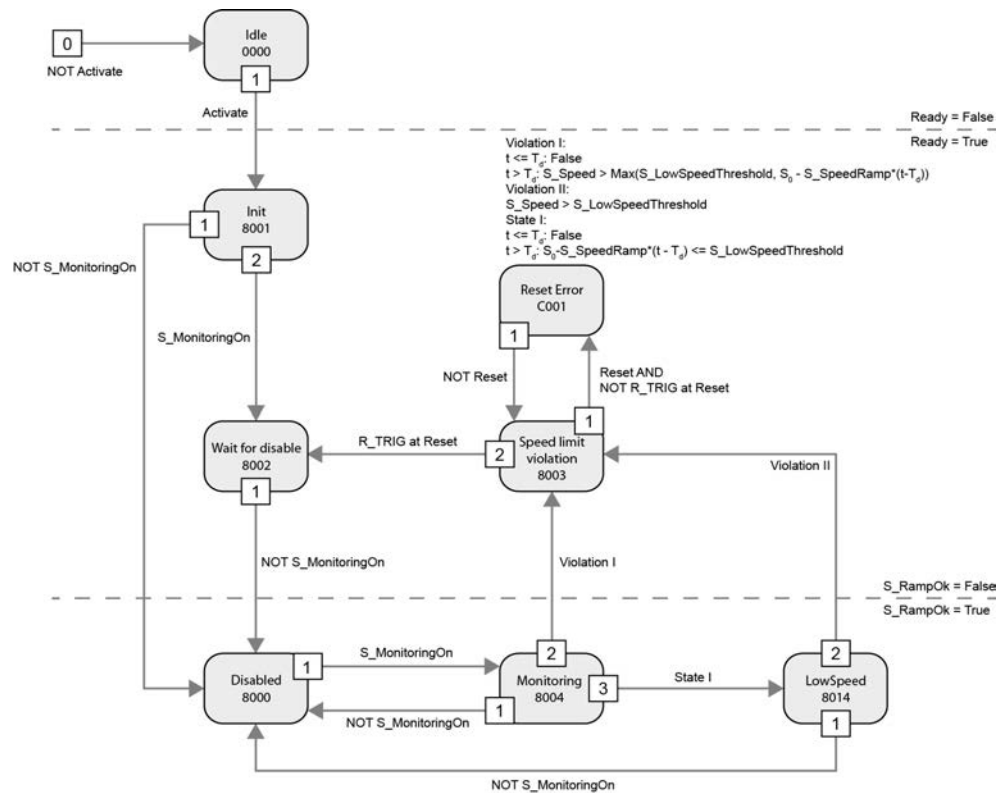
Fehlerverhalten

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal im Status 8003 Ready = TRUE S_RampOk = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_RampOk = FALSE Error = FALSE
8001	Init	Eine Aktivierung wurde vom FB erkannt und der Status ist nun aktiviert. Ready = TRUE S_RampOk = FALSE Error = FALSE
8002	Wait for Disable	Warten auf Inaktivierung der Überwachung. Ready = TRUE S_RampOk = FALSE Error = FALSE
8003	Speed Limit Violation	Geschwindigkeit am Eingang liegt über der berechneten Geschwindigkeit. Ready = TRUE S_RampOk = FALSE Error = FALSE
8000	Disabled	Überwachung ist deaktiviert. Es findet keine Überprüfung des Eingangswertes statt. Ready = TRUE S_RampOk = TRUE Error = FALSE
8004	Monitoring	Überwachung ist aktiviert. Nach Ablauf der Delaytime findet eine Überwachung des Eingangs anhand von berechneten Werten statt. Ready = TRUE S_RampOk = TRUE Error = FALSE
8014	Low Speed	Überwachung ist aktiviert. Die berechneten Werte haben den Wert von S_LowSpeedThreshold unterschritten. Die Überwachung des Eingangswertes findet nur noch auf diese Grenze hin statt. Ready = TRUE S_RampOk = TRUE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Status Diagramm SF_RampMonitor



13.5.26.5 SF_DirectionMonitor

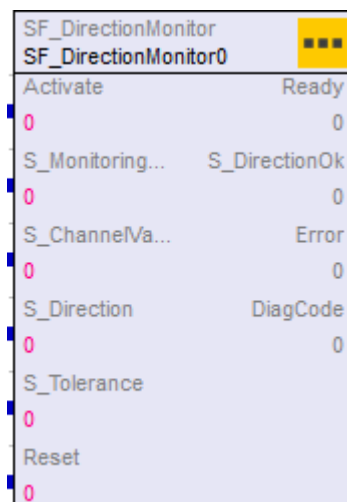
Der Eingang S_Direction bestimmt die Richtung in welche der Eingangswert geprüft werden soll (positive Fahrt S_Direction = FALSE, negative Fahrt S_Direction = TRUE).

Die Überwachung wird bei einer steigenden Flanke auf S_MonitoringOn gestartet und setzt dabei den aktuellen Wert als Maximalwert (oder Minimalwert wenn S_Direction = TRUE).

Während der Überwachung wird jeweils der höchste Wert (oder der niedrigste Wert wenn S_Direction = TRUE) gespeichert.

S_Tolerance ist die maximale Differenz zwischen aktuellem Wert und gespeichertem Maximalwert (Minimalwert wenn S_Direction = TRUE) in falscher Richtung. S_DirectionOk wird FALSE, wenn S_MonitoringOn TRUE ist und die Toleranz unter- bzw. überschritten wird.

Fehler können durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock. (BOOL)
S_MonitoringOn	Aktivierung der Überwachungsfunktion. FALSE: Keine Überwachung. Es werden keine Regelverletzungen ausgewertet. TRUE: Überwachungsfunktion ist eingeschaltet. (SAFEBOOL)
S_ChannelValue	Aktueller Wert aus dem XS-322-2INC-Modul. (SAFEDINT)
S_Direction	Überwachungsrichtung in welche sich die Werte verändern dürfen. (SAFEDINT) FALSE: positive Bewegung TRUE: negative Bewegung
S_Tolerance	Maximaler Wert in falscher Richtung. (SAFEDINT)
Reset	Zurücksetzen (BOOL)

Ausgangsvariablen

Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind. (BOOL)
-------	---

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

S_DirectionOk	Ergebnis der Überwachungsfunktion: FALSE: wenn S_MonitoringOn TRUE ist und der Eingangswert in falscher Bewegungsrichtung, gegenüber dem zuletzt ermittelten Wert in richtiger Richtung, größer als die Toleranz ist. TRUE: wenn die Richtung stimmt oder nicht aktiviert. S_DirectionOk wird ungültig, wenn einer der Safety-Eingänge nicht gültig ist. (SAFEBOOL)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)

S_ChannelValue wird als vorzeichenloser Wert interpretiert. Eine Verletzung des Wertebereiches wird ignoriert. Der FB kann daher auch für Inkrementalgeber verwendet werden.

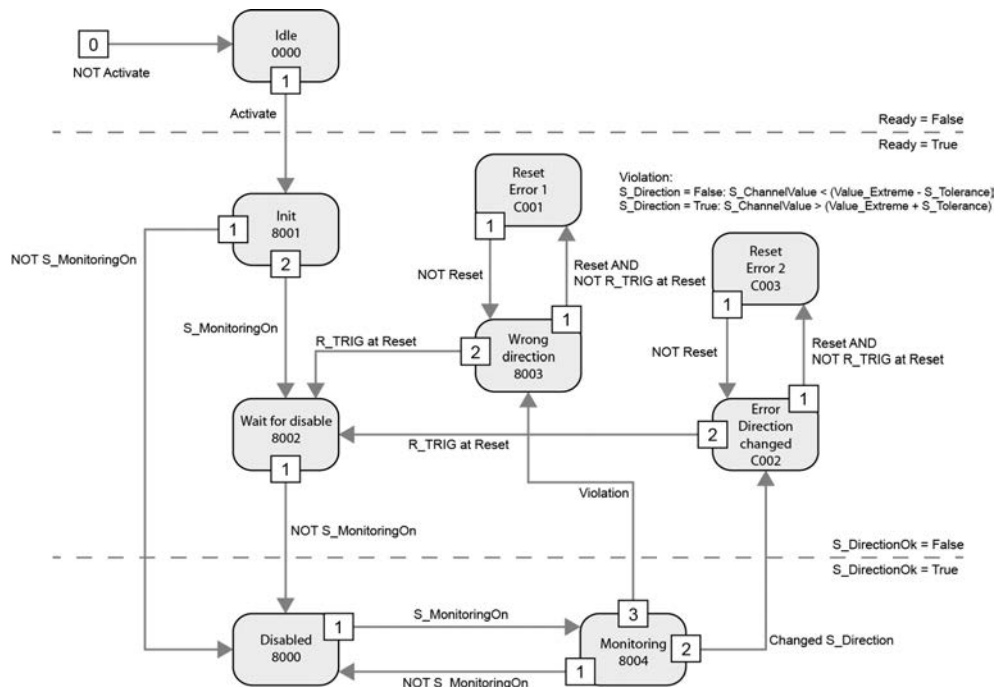
Fehlerverhalten

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Reset Error 1	Statisches Reset-Signal im Status 8003 Ready = TRUE S_DirectionOk = FALSE Error = TRUE
C002	DirectionChanged	Das Richtungsignal S_Direction hat sich während der Überwachung geändert. Ready = TRUE S_DirectionOk = FALSE Error = TRUE
C003	Reset Error 2	Statisches Reset-Signal im Status C002 Ready = TRUE S_DirectionOk = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_DirectionOk = FALSE Error = FALSE
8001	Init	Eine Aktivierung wurde vom FB erkannt und der Status ist nun aktiviert. Ready = TRUE S_DirectionOk = FALSE Error = FALSE
8002	Wait for Disable	Warten auf Inaktivierung der Überwachung. Ready = TRUE S_DirectionOk = FALSE Error = FALSE
8003	Wrong Direction	Der Eingangswert hat das vorgeschriebene Limit S_Tolerance in die falsche Richtung überschritten. Ready = TRUE S_DirectionOk = FALSE Error = FALSE
8000	Disabled	Es findet keine Überprüfung des Eingangswertes statt. Ready = TRUE S_DirectionOk = TRUE Error = FALSE
8004	Monitoring	Überwachung ist aktiviert. Es wird der Eingangswert S_ChannelValue überwacht, sodass dieser nur S_Tolerance in die umgekehrte Richtung machen darf. Ready = TRUE S_DirectionOk = TRUE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

Status Diagramm SF_DirectionMonitor



13.5.26.6 SF_SkewMonitor

Eine steigende Flanke beim Eingang S_Calibrate startet die Kalibrierung und setzt dabei S_CalibrationOk auf FALSE. Zum Kalibrieren muss der Eingang S_Monitoring auf FALSE gesetzt werden. Während der Kalibrierung wird die Differenz der Eingangssignale S_ChnValue1 und S_ChnValue2 auf S_MaxOffset überprüft. Wenn die Differenz größer als S_MaxOffset ist, wird der Ausgang S_SkewOk auf FALSE gesetzt.

Eine fallende Flanke bei S_Calibrate übernimmt die aktuelle Differenz und speichert diese remanent, wodurch der Ausgang S_CalibrationOk auf TRUE gesetzt wird. Ausgenommen der Wert übersteigt S_MaxOffset, in diesem Fall bleibt S_CalibrationOk auf FALSE.

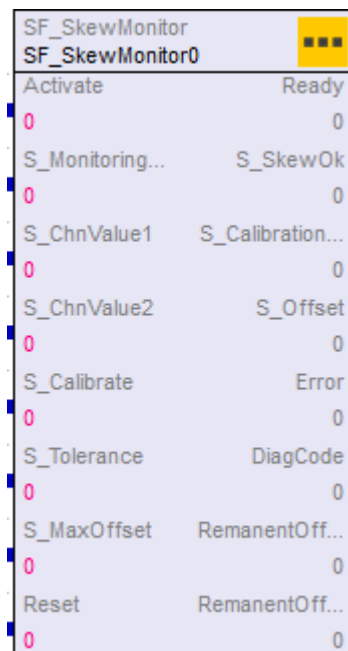
S_CalibrationOk ist nur dann TRUE, wenn der gespeicherte Offset nach einer fehlerfreien Kalibrierung gültig ist.

Wenn S_Calibrate = FALSE und S_MonitoringOn = TRUE erfolgt eine Überwachung nach folgender Regel:

$$\text{ABS}(\text{S_ChnValue1} - \text{S_ChnValue2} - \text{S_Offset}) > \text{ABS}(\text{S_Tolerance})$$

Wenn diese Überwachung zutrifft, wird S_SkewOk auf FALSE gesetzt.

Fehler können durch eine steigende Flanke am Reset-Eingang zurückgesetzt werden.



Eingangsvariablen

Activate	Aktiviert den Funktionsblock. (BOOL)
S_MonitoringOn	Aktivierung der Überwachungsfunktion. FALSE: Keine Überwachung. Es werden keine Regelverletzungen ausgewertet. TRUE: Überwachungsfunktion ist eingeschaltet. (SAFEBOOL).

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

S_ChnValue1	Gefilterter Wert von Kanal 1 des SSI-Moduls. (SAFEDINT)
S_ChnValue2	Gefilterter Wert von Kanal 2 des SSI-Moduls. (SAFEDINT)
S_Calibrate	TRUE: Startet die Kalibrierung und setzt den Ausgang S_CalibrationOk auf FALSE (SAFEBOOL)
S_Tolerance	Maximale Abweichung von Kanalwerten während der Überwachung. (SAFEDINT)
S_MaxOffset	Maximale Abweichung von Kanalwerten während der Kalibrierung. (SAFEDINT)
Reset	Zurücksetzen (BOOL)
Ausgangsvariablen	
Ready	Wenn TRUE: Anzeige, dass der Funktionsblock aktiviert ist und die Output-Ergebnisse gültig sind. (BOOL)
S_SkewOk	Ergebnis der Überwachungsfunktion: FALSE: Differenz zwischen S_ChnValue1 und S_ChnValue2 überschreitet den Toleranzwert (Absolutwerte). TRUE: beide Werte befinden sich in der aktuellen Toleranz oder sind inaktiv. S_SkewOk wird ungültig, wenn einer der Safety-Eingänge nicht gültig ist. (SAFEBOOL)
S_CalibrationOk	Information ob der FB kalibriert ist. FALSE: Der Kalibrierungswert ist ungültig oder die Kalibrierung wurde gestartet. TRUE: FB ist kalibriert (und die remanenten Werte sind gültig) (SAFEBOOL)
S_Offset	Offset-Wert Wenn der Ausgang S_CalibrationOk FALSE ist, dann ist der Offset nicht gültig. Während der Kalibrierung ist der Wert die aktuelle Differenz. (SAFEDINT)
Error	Fehleranzeige (BOOL)
DiagCode	Statusanzeige (HDINT)
RemanentOffset1	Remanent gespeicherter Offsetwert (DINT)
RemanentOffset2	Remanent gespeicherter invertierter Offsetwert (DINT)

S_MaxOffset wird bei der Kalibrierung verwendet. S_Tolerance wird bei der regulären Überwachung verwendet.

Der Kalibrierungswert (Offset) wird remanent auf den Ausgängen RemanentOffset1 und RemanentOffset2 gespeichert. Um zu erkennen, ob der gespeicherte Wert gültig ist (Speicher nicht initialisiert), wird zweimal gespeichert: einmal mit dem gleichen Wert und ein zweites mal als Einerkomplement. Der Ausgang S_CalibrationOk wird nur dann TRUE, wenn beide Werte übereinstimmen. Die remanenten Ausgänge werden während des Kalibrierens nicht beschrieben. Sie werden nur am Beginn (um den Wert ungültig zu setzen) und am Ende der Kalibrierung beschrieben, um den neuen Offset-Wert zu speichern.

Während der Kalibrierung ist die reguläre Überwachung ausgeschaltet.

Fehlerverhalten

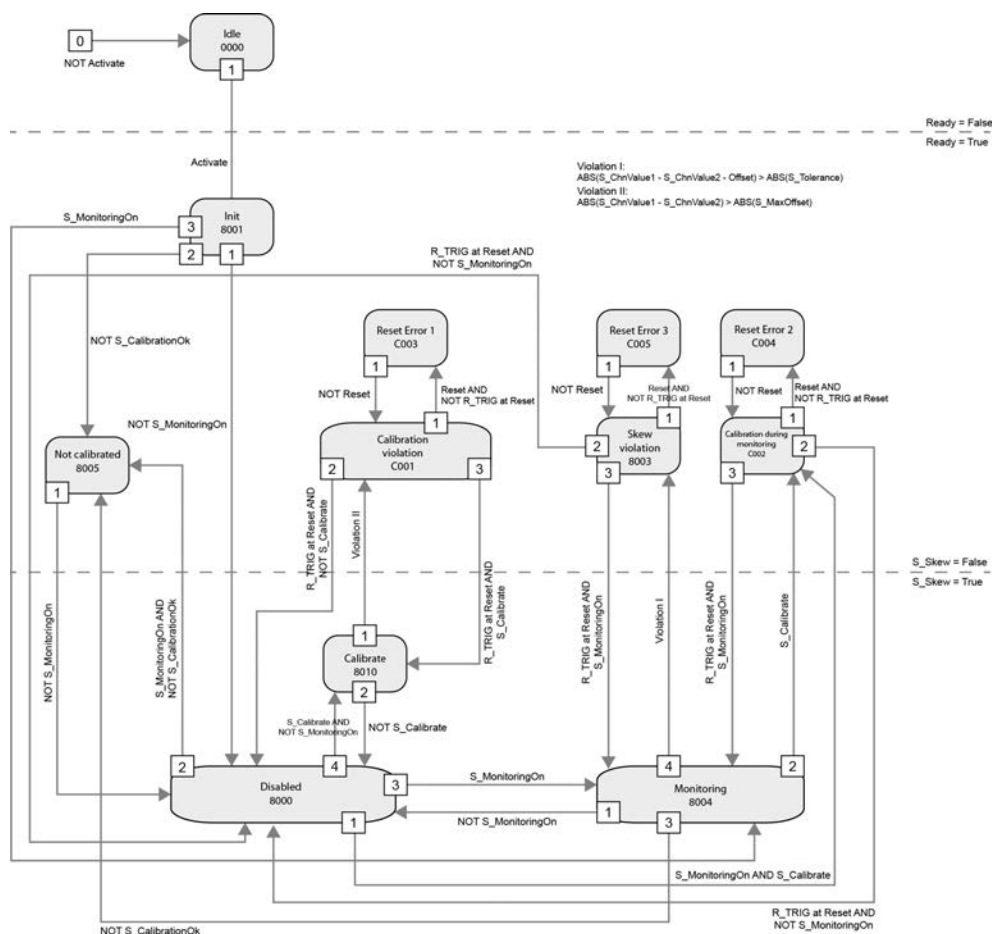
DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
FB-spezifische Fehler-Codes		
C001	Calibration Violation	Während der Kalibrierung hat die Differenz der beiden Eingangswerte den maximalen Wert S_MAXOFFSET überschritten. Ready = TRUE S_SkewOk = FALSE Error = TRUE
C002	Calibration during Monitoring	Während der Überwachung wurde die Kalibrierung aktiviert (S_MonitoringOn = TRUE und S_Calibrate = TRUE). Ready = TRUE S_SkewOk = FALSE Error = TRUE
C003	Reset Error 1	Statisches Reset-Signal im Status C001 Ready = TRUE S_SkewOk = FALSE Error = TRUE
C004	Reset Error 2	Statisches Reset-Signal im Status C002 Ready = TRUE S_SkewOk = FALSE Error = TRUE
C005	Reset Error 3	Statisches Reset-Signal im Status 8003 Ready = TRUE S_SkewOk = FALSE Error = TRUE
FB-spezifische Status-Codes (kein Fehler)		
0000	Idle	Der Funktionsblock ist nicht aktiv (Ausgangszustand). Ready = FALSE S_SkewOk = FALSE Error = FALSE
8001	Init	Eine Aktivierung wurde vom FB erkannt und der Status ist nun aktiviert. Ready = TRUE S_SkewOk = FALSE Error = FALSE
8003	Skew Violation	Differenz der Eingangswerte (inkl. S_Offset) hat die Toleranz S_Tolerance überschritten. Ready = TRUE S_SkewOk = FALSE Error = FALSE
8005	Not Calibrated	Überwachung sollte aktiviert werden, FB ist jedoch noch nicht kalibriert. Ready = TRUE S_SkewOk = FALSE Error = FALSE
8000	Disabled	Es findet keine Überprüfung des Eingangswertes statt. Ready = TRUE S_SkewOk = TRUE Error = FALSE

13. Funktionsblöcke

13.5 Komplexe Funktionsblöcke

DiagCode	Statusname	Statusbeschreibung und Ausgangseinstellung
8004	Monitoring	Überwachung ist aktiviert. Die Differenz der Eingangswerte inklusive Kalibrierungswert S_Offset wird mit S_Tolerance verglichen. Ready = TRUE S_SkewOk = TRUE Error = FALSE
8010	Calibrate	Ermitteln der Differenz der beiden Eingangswerte. Bei Verlassen des Status wird der Wert remanent gespeichert (Fallende Flanke bei S_Calibrate). Ready = TRUE S_SkewOk = TRUE S_CalibrationOk = FALSE Error = FALSE

Status Diagramm SF_SkewMonitor



13.6 Funktionsblock-Makros

13.6.1 Beschreibung

Funktionsblock Makros sind komplette Netzwerke, die wie ein Funktionsblock als Instanz in einem Netzwerk verwendet werden können. Macros können mittels Drag & Drop in ein Netzwerk gezogen werden und im Netzwerk wie Funktionsblöcke verwendet werden.

13.6.2 Begriffe

Lokales Makro	Ein Funktionsblock-Makro, welches zum aktuellen Projekt gehört und auch mit diesem gespeichert wird.
Referenz-Netzwerk	Netzwerk des Funktionsblock-Makros. Dieses Öffnet man durch einen Doppelklick auf das Symbol in der Tree-Ansicht des Makros oder mittels „Open Macro Referenz Network“ aus dem Kontextmenü einer Instanz des Makros (Makro wird in einem Netzwerk als Instanz verwendet).
Objekt-Netzwerk	Kopie des Referenz-Netzwerkes, welches in einer Instanz des Makros gespeichert ist. Dieses Netzwerk öffnet man durch einen Doppelklick in die rechte obere Ecke der Instanz eines Makros oder durch „Open Macro Object Network“ aus dem Kontextmenü der Instanz.
Makro-Bibliothek	Eine Sammlung von FB-Makros unter einem Library-Symbol, die abgespeichert und verwendet werden können.
Embedded Makro	In ein anderes Makro eingebundenes Makro (Makro in Makro).
Derived Makro	Ein abgeleitetes Makro / übergehendes Makro.



Veränderungen, die man im Referenz-Netzwerk oder einem Objekt-Netzwerk vornimmt, werden auf alle Objekt-Netzwerke und das Referenz-Netzwerk übertragen.

13. Funktionsblöcke

13.6 Funktionsblock-Makros

13.6.3 Erstellen von Makros

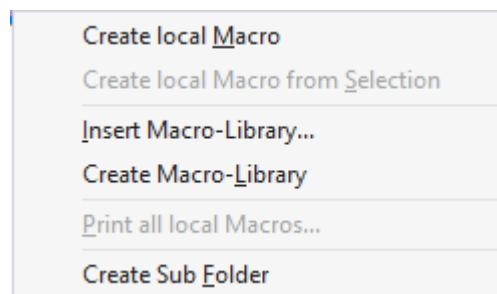
- "Zusammenstellen eines Makros", Seite 363
- "Erstellen eines Makros aus vorhandenen Elementen", Seite 365
- "Einfügen der Anschlüsse in das Makro-Netzwerk", Seite 366
- "Erstellen von eingebetteten Makros", Seite 368
- "Erstellen einer Kopie eines Makros", Seite 368

Siehe auch

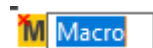
- "Fehler-Codes", Seite 386
- "Standard-Funktionsblöcke", Seite 167
- "Komplexe Funktionsblöcke", Seite 210

13.6.3.1 Zusammenstellen eines Makros

Im Funktionsblock-Macro-Tree kann mittels Rechtsklick beim Folder „Macro Libraries“ über „Create local Macro“ ein lokales Makro erstellt werden. Mit einem Rechtsklick auf eine bereits bestehende Makro-Bibliothek und „Create Macro“ kann ein Makro in der Makro-Bibliothek erstellt werden.



Der Namen des Makros kann mittels [F2]-Taste oder über das Kontextmenü geändert werden. Bei der Eingabe von ungültigen Namen wird der Text rot angezeigt.



Nach dem Erstellen kann der Baum des Makros aufgeklappt und mit Doppelklick auf das Netzwerksymbol (trägt den Namen des Makros), das noch leere Referenz-Netzwerk geöffnet werden.

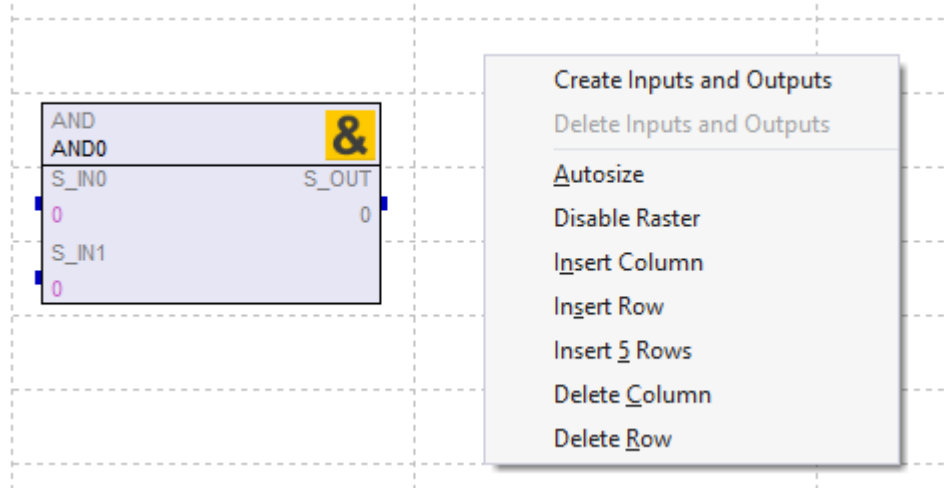


Mittels Drag & Drop können Funktionsblöcke, Konstanten und Makros hineingezogen und untereinander verbunden werden. Den Konstanten kann man wie gewohnt Werte zuweisen. Die Makroanschlüsse können als Elemente in den Ein- und Ausgangsspalten mittels Kontextmenü im Netzwerk „Create Inputs and Outputs“ auto-

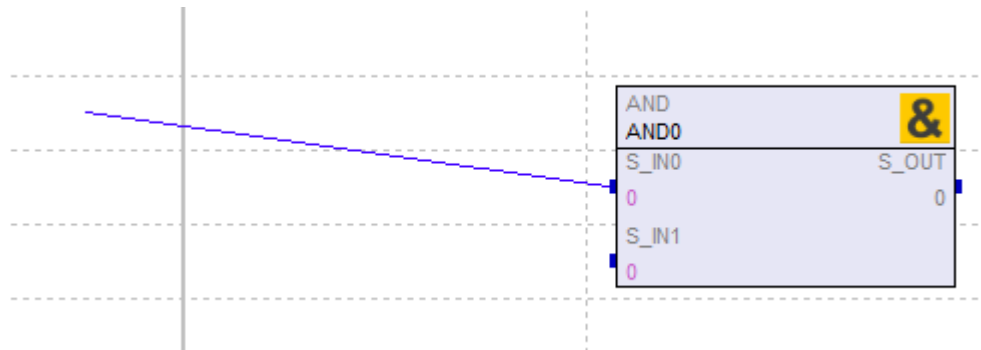
13. Funktionsblöcke

13.6 Funktionsblock-Makros

matisch erstellt werden, es werden hier automatisch alle Eingänge und Ausgänge von Instanzen nach Außen gezogen.



Sollen nur einzelne Eingänge und Ausgänge nach Außen gezogen werden, kann dies durch ein manuelles Ziehen der Eingänge bzw. Ausgänge der Instanzen in die Ein- oder Ausgangsspalte des Netzwerks erfolgen.

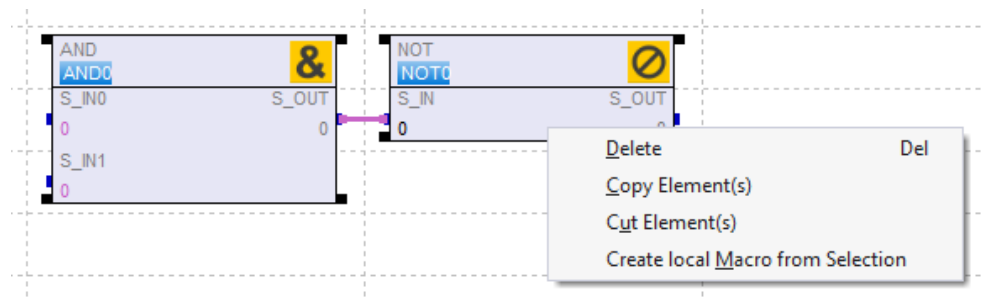


13. Funktionsblöcke

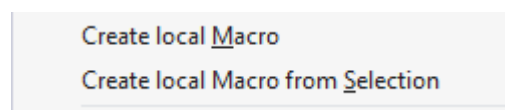
13.6 Funktionsblock-Makros

13.6.3.2 Erstellen eines Makros aus vorhandenen Elementen

Markiert man in einem Netzwerk eine Gruppe mit Instanzen, Konstanten, Makros und Kommentaren und wählt anschließend über das Kontextmenu „Create local Macro from Selection“ aus, wird ein Makro generiert.



Zusätzlich besteht die Möglichkeit mittels „Cut“ oder „Copy“ die Elemente zu selektieren und dann im Funktionsblock-Macro-Tree beim Kontextmenü „Create local Macro from Selection“ auszuwählen



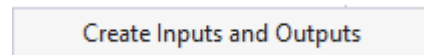
Im Funktionsblock-Macro-Tree kann das neu generierte Makro ausgewählt werden. Beim Anlegen des Makros wird der Name „Macro“ mit einer Zahl (z.B. Macro0) vergeben. Das Makro kann danach, wie zuvor beschrieben, bearbeitet werden.



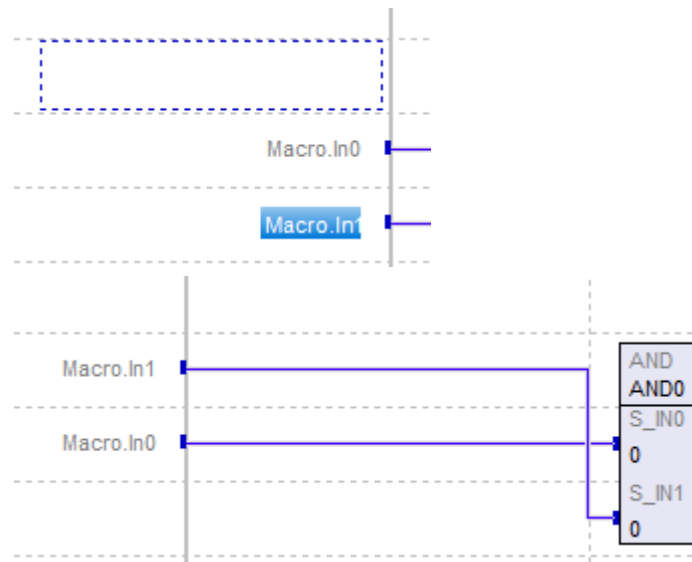
Funktion Block Makros können ausschließlich folgende Elemente enthalten: Funktionsblöcke, Konstanten, Verbindungen, Kommentare sowie eingebettete Makros.

13.6.3.3 Einfügen der Anschlüsse in das Makro-Netzwerk

Nach Rechtsklick in ein Referenz- oder Objekt-Netzwerk erscheint der Punkt „Create Macro Inputs and Outputs“.



Mit dieser Funktion wird automatisch für jeden Ein- und Ausgang des Makros ein Element in der Ein- oder Ausgangsspalte erstellt und mit der zugehörigen Instanz verbunden. Möchte man eine Instanz mit einem anderen, schon vorhandenen Eingang verbinden, muss man die Verbindung löschen und zu dem anderen Eingang neu ziehen. Die Reihenfolge der Makro-Anschlüsse richtet sich dabei nach den Elementen in den Ein- und Ausgangsspalten, d.h. durch Verschieben kann man diese ändern.



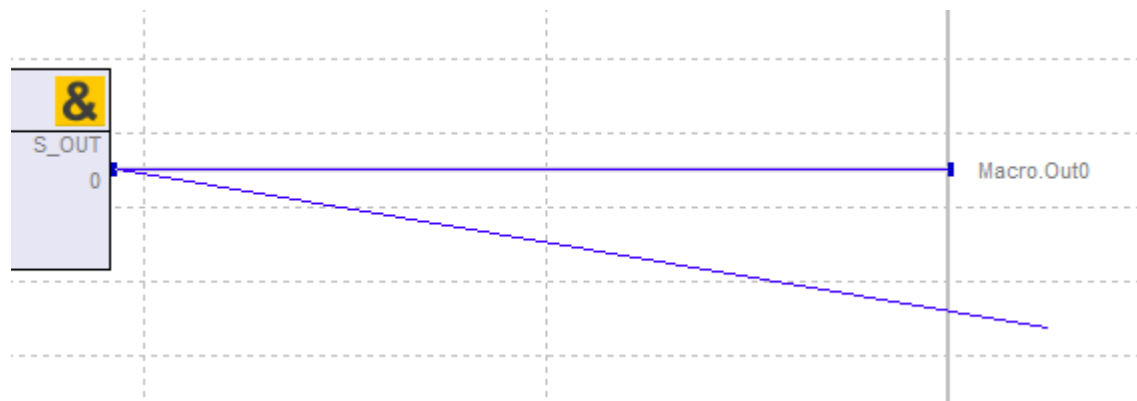
Die Elemente in der Eingangs- und Ausgangsspalte können auch durch das Ziehen von Verbindungen erstellt werden. Dazu wird eine Verbindung von einem nicht verbundenen Eingang einer Instanz in die Eingangsspalte gezogen werden. Durch den Klick in die Eingangsspalte wird ein neuer Makroeingang erstellt.



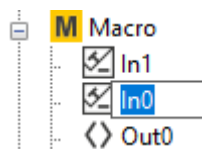
Für die Ausgänge funktioniert dies auch, auch wenn diese bereits verbunden sind, um einen zusätzlichen Ausgang zu erstellen.

13. Funktionsblöcke

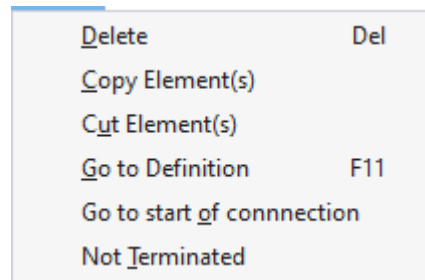
13.6 Funktionsblock-Makros



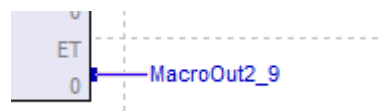
Die Makro-Anschlüsse in der Eingangs- Und Ausgangsspalte kann man im Tree umbenennen.



Nicht erforderliche Ausgänge können in der Ausgangsspalte auf „Not Terminated“ gesetzt werden.



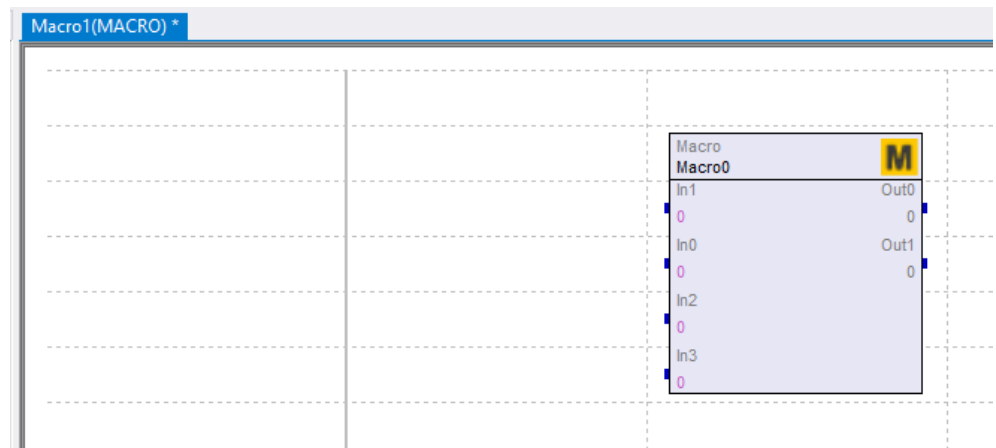
Der Makroausgang wird gelöscht und es wird eine temporäre globale Verbindung angelegt.



➔ Solange die Anschlüsselemente des Makros nicht erzeugt sind, zeigt eine Instanz keine Ein und Ausgänge an.

13.6.3.4 Erstellen von eingebetteten Makros

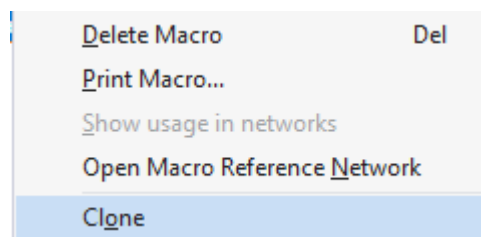
Es können Makros als Instanzen in Makros eingefügt werden. Das Einfügen erfolgt mittels Drag & Drop in das Netzwerk eines Makros. Wichtig hierbei ist, dass das eingebundene Makro bereits vollständig erstellt ist und die Ein- und Ausgangselemente erstellt wurden.



Es ist darauf zu achten, dass beim Platzieren von eingebetteten Makros, Makros nicht im eigenen Netzwerk eingefügt werden (auch nicht rekursiv).

13.6.3.5 Erstellen einer Kopie eines Makros

Bei einem selektierten Makro kann über das Kontextmenü mit dem Befehl „Clone“ eine Kopie des Makros angelegt werden.



13. Funktionsblöcke

13.6 Funktionsblock-Makros

13.6.4 Makro-Bibliotheken

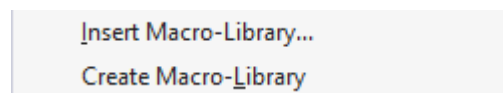
- "Erstellen von Makro-Bibliotheken", Seite 369
- "Speichern von Makro-Bibliotheken", Seite 370
- "Anzeigen von Makro-Bibliotheken", Seite 371
- "Verwenden von Makros aus Bibliotheken", Seite 371
- "Lokale Makro-Bibliothek", Seite 372
- "Sperrern von Makro-Bibliotheken", Seite 373
- "Vergleich von Makros", Seite 375

Siehe auch

- "Erstellen von Makros", Seite 363

13.6.4.1 Erstellen von Makro-Bibliotheken

Im Kontextmenü des Elements "Macro Libraries" des FunctionBlock-Macro-Tree den Punkt "Create Macro Library" auswählen.



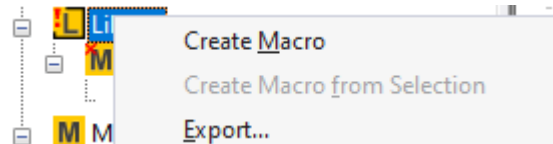
Es wird eine Library für Makros angelegt. Der Namen der Makro Libray kann mittels [F2]-Taste oder über das Kontextmenü geändert werden.



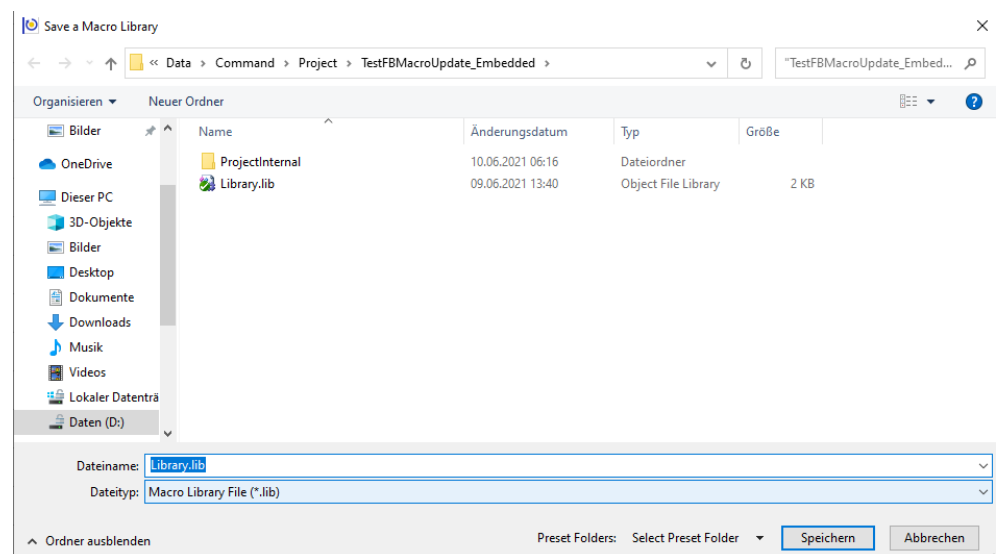
Bei der Eingabe von ungültigen Namen wird der Text rot angezeigt. Innerhalb der Makro-Library können über das Kontextmenü Makros in die Library eingefügt werden.

13.6.4.2 Speichern von Makro-Bibliotheken

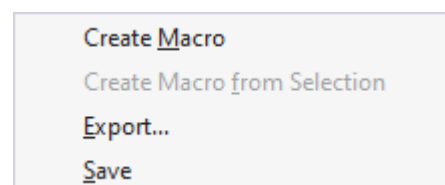
Jede neu erstellte Library muss zuerst einmal exportiert werden. Über das Kontextmenü wählt man den Punkt „Export...“ aus.



Im Speicherdialog wählt man den Speicherort und ändert eventuell den vorgeschlagenen Dateinamen.



In weiterer Folge kann man dann über den Punkt „Save“ im Kontextmenü Änderungen speichern in der Library speichern, es ist kein erneuter Export nötig.



Man kann die Bibliothek, falls sie nicht gesperrt wurde, wieder exportieren, d.h. einen anderen Speicherort und Dateinamen zuweisen. Dabei bleibt die zuvor angelegte Datei unverändert erhalten.



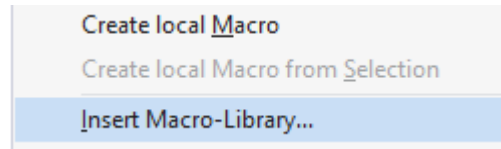
Für den Menüpunkt „Save“ einer Makro-Bibliothek wird immer die Datei verwendet, die beim letzten Export angelegt bzw. aus der sie geladen wurde.

13. Funktionsblöcke

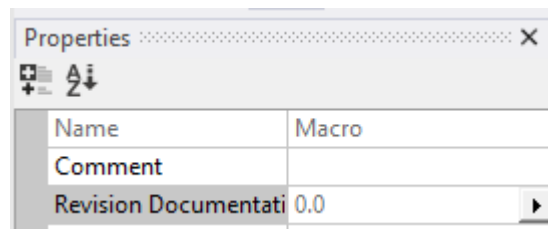
13.6 Funktionsblock-Makros

13.6.4.3 Anzeigen von Makro-Bibliotheken

Mit der Funktion „Insert Macro Library“ aus dem Kontextmenü von „Macro Libraries“ im Funktion-Block-Macro-Tree kann man eine Bibliothek aus einer Datei laden.

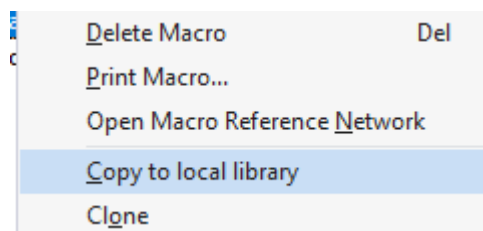


Durch Aufklappen der Tree-Elemente kann man den Inhalt anzeigen und durch Öffnen der Referenz-Netzwerke der Makros erkennt man deren Funktionalität. Weiters können jede Library und jedes Makro einen Kommentar und eine eigene Versions-Dokumentation enthalten.



13.6.4.4 Verwenden von Makros aus Bibliotheken

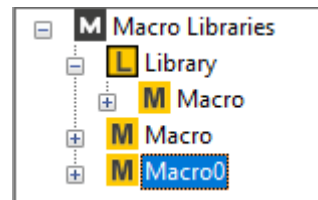
Sowohl Makros aus Libraries, als auch lokale Makros lassen sich einfach per Drag & Drop aus dem Tree in ein Netzwerk ziehen. Bei Makros aus Libraries wird dabei automatisch eine Kopie im Projekt d.h. unter den lokalen Makros angelegt. Eingebettete Makros werden dabei mitkopiert. Diese haben, falls der Name noch nicht verwendet wird, denselben Schlüssel wie in der Bibliothek, ansonsten wird eine Zahl an den Namen angehängt. Makros in Libraries lassen sich auch per Kontextmenupunkt „Copy to local library“ ins Projekt kopieren.



13.6.4.5 Lokale Makro-Bibliothek

Die lokalen Makros sind diejenigen, die zu einem Projekt gehören und auch mit diesem abgespeichert werden.

Sie werden im Funktionsblock-Macro-Tree unterhalb der Libraries aufgelistet, da sie zu keiner externen Bibliothek gehören.



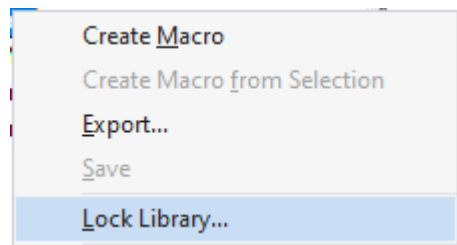
Alle Makros, die im Projekt verwendet werden, sind an dieser Stelle aufgelistet. Diese werden mit grauer Schrift angezeigt und lassen sich nicht löschen. Lokale Makro-Bibliotheken von Project Libraries, werden nicht im Library-Tree angezeigt, sondern im FunctionBlock-Library-Tree. Dies erfolgt wie bei einer gesperrten externen Bibliothek mit dem Namen „Local“ und dem Projekt-Library-Namen in runden Klammern. Man kann daran nichts verändern, sondern nur ganze Makros in die locale Makro-Library des aktuellen Projekts kopieren.

13. Funktionsblöcke

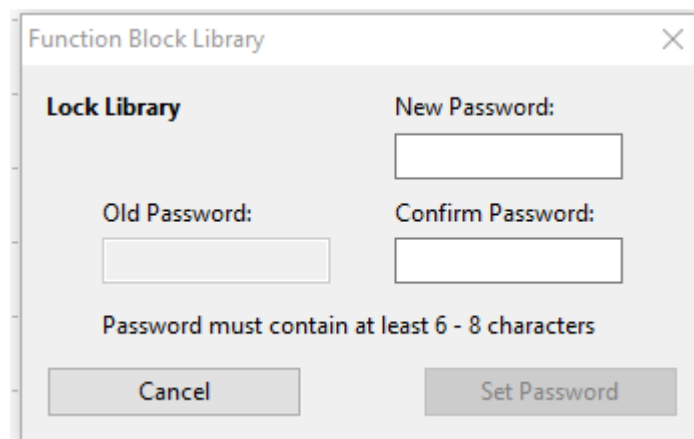
13.6 Funktionsblock-Makros

13.6.4.6 Sperren von Makro-Bibliotheken

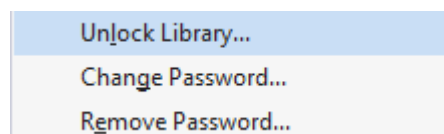
Macro-Libraries können über der Kontextmenü gesperrt werden.



Bei Aufruf dieses Befehls öffnet sich ein Dialog und es kann ein Passwort zum Sperren der Library eingefügt werden. Das Festlegen eines Passworts muss durch eine zweite Eingabe bestätigt werden.



Wenn eine Sperre der Library festgelegt ist, kann die Library nur nach Eingabe des Passworts geöffnet werden. Der Aufruf erfolgt über das Kontextmenü der Library. Ein Löschen bzw. Ändern des Passworts ist ebenfalls über das Kontextmenü möglich.



Bei der Auswahl einer der 3 Punkte erscheint wieder ein Dialog mit entsprechender Eingabeaufforderung.

Function Block Library

×

Change Password

New Password:

Old Password:

Confirm Password:

Password must contain at least 6 - 8 characters

Cancel

Set Password

13. Funktionsblöcke

13.6 Funktionsblock-Makros

Compare Macros

Parameter	Values
Name	Macro
ExternLibrary	Local
Comment	
Revision	0.1
Input	In0
Input	In1
Output	Out0
Macro-Connection	In0 -> AND0.S_IN0
Macro-Connection	In1 -> AND0.S_IN1
Macro-Connection	AND0.S_OUT -> Out0
Instance	AND0

Values
Macro
Local
0.0
In0
In1
Out0
In0 -> AND0.S_IN0
In1 -> AND0.S_IN1
AND0.S_OUT -> Out0
AND0

Static

Ok Cancel

Über die Befehle „Show Revision Documentation“ werden die Dokumentationen im Projekt und in der Library angezeigt.

Revision Documentation of Macro(Project)

Owner

Company: Author:

Revision	Date	Company	Author	Description
0.1	2020-08-26			

Insert OK Cancel

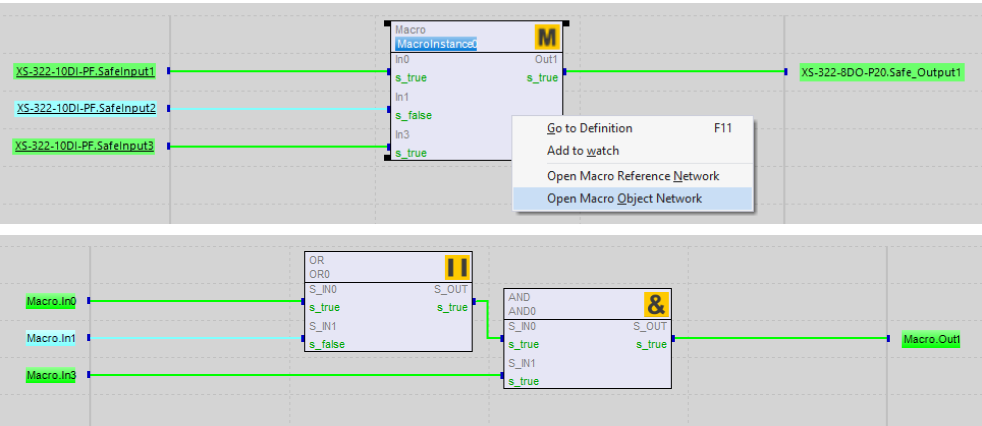
Über den Update-Button wird das Makro im Projekt durch das Makro aus der Library ersetzt. Im XN Safety Designer wird überprüft, ob ein Ersetzen möglich ist. Durch das Update kann es sein, dass neue Eingänge/Ausgänge hinzugefügt werden. Dies kann dazu führen, dass das Makro im Netzwerk nicht genügend Platz hat und andere Elemente verschoben werden müssen. Falls dies nicht möglich ist, kann ein Update untersagt werden. Der Anwender muss dann zuerst genügend Platz im Netzwerk schaffen.

13. Funktionsblöcke

13.6 Funktionsblock-Makros

13.6.5 Debuggen von Funktion Block Macros

Im Debug-Modus kann man überprüfen, was innerhalb jeder einzelnen Makroinstanz abläuft. Dazu muss man das Objekt-Netzwerk der Instanz öffnen, durch Doppelklick auf die Instanz oder über das Kontextmenu.



Der Name und die Reihenfolge der Elemente in Ein- und Ausgangsspalte des Objekt-Netzwerkes entsprechen den Anschlüssen der Makro-Instanz. Die Werte der Anschlusselemente kann man nicht forcen, ansonsten gilt alles, was im Kapitel „Gra- fischer Editor“ über Debuggen, Forcen und Watchfenster zu lesen ist.

Watch			
Name		Value	Type
☒ XS-102-C00-000/Network/XS-322-10DI-PF.Safe_Input1		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/MacroInstance0.In0		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/MacroInstance0/Macro.In0		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/MacroInstance0/OR0.S_IN0		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/MacroInstance0/AND0.S_IN0		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/MacroInstance0/AND0.S_OUT		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/MacroInstance0/Macro.Out1		s_true	SAFEBOOL
☒ XS-102-C00-000/Network/XS-322-8DO-P20.Safe_Output1		s_true	SAFEBOOL
Insert new via Drag&Drop or context menu (only in Online Mode)			

14. Safety-Module

14.1 Beschreibung Safety-CPU-Modul XS-102-C00-000

Das XN Safety CPU-Modul XS-102-C00-000 unterstützt bis zu 16 Safety I/O-Module.

Zusätzlich ist das Safety-CPU-Modul in der Lage, Handbediengeräte mit Not-Halt- bzw. Zustimmungstaster zu bedienen.

Das Safety-CPU-Modul besitzt den Sicherheitsintegritätslevel SIL3 (EN IEC 62061) bzw. Performancelevel e (PL e) (EN ISO 13849).

Das sicherheitsbezogene Safety-CPU-Modul ist geeignet für die Verwendung in Systemen mit optionalen Modulen und Interface-Variablen gemäß XN Safety Systemhandbuch, → "Weitere Nutzungsinformationen", Seite 447

Das Safety-CPU-Modul allein bildet bereits ein Minimalsystem einer Sicherheitssteuerung.



XS-102-C00-000

Darüber hinaus regelt das Safety-CPU-Modul die zeitkorrekte Kommunikation mit den entfernten Sicherheitsmodulen über sichere Bustelegramme. Zu ihren Aufgaben gehört:

- die Abarbeitung der sicheren Applikation
- und
- die Verteilung der Konfigurationsdaten an entfernte Sicherheitsmodule

Lieferumfang

- 1x XS-102-C00-000
- 1x Steckverbinder PLUG-FMC-3S (1,5/3-ST-3,5)
- 1x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

14. Safety-Module

14.2 Beschreibung Safety Analog-Eingangsmodul XS-322-4AI-I2

14.2 Beschreibung Safety Analog-Eingangsmodul XS-322-4AI-I2

Das Safety Analog-Eingangsmodul XS-322-4AI-I2 stellt die Werte von vier analogen Stromeingängen der sicheren CPU (XN Safety CPU-Modul) zur Verfügung.

Die vier Stromeingänge besitzen einen Eingangsbereich von 4-20 mA mit einer Auflösung von 16 Bit.

Für die Analogeingänge wird eine +24 V-Sensorversorgung zur Verfügung gestellt. Diese muss extern gespeist werden. Die externe +24 V-Spannungsversorgung für die Analogeingänge wird auf Über- und Unterspannung überwacht.

Die sichere Auswertung der Analogeingänge wird durch zwei unabhängige diversitäre Auswertungen mit gegenseitiger Überwachung sichergestellt.



XS-322-4AI-I2

Die Sicherheitsfunktionen der Baugruppe erfüllen

- **bei zweikanaliger Verwendung**
die Anforderungen nach SIL3 gemäß EN IEC 62061 und Performancelevel e (PL e), Kat. 4 gemäß EN ISO 13849.

sowie

- **bei einkanaliger Verwendung**
die Anforderungen nach SIL3 gemäß EN IEC 62061 und Performancelevel e (PL e), Kat. 3 gemäß EN ISO 13849.

Lieferumfang

1x XS-322-4AI-I2

5x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

Siehe auch



Handbuch Safety Analog-Eingangsmodul XS-322-4AI-I2

MN050025

14.3 Beschreibung Safety Digital-Eingangsmodul XS-322-10DI-PF

14.3 Beschreibung Safety Digital-Eingangsmodul XS-322-10DI-PF

Das Safety Digital-Eingangsmodul XS-322-10DI-PF besitzt den Sicherheitsintegritätslevel SIL3 (EN IEC 62061) bzw. Performancelevel e (PL e) (EN ISO 13849).

Das Safety Digital-Eingangsmodul verfügt über:

- 10 sichere Eingänge (EN 61131-2; EN IEC 62061 und EN ISO 13849)
- Doppelt ausgeführter Taktausgang (kurzschlussfest)

Die sicheren Eingänge dienen zum sicherheitsgerichteten Einlesen von 10 Sensor-Signalen (NOT-Halt, Zustimmungstaster usw.).

Um Eingänge testen und Querschlüsse erkennen zu können (z.B. NOT-Halt) besitzt das Safety Digital-Eingangsmodul zwei nicht sicherheitsgerichtete Taktausgänge TA und TB.

Das sicherheitsbezogene Safety Digital-Eingangsmodul ist geeignet für die Verwendung in Systemen mit optionalen Modulen und Interface-Variablen gemäß XN Safety Systemhandbuch, → Abschnitt "Weitere Nutzungsinformationen", Seite 447



XS-322-10DI-PF

Um in einer Applikation verwendet werden zu können, benötigt das Safety Digital-Eingangsmodul mindestens auch ein XN Safety CPU-Modul XS-102-C00-000, welches über sichere Bustelegramme die zeitkorrekte Kommunikation mit den Sicherheitsmodulen regelt. Dazu gehört ferner

- die Abarbeitung der sicheren Applikation
- und
- die Verteilung der Konfigurationsdaten an entfernte Sicherheitsmodule.

Lieferumfang

1x XS-322-10DI-PF

5x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

Siehe auch

Handbuch XN Safety-Modul XS-322-10DI-PF

MN050021

14. Safety-Module

14.4 Beschreibung Safety Digital-Ausgangsmodul XS-322-8DO-P20

14.4 Beschreibung Safety Digital-Ausgangsmodul XS-322-8DO-P20

Das Safety Digital-Ausgangsmodul XS-322-8DO-P20 besitzt den Sicherheitsintegritätslevel SIL3 (EN IEC 62061) bzw. Performancelevel e (PL e) (EN ISO 13849).

Das Safety Digital-Ausgangsmodul verfügt über:

- 8 sichere Ausgänge (EN 61131-2; EN IEC 62061 und EN ISO 13849)
- Doppelt ausgeführter Taktausgang (kurzschlussfest)

Die sicheren Ausgänge dienen zur sicherheitsgerichteten Ausgabe von 8 Sensor-Signalen, beispielsweise zur Ansteuerung von Relais oder Ventilen.

Das sicherheitsbezogene Safety Digital-Ausgangsmodul ist geeignet für die Verwendung in Systemen mit optionalen Modulen und Interface-Variablen gemäß XN Safety Systemhandbuch, → "Weitere Nutzungsinformationen", Seite 447



XS-322-8DO-P20

Um in einer Applikation verwendet werden zu können, benötigt das Safety Digital-Ausgangsmodul mindestens auch ein XN Safety CPU-Modul XS-102-C00-000, welches über sichere Bustelegramme die zeitkorrekte Kommunikation mit den Sicherheitsmodulen regelt. Dazu gehört ferner

- die Abarbeitung der sicheren Applikation
- und
- die Verteilung der Konfigurationsdaten an entfernte Sicherheitsmodule.

Lieferumfang

1x XS-322-8DO-P20

5x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

Siehe auch



Handbuch Safety Digital-Ausgangsmodul XS-322-8DO-P20

MN050022

14.5 Beschreibung Safety Digital-Ein-/Ausgangsmodul XS-322-8DIO-PF20

Das Safety Digital-Ein-/Ausgangsmodul XS-322-8DIO-PF20 besitzt den Sicherheitsintegritätslevel SIL3 (EN IEC 62061) bzw. Performancelevel e (PL e) (EN ISO 13849).

Das Safety Digital-Ein-/Ausgangsmodul verfügt über:

- 2 sichere Ausgänge +24 V mit maximal 2 A pro Ausgang (EN 61131-2; EN IEC 62061 und EN ISO 13849)
- 6 sichere Eingänge +24 V DC/0,5 ms (EN 61131-2; EN IEC 62061 und EN ISO 13849)
- Doppelt ausgeführter Taktoutput (kurzschlussfest)

Die sicheren Ausgänge dienen zur sicherheitsgerichteten Ausgabe von zwei Sensor-Signalen, beispielsweise zur Ansteuerung von Relais oder Ventilen.

Die sicheren Eingänge dienen zum sicherheitsgerichteten Einlesen von sechs Sensor-Signalen (NOT-Halt, Zustimmungstaster usw.).

Um Eingänge testen und Querschüsse erkennen zu können (z.B. NOT-Halt) besitzt das Safety Digital-Ein-/Ausgangsmodul zwei nicht sicherheitsgerichtete Taktgänge TA und TB.



XS-322-8DIO-PF20

Das sicherheitsbezogene Safety Digital-Ein-/Ausgangsmodul ist geeignet für die Verwendung in Systemen mit optionalen Modulen und Interface-Variablen gemäß XN Safety Systemhandbuch, → Abschnitt "Weitere Nutzungsinformationen", Seite 447

Um in einer Applikation verwendet werden zu können, benötigt das Safety Digital-Ein-/Ausgangsmodul mindestens auch ein XN Safety CPU-Modul XS-102-C00-000, welches über sichere Bustelegramme die zeitkorrekte Kommunikation mit den Sicherheitsmodulen regelt. Dazu gehört ferner

- die Abarbeitung der sicheren Applikation
- und
- die Verteilung der Konfigurationsdaten an entfernte Sicherheitsmodule.

Lieferumfang

- 1x XS-322-8DIO-PF20
- 5x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

Siehe auch



Handbuch Safety Digital-Ein-/Ausgangsmodul XS-322-8DIO-PF20 MN050023

14. Safety-Module

14.6 Beschreibung Safety Relais-DC-Ausgangsmodul XS-322-2DO-RNODC

14.6 Beschreibung Safety Relais-DC-Ausgangsmodul XS-322-2DO-RNODC

Das Safety Relais-DC-Ausgangsmodul XS-322-2DO-RNODC besitzt den Sicherheitsintegritätslevel SIL3 (EN IEC 62061) bzw. Performancelevel e (PL e) (EN ISO 13849).

Das Safety Relais-DC-Ausgangsmodul verfügt über:

- 2 sichere Ausgänge (EN 61131-2; EN IEC 62061 und EN ISO 13849)

Die beiden Ausgänge dienen dem sicherheitsgerichteten Schließen (NO) eines Stromkreises bei einer zulässigen Nennspannung von +24 V DC und einem maximalen Dauerstrom von 6 A.

Das Safety Relais-DC-Ausgangsmodul findet beim Schalten höherer Leistungen bzw. potentialgetrennten Kreisen seinen Einsatz – wie beispielsweise hydraulischen Ventilen oder Weitergabe von Not-Halt-Kreisen.



XS-322-2DO-RNODC

Das sicherheitsbezogene Safety Relais-DC-Ausgangsmodul ist geeignet für die Verwendung in Systemen mit optionalen Modulen und Interface-Variablen gemäß XN Safety Systemhandbuch, → "Weitere Nutzungsinformationen", Seite 447

Um in einer Applikation verwendet werden zu können, benötigt das Safety Relais-DC-Ausgangsmodul mindestens auch ein XN Safety CPU-Modul XS-102-C00-000, welches über sichere Bustelegramme die zeit-korrekte Kommunikation mit den Sicherheitsmodulen regelt. Dazu gehört ferner

- die Abarbeitung der sicheren Applikation
- und
- die Verteilung der Konfigurationsdaten an entfernte Sicherheitsmodule.

Lieferumfang

1x XS-322-2DO-RNODC

2x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

Siehe auch



Handbuch Safety Relais-DC-Ausgangsmodul XS-322-2DO-RNODC

MN050027

14.7 Beschreibung Safety Relais-Ausgangsmodul XS-322-2DO-RNOAC

Das Safety Relais-Ausgangsmodul XS-322-2DO-RNOAC besitzt den Sicherheitsintegritätslevel SIL3 (EN IEC 62061) bzw. Performancelevel e (PL e) (EN ISO 13849).

Das Safety Relais-Ausgangsmodul verfügt über:

- 2 sichere Ausgänge (EN 61131-2; EN IEC 62061 und EN ISO 13849)

Die beiden Ausgänge dienen dem sicherheitsgerichteten Schließen (NO) eines Stromkreises bei einer zulässigen Nennspannung von +24 V DC/230 AC und einem maximalen Dauerstrom von 6 A.

Das Safety Relais-Ausgangsmodul findet beim Schalten höherer Leistungen bzw. potentialgetrennten Kreisen seinen Einsatz – wie beispielsweise hydraulischen Ventilen oder Weitergabe von Not-Halt-Kreisen.



XS-322-2DO-RNOAC

Das sicherheitsbezogene Safety Relais-Ausgangsmodul ist geeignet für die Verwendung in Systemen mit optionalen Modulen und Interface-Variablen gemäß XN Safety Systemhandbuch, → "Weitere Nutzungsinformationen", Seite 447

Um in einer Applikation verwendet werden zu können, benötigt das XS-322-2DO-RNOAC mindestens auch ein XN Safety CPU-Modul XS-102-C00-000, welches über sichere Bustelegramme die zeit-korrekte Kommunikation mit den Sicherheitsmodulen regelt. Dazu gehört ferner

- die Abarbeitung der sicheren Applikation
- und
- die Verteilung der Konfigurationsdaten an entfernte Sicherheitsmodule.

Lieferumfang

- 1x XS-322-2DO-RNOAC
- 2x Steckverbinder PLUG-FKC-4L (2,5/4-ST-5,08)

Siehe auch



Handbuch Safety Relais-Ausgangsmodul XS-322-2DO-RNOAC MN050026

14. Safety-Module

14.8 Beschreibung Safety Inkrementalgeber-Modul XS-322-2INC

14.8 Beschreibung Safety Inkrementalgeber-Modul XS-322-2INC

Das Safety Inkrementalgeber-Modul XS-322-2INC stellt die Werte zweier Inkrementalgeber der sicheren CPU (XN Safety CPU-Modul) zur Verfügung.

Die zweikanalige Sicherheitsfunktion wird durch „Mithorchen“ der Inkremente auf den Inkrementalgeber-Schnittstellen und Verarbeitung in zwei Mikrocontrollern, dem sogenannten „Safetykern“, mit Kreuz-Kommunikation realisiert.

Die Inkrementalgeberfunktion wird durch eine zweikanalige Auswertung der Gebersignale mit zweikanaliger Fehlererkennung jeder Datenleitung und durch die Messungen der Geberspannungsspannungen und -ströme überwacht.

Die sichere Überwachung von Geschwindigkeit, Position, Richtung und Beschleunigung erfolgt in der sicheren Applikation in der Safety-CPU.

Das Modul erkennt verschiedene Fehlerarten wie z.B. Kabelbruch, Querschluss und Vertauschung der Eingangssignale.



XS-322-2INC

Die Sicherheitsfunktionen der Baugruppe erfüllen

- **bei zweikanaliger Verwendung**
die Anforderungen nach SIL 3, HFT 1 gemäß EN IEC 62061 und PL e / Kat. 4 gemäß EN ISO 13849.

sowie

- **bei einkanaliger Verwendung**
die Anforderungen nach SIL 3, HFT 1 gemäß EN IEC 62061 und PL d, Kat. 2 gemäß EN ISO 13849.

Lieferumfang

- 1x XS-322-2INC
- 2x Steckverbinder PLUG-FMC-8S (1,5/8-ST-3,5)
- 1x Steckverbinder PLUG-FMC-4S (1,5/4-ST-3,5)

Siehe auch



Handbuch Safety Inkrementalgeber-Modul XS-322-2INC MN050024

15. Fehler-Codes

Sollte keine der angeführten Maßnahmen Abhilfe schaffen, wenden Sie sich bitte an Eaton.

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(0)	Kein Fehler			Normaler Betrieb	
ERRVAL(1000)	Kein Fehler			Normaler Betrieb	
ERRVAL(1001)	Die zyklische Verarbeitung ist innerhalb der erlaubten Zeit nicht fertig geworden. Mögliche Ursache(n): ■ fehlerhafte Applikation	aktuelle Zykluszeit in Mikrosekunden (das Maximum aus den beiden Werten von Controller 1 und 2)	Limit der Zykluszeit in Mikrosekunden	Konfiguration muss geändert werden, z. B. Erhöhung der max. Cyclic Time bei der Safety-CPU Fehlerbehebung durch „Quit Error“.	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1002)	Bei einem Funktionsaufruf wurde ein Buffer als Parameter übergeben. Die Größe des Buffers ist allerdings zu klein. Mögliche Ursache(n): ■ Programmfehler	tatsächliche Buffer-Größe oder bei einem Wert $\geq 0x1000$ Kennung der Programmstelle des Fehlers (mögliche Werte 0x1000-0x100B)	benötigte Buffer-Größe	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1003)	Es wurde festgestellt, dass eine globale Variable überschrieben wurde. Mögliche Ursache(n): ■ Programmfehler ■ defektes RAM	betroffene Variable, die überschrieben wurde mögliche Werte: 0: aktueller Fehler-Code	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1004)	Es wird ein Programmcode ausgeführt, der laut Programmlogik nie ausgeführt werden dürfte. Mögliche Ursache(n): ■ Programmfehler ■ defekte CPU	dient zur Identifizierung der Code-Stelle, die ausgeführt wurde mögliche Werte: 0 bis 104	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1005)	Beim Zugriff auf ein Array liegt der Index außerhalb der Array-Grenzen. Mögliche Ursache(n): ■ Programmfehler	dient zur Identifizierung der Ursache des Fehlers mögliche Werte: 0 bis 57	beinhaltet den max. möglichen Array-Index	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1006)	Ungültiger Parameterwert bei einem Funktionsaufruf (NULL-Pointer). Mögliche Ursache(n): ■ Programmfehler	dient zur Identifizierung der Ursache des Fehlers mögliche Werte: 0 bis 28	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1007)	Beim RAM-Test, Algorithmus March C-wurde ein Lesefehler oder CRC-32-Fehler festgestellt. Hierbei handelt es sich um einen permanenten Fehler, der nicht quittiert werden kann. Mögliche Ursache(n): ■ Hardware-Fehler ■ RAM, Flash, Adress-, Datenleitung Programmfehler ■ RAM-Test wird bei nicht gesperrten Interrupts durchgeführt	betreffender Test-Schritt	betroffener Bereich im RAM	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1008)	Beim RAM-Test, Algorithmus Galpat wurde ein Lesefehler oder CRC-32-Fehler festgestellt. Hierbei handelt es sich um einen permanenten Fehler, der nicht quittiert werden kann. Mögliche Ursache(n): ■ Hardware-Fehler ■ RAM, Flash, Adress-, Datenleitung Programmfehler ■ RAM-Test wird bei nicht gesperrten Interrupts durchgeführt	betreffender Test-Schritt	betroffene Bereiche im RAM (CRC32-Prüfung und GalPat-Prüfung)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1009)	Die Konfiguration der benötigten Module stimmt nicht. Mögliche Ursache(n): ■ Tausch mehrerer Module ■ Module falsch konfiguriert ■ inkompatibler Interface Frame (falsche CRC)	0: es fehlt mehr als ein Modul	wenn Reason-Code 0 zwischen 0 und 1 ist: immer 0	- Wenn mehr als ein Modul getauscht wurde, muss die Applikation neu runtergespielt werden. - Bei falscher Konfiguration muss die Applikation neu runtergespielt werden. - Überprüfung, ob der Hardware-Aufbau mit der Applikation übereinstimmt; Applikation entsprechend umbauen. - Bei falscher Interface CRC muss das schreibende Interface neu exportiert werden und auf der lesenden Seite importiert werden. Applikation muss neu runtergespielt werden. - Fehlerbehebung durch Quit Error	Error-LED des Sicherheitsmoduls leuchtet rot
		1: es fehlt ein Modul, und es hat zuvor schon ein Modul-Update stattgefunden			
		2: an das fehlende Modul kann nichts verteilt werden (S-CPU)	wenn Reason-Code 0 2 ist: Sicherheitsnummer des entfernten Moduls		
		3: maximale Anzahl der Wiederholungsversuche der Konfigurationsverteilung oder Konfigurationsprüfung wurde überschritten	wenn Reason-Code 0 3 ist: Anzahl der Wiederholungsversuche		
		4: beim Versuch die Konfiguration zu versenden, wurde festgestellt, dass das Zielmodul im Fehlerzustand ist	wenn Reason-Code 0 4 ist Laufzeitzustand des entfernten Moduls		

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
		5: es fehlt mindestens ein lokales SDI-Modul	wenn Reason-Code 0 zwischen 5 und 10 ist: immer 0		
		6: es fehlt mindestens ein lokales Safety-Interface-Modul			
		7: es fehlt mindestens ein lokales STO-Modul			
		8: es fehlt mindestens ein lokales SRO-Modul			
		9: ein Modul (Master-Modul) hat versucht, die Konfiguration zu beschreiben, während die eigene Konfiguration nicht gelöscht und auf Master eingestellt war			
		10: ein Slave Modul hat festgestellt, dass die entfernten Module nicht richtig konfiguriert sind	wenn Reason-Code 0 zwischen 11 und 13 ist: Steckplatznummer des Moduls		
		11: lokales Eingangsmodul auf dem falschen Steckplatz (Zählung beginnt bei 1)			
		12: lokales Ausgangsmodul auf dem falschen Steckplatz (Zählung beginnt bei 1)			
		13: DINT-Eingangsmodul auf dem falschen Steckplatz (Zählung beginnt bei 1)			

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1010)	Der RAM-Speicher, der für Datenstrukturen mit einer konfigurationsabhängigen Größe verwendet wird, ist zu klein. Mögliche Ursache(n): - zu viele Module - zu große Applikation	14: Anzahl lokaler Eingänge falsch		- Größe der Applikation prüfen und entsprechend verkleinern Applikation erneut runterspielen, Fehlerbehebung durch Quit Error. - Anzahl der Module prüfen, und entsprechend Module entfernen, Fehlerbehebung durch Quit Error	Error-LED des Sicherheitsmoduls leuchtet rot
		15: Anzahl lokaler Ausgänge falsch			
		16: es wurde versucht ein lokales Modul anzusprechen, welches fehlt			
		17: ein Erweiterungsmodul konnte nicht konfiguriert werden (Zählung beginnt bei 1)	wenn Reason-Code 0 17 ist: Steckplatznummer des Moduls		
		18: zu viele Erweiterungsmodule angesteckt			
		19: zu wenige Erweiterungsmodule angesteckt			
		20: falsches Safety-Interface Modul angesteckt			
		0: Daten für I/O-Listen und entfernte Module	immer 0		
		1: Status der FSoE-Verbindung			
		2: FSoE-Buffer			
ERRVAL(1011)	Beim Verteilen der Konfigurationsdaten an ein entferntes Modul ist die max. Anzahl von Fehlern überschritten worden. Mögliche Ursache(n): - Übertragungsfehler - anderes Modul defekt	Fehler-Code des letzten Verteilungsversuchs	Returncode des zuletzt ausgeführten SSDO-Befehls	Die Verteilung mehrmals anstoßen Falls defekte Module: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1012)	Beim Exception-Handling ist ein Fehler aufgetreten. Mögliche Ursache(n): Software-Fehler	0: maximale Anzahl der verschachtelten TRY-CATCH-ETRY-Blöcke wurde überschritten 1: Ende eines TRY-CATCH-ETRY-Blocks ohne beginnendem TRY erreicht 2: Exception nicht im normalen Programm-Mode geworfen 3: kein TRY-Block zum Abfangen der Exception vorhanden	Anzahl der offenen TRY-CATCH-ETRY-Blöcke	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1013)	Nach dem Verteilen der Konfigurationsdaten an ein entferntes Modul wurde beim Rücklesen ein Unterschied festgestellt.	Modul-Index des betroffenen entfernten Moduls	immer 0	Applikation nochmals runterspielen und starten Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1014)	Der Interpretercode in den Konfigurationsdaten wurde nicht gefunden. Mögliche Ursache(n): - Fehlerhafte Konfiguration - Software-Fehler	immer 0	immer 0	Applikation nochmals runterspielen und starten Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1015)	In der Interpreterliste der Konfiguration wird ein FSB angegeben, zu dem kein FSB-Code existiert. Mögliche Ursache(n): - es wurde eine fehlerhafte Konfiguration eingegeben - Flash defekt	FSB-ID des nicht gefundenen FSB-Codes	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1016)	Beim ROM-Test in einem unveränderlichen Flash Bereich wurde ein Fehler festgestellt. Hierbei handelt es sich um einen permanenten Fehler, der nicht quitiert werden kann. Mögliche Ursache(n): ■ Flash defekt	Informationen über die Ursache des Fehlers	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1017)	Der aufgerufene FSB hat einen Fehler-Code zurückgegeben.	Fehler-Code des FSB (ist immer ungleich 0)	FSB-ID und Objekt-Index des betroffenen FSB	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1018)	Der entfernte sichere Eingang eines fehlenden optionalen Moduls (entferntes Modul oder Safety-Interface-Modul) hat das maximale Alter überschritten.	immer 0	immer 0	Es ist zu überprüfen, ob die Standardkommunikation zu dem benötigten Sicherheitsmodul intakt ist. Zudem ist sicherzustellen, dass sich das benötigte Sicherheitsmodul nicht im Fehlerzustand befindet. Außerdem kann es sein, dass die Maximum Transmission Time zu niedrig konfiguriert wurde. Diese Punkte sind zu überprüfen. Wurde der Fehler behoben, so kann mit einem Quit Error bzw. einem Neustart des Systems der Fehler aufgehoben werden.	Error-LED des Sicherheitsmoduls blinkt
ERRVAL(1019)	Nach dem Aufruf des FSB wurde im Applikationsspeicher ein unerlaubter Wert festgestellt. Mögliche Ursache(n): ■ fehlerhafter FSB	tatsächliches Muster am Ende des Objektspeichers	Soll-Muster am Ende des Objektspeichers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1020)	Der aufgerufene FSB hat die Zählervariable nicht inkrementiert. Mögliche Ursache(n): ■ fehlerhafter FSB	tatsächlicher Wert der Zählervariable	Soll-Wert der Zähler-variable	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1021)	Bei der Verwaltung des RAM-Speichers für Datenstrukturen mit einer konfigurationsabhängigen Größe wurde ein Fehler festgestellt. Mögliche Ursache(n): Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1022)	Die Antwort auf einen SSDO Request liegt noch nicht vor. Mögliche Ursache(n): - Laufzeit über den Bus - Verarbeitungszeit im Zielmodul zu groß	immer 0	immer 0	Zeit am Bus erhöhen, falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1023)	In einer SSDO Antwort wurde im Returncode-Feld ein Fehler angezeigt. Mögliche Ursache(n): Fehler im entfernten Modul	immer 0	immer 0	Entfernte Module auf Fehler überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1024)	In einer SSDO-Antwort sind weniger Daten enthalten als erwartet. Mögliche Ursache(n): Fehler im entfernten Modul	tatsächliche Länge der Daten in der SSDO-Antwort	Soll-Länge der Daten in der SSDO-Antwort	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1025)	Beim Test der Prüfsummen über die konstanten Elemente der Konfigurationstabellen im RAM wurde ein Fehler entdeckt. Mögliche Ursache(n): - Software-Fehler - RAM defekt	Identifizierung der Tabelle mögliche Werte: 0 bis 11	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1026)	Der Zugriff auf den Applikationsspeicher war nicht möglich. Mögliche Ursache(n): ■ Software-Fehler ■ fehlerhafte Anfrage vom XN Safety Designer	0	bei Reason-Code 0 = 0: In den beiden niederwertigen Bytes ist der ermittelte ungültige Zellen-Index enthalten, in den beiden höherwertigen Bytes steht die Anzahl vorhandener Zellen	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
		1	bei Reason-Code 0 = 1: ungültiger Zellen-Index		
		2	bei Reason-Code 0 = 2: ungültiger Adress-Offset		
		3	bei Reason-Code 0 = 3: in den beiden niederwertigen Bytes ist Adress-Offset enthalten, in den beiden höherwertigen Bytes steht die Länge;		
		4	bei Reason-Code 0 = 4: ungültige Länge (nicht durch 4 teilbar)		
ERRVAL(1027)	Bei einem Speicherbereich im Flash wurde ein ungültiger Blockheader erkannt. Mögliche Ursache(n): ■ Hardware-Fehler (Flash defekt) ■ Programmiervorgang wurde unterbrochen	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1028)	Bei einem Anmeldeversuch wurde ein ungültiges Passwort angegeben. Fehler wird im XN Safety Designer im Konsolenfenster angezeigt. Mögliche Ursache(n): ■ Tippfehler ■ Hackerangriff	immer 0	immer 1	Das Passwort muss erneut eingegeben werden.	Weiterhin normaler Betrieb
ERRVAL(1029)	In einem SSDO Frame ist ein ungültiger Parameter vorhanden. Mögliche Ursache(n): ■ Software-Fehler	0 1 2 3 4	immer 0 bei Reason-Code 0 = 4: die niederwertigen 2 Bytes enthalten die Größe des übergebenen Buffers, die höherwertigen 2 Bytes die max. Größe	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1030)	In einem SSDO Frame fehlt ein Parameter. Mögliche Ursache(n): ■ Software-Fehler	Wird nur in der Firmware für den Hardwaretest verwendet.		Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1031)	Interner Fehler bei der Verarbeitung der SSDOs. Mögliche Ursache(n): ■ Software-Fehler	Länge des zu schreibenden SSDOs	maximale Länge eines SSDOs	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1032)	Das SSDO Paket enthält ein ungültiges Befehlsbyte. Mögliche Ursache(n): ■ Software-Fehler	ungültiges Kommando	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1033)	Bei dem Versuch zur Passwortänderung wurde ein ungültiges neues Passwort angegeben. Mögliche Ursache(n): - Software-Fehler - fehlerhafter Safety Designer	immer 0	immer 0	Wird ein falsches Passwort eingegeben, so kommt im Safety Designer eine Warnung. Kommt der Fehler 33, so handelt es sich um einen internen Fehler des Betriebssystems. Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1034)	Das SSDO Paket enthält eine ungültige virtuelle Adresse. Mögliche Ursache(n): Software-Fehler	Identifizierung der Fehlerursache mögliche Werte: 0 bis 1	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1035)	Bei der Verarbeitung der SSDOs wurde festgestellt, dass ein interner Buffer zu klein für die angeforderte Datenmenge ist. Mögliche Ursache(n): - Software-Fehler - fehlerhafter Safety Designer	Identifizierung der Fehlerursache mögliche Werte: 0 bis 1	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1036)	Beim Selbsttest des Safety-Interface-Controllers wurde ein Fehler festgestellt. Mögliche Ursache(n): Hardware-Fehler	Identifizierung der Fehlerursache mögliche Werte: 0: Testnachricht konnte nicht versendet werden 1: es wurde keine Antwort auf die Testnachricht empfangen 2: Daten der empfangenen Testnachricht sind falsch	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1037)	Die remanenten Variablen sind inkonsistent. Mögliche Ursache(n): - Hardware-Fehler - Software-Fehler	0: im POST wurde festgestellt, dass die Checksumme der remanenten Variablen im EEPROM nicht korrekt ist 1: Fehler in der Konsistenz der remanenten Variablen während der zyklischen Verarbeitung festgestellt	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1038)	Es ist ein Watchdog Fehler aufgetreten Mögliche Ursache(n): Software-Fehler	Adresse des Codes	bei einer Exception außerhalb des FSB Aufrufs: - immer 0 bei einer Exception während des FSB Aufrufs: - FSB-ID in den unteren 2 Byte - Objekt-Index in den oberen 2 Byte	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1039)	Die Version des Bootloaders ist falsch Mögliche Ursache(n): falsche Firmware	vorhandene Version des Bootloaders	minimal benötigte Version des Bootloaders	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1040)	Beim Zugriff auf den internen I/O-Bus ist ein Fehler aufgetreten. Mögliche Ursache(n): ■ Hardware-Fehler ■ Modul wurde abgesteckt	0: Fehler beim Lesezugriff eines Bytes	bei Reason-Code 0 zwischen 0 und 1:	Wenn ein Modul abgesteckt wurde, das Modul anstecken und neu starten Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
		1: Fehler beim Lesezugriff eines 2 Byte Werts	Byte 0 und 1 sind die Adresse des Fehlers		
		2: Fehler beim Schreibzugriff eines Bytes	bei Reason-Code 0 zwischen 2 und 3:		
		3: Fehler beim Schreibzugriff eines 2 Byte Werts	Byte 0 und 1 sind die Adresse des Fehlers Byte 2 und 3 ist der zu schreibende Wert		
		4: unterschiedliche Werte der Controller für die Modul-ID			
		5: Zeitfehler beim Warten auf das Grant des SPI-Masters	bei Reason-Code 0 zwischen 5 und 8:		
		6: Zeitfehler beim Warten auf das Ready des SPI-Masters	Position des Moduls bei welchen der Fehler aufgetreten ist (beginnend bei 0)		
		7: falsche Länge des SPI-Flash			
		8: falsche CRC des SPI-Flash			
		9: unterschiedliche Sicherheitsnummer in beiden Controllern			
		10: keine eindeutigen Sicherheitsnummern in den Erweiterungsmodulen	bei Reason-Code 0 = 10: Wert der Sicherheitsnummer, welche doppelt ist		

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
		11: beim Ermitteln der Seriennummern liefern beide Controller unterschiedliche Werte	bei Reason-Code 0 = 11: Position des Moduls bei welchen der Fehler aufgetreten ist (beginnend bei 0)		
		12: unterschiedliche Anzahl von Erweiterungsmodulen in beiden Controllern	bei Reason-Code 0 = 12: Byte 0: Anzahl der erkannten I/O-Module am Controller Byte 1: Anzahl der erkannten I/O-Module am anderen Controller		
		13: fehlende Sicherheitsnummer in einem Erweiterungsmodul	bei Reason-Code 0 = 13: Position des Moduls bei welchen der Fehler aufgetreten ist (beginnend bei 0)		
ERRVAL(1041)	Die Werte der zurückgelesenen Ausgänge, welche bereits im Fehler sind, sind nicht auf 0. Mögliche Ursache(n): - Hardware-Fehler (z.B. Ausgangstransistor defekt) - externer Fehler (z.B. Kurzschluss) - Software-Fehler	Ausgänge, welche nicht enabled sind (Bit 0 = 1. Ausgang)	Istwert (Bit 0 = 1. Ausgang)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1042)	Beim Test des OUTRESET-Signals sind nicht alle Ausgänge auf 0 gegangen. Mögliche Ursache(n): - Hardware-Fehler (z.B. Ausgangstransistor defekt) - externer Fehler (z.B. Kurzschluss) - Software-Fehler	getestete Ausgänge (Bit 0 = 1. Ausgang)	Istwert (Bit 0 = 1. Ausgang)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1043)	Ein lokales Erweiterungsmodul ist im Fehlerzustand. Mögliche Ursache(n): ■ Fehler im Erweiterungsmodul	0: Fehler des Erweiterungsmoduls	Reason-Code 0 des Erweiterungsmoduls	Erweiterungsmodul tauschen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
		1: 0 basierende Position des Erweiterungsmoduls			
ERRVAL(1044)	Bei einem lokalen Erweiterungsmodul ist in der Init-Phase ein Fehler aufgetreten. Mögliche Ursache(n): ■ Fehler im Erweiterungsmodul ■ Falsche Konfiguration	0: Stromwerte des INC-Moduls können nicht aus dem Modul ausgelesen werden	0 basierende Position des Erweiterungsmoduls	Stromwerte prüfen, Safety-CPU auf den Master-Modus umstellen; INC-Modul tauschen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
		1: Es sind keine Stromwerte verfügbar, die ins INC-Modul geschrieben werden sollten			
		2: Es ist ein INC-Modul auf einer Slave-Safety-CPU vorhanden			
		3: Es ist ein INC-Modul mit einer nicht unterstützten Versionsnummer vorhanden			
ERRVAL(1045)	Das zum Eingang oder Ausgang gehörende Erweiterungsmodul ist nicht vorhanden. Mögliche Ursache(n): ■ ein optionales Modul ist nicht angeschlossen	0: digitaler Ausgang	immer 0	Optionale Module prüfen	Error-LED des Sicherheitsmoduls leuchtet rot
		1: digitaler Eingang			
		2: Inkrementalgeber-Modul			
ERRVAL(1046)	Es sind zu wenige Ressourcen für die angeforderte Funktion vorhanden Mögliche Ursache(n): ■ Konfiguration fehlerhaft	Immer 0	immer 0	Konfiguration überprüfen. Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1047)	Ein Fehler mit der Parameterliste ist aufgetreten. Mögliche Ursache(n): - Hardware-Fehler (Flash defekt) - Software-Fehler	dient zur Identifizierung der Fehlerursache mögliche Werte: 0 bis 8	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls
ERRVAL(1048)	Beim OUTRESET Watchdog-Test ist ein Fehler aufgetreten. Mögliche Ursache(n): Hardware-Fehler	Bit-Maske zur Identifizierung der Testschritte, bei denen der Fehler entdeckt wurde	aktueller Testschritt	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls
ERRVAL(1049)	Bei der Prüfung des LOG-Speichers im POST wurde festgestellt, dass beide Sektoren voll sind. Mögliche Ursache(n): - Hardware-Fehler - Programmfehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1050)	Bei der Prüfung des LOG-Speichers im POST wurde festgestellt, dass beide Sektoren Daten enthalten aber nicht voll sind. Mögliche Ursache(n): - Hardware-Fehler - Programmfehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1051)	Die Meldungen im LOG-Speicher sind nicht zusammenhängend (Löcher im LOG-Speicher). Mögliche Ursache(n): Programmfehler	Startadresse des ersten gefundenen leeren Eintrags im LOG-Speicher	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1052)	Die Information über die gespeicherten Sicherheitsnummern der entfernten Module ist inkonsistent. Mögliche Ursache(n): ■ Programmfehler	immer 0xFFFFFFFF	immer 0xFFFFFFFF	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1053)	Der SSDO Befehl kann im aktuellen Zustand nicht ausgeführt werden. Mögliche Ursache(n): ■ Modul führt den Bootcode aus	immer 0	immer 0	Neu starten Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1054)	Es wurde versucht auf einen Speicherbereich zuzugreifen (Flash oder SD-Karte) auf dem kein Zugriff möglich oder erlaubt ist. Mögliche Ursache(n): ■ Programmfehler ■ Fehler im Safety Designer	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1055)	Beim Flash programmieren ist ein Fehler aufgetreten. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1056)	Beim Status-Wechsel im STM Modul wurde festgestellt, dass bereits ein Wechsel aktiv ist. Mögliche Ursache(n): ■ Programmfehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1057)	Beim Aktivieren des Timers im STM Modul wurde festgestellt, dass bereits ein Timer aktiv ist. Mögliche Ursache(n): ■ Programmfehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1058)	Es sind bei einer Abfrage im STM Modul nicht alle benötigten Flags gesetzt. Mögliche Ursache(n): ■ Programmfehler	Flags, die gesetzt sein sollten	Flags, die aktuell gesetzt sind	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1059)	Der Diagnosecode bzw. der Login-Status am Funktionsblock HGW-Login hat einen falschen Wert Mögliche Ursache(n): ■ Fehler im Safety Designer ■ Programmierfehler	dient zur Identifizierung der Fehlerursache 1: Diagnosecode hat einen ungültigen Wert angenommen 2: Login-Status hat einen ungültigen Wert angenommen	Wert vom Diagnose-Code bzw. Login-Status	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1060)	Es ist das Flag zum Deaktivieren der Safety Interface Kommunikation (HBG/SIB061) gesetzt, das Deaktivieren ist aber nicht erlaubt. Mögliche Ursache(n): ■ Programmierfehler	1: es ist mindestens ein HBG/SIB061 konfiguriert 2: es besteht eine Verbindung zu einem HBG/SIB061	immer 0	Überprüfung der Konfiguration; entweder Flag auf false stellen oder HBG/SIB061 entfernen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1061)	Beim Programmieren des externen Flashs ist ein Fehler aufgetreten. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1062)	Einer der ADC Kanäle für die Spannungs- und Temperaturüberwachung hat keinen gültigen Messwert geliefert.	dient zur Identifizierung der Fehlerursache 0: Temperatur 1: 1V8 2: 5V0 4-6: ADC Konfiguration fehlgeschlagen 7: ADC hat trotz Retry keinen Messwert geliefert	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1063)	Einer der ADC Kanäle für die Spannungs- und Temperaturüberwachung hat den gültigen Wertebereich verlassen.	dient zur Identifizierung der Fehlerursache 0: Temperatur 1: 1V8 2: 5V0 3: 24V	gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1064)	Beim zyklischen Test des OUTRESET-Watchdogs ist ein Fehler aufgetreten. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache 1: Test wurde nicht beendet 2: Reaktion auf Aussetzen des Toggle-Signals nicht rechtzeitig aufgetreten 3: Watchdog Ausgang war bereits beim Hochfahren aktiv	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1065)	Bei einem I2C-Zugriff (z.B. EEPROM/EERAM) ist ein Fehler aufgetreten. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1066)	Nicht verwendet				
ERRVAL(1067)	Die FSoE-Verbindung zwischen HGW Modul und Safety-CPU ist unkontrolliert abgerissen. Mögliche Ursache(n): ■ Verbindung gestört ■ Entfernung zwischen den Modulen zu groß	immer 0	immer 0	Fehlerquittierung und erneute Anmeldung	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1068)	Ungültiger Parameterwert bei einem Funktionsaufruf. Mögliche Ursache(n): ■ Programmfehler	dient zur Identifizierung der Fehlerursache mögliche Werte: 0 bis 95	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1069)	Der Platz im SPDO für die zu übertragenden Sicherheitswerte ist zu klein. Mögliche Ursache(n): ■ Software-Fehler	Index des höchstwertigen Bytes der zu übertragenden Sicherheitswerte.	maximale Größe der zu übertragenden Sicherheitswerte.	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1070)	Kreuzkommunikation funktioniert nicht (Übertragungsfehler). Mögliche Ursache(n): ■ Hardware-Fehler	1: Fehler beim Selbsttest im Post 2: Fehler im UART erkannt	Fehlerflags	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1071)	Nicht verwendet				
ERRVAL(1072)	Es wurde versucht, eine Division durch 0 auszuführen. Mögliche Ursache(n): ■ Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1073)	Über die Kreuzkommunikation wurde ein unerwarteter Wert empfangen Mögliche Ursache(n): ■ Software-Fehler	empfangener Wert	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1074)	Die logische Programmlaufüberwachung hat festgestellt, dass ein im Op/OpTemp Modus auszuführender Programmteil nicht ausgeführt wurde. Mögliche Ursache(n): ■ Software-Fehler	ID des Programmteils	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1075)	Die in den beiden Controllern eingelesenen Werte für die lokalen Eingänge weisen für eine unzulässig lange Zeit unterschiedliche Werte auf. Mögliche Ursache(n): - Hardware-Fehler - Eingangsschaltung	Bitmaske mit den Eingängen des eigenen Controllers	Bitmaske mit den Eingängen des anderen Controllers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1076)	Kreuzkommunikation funktioniert nicht (Timeout). Mögliche Ursache(n): - Hardware-Fehler - Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1077)	Der Inhalt einer STDO Antwort ist in beiden Controllern unterschiedlich. Mögliche Ursache(n): Software-Fehler	0 oder 1, je nach Art des Fehlers	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1078)	Die Laufzeitüberwachung wurde unerlaubterweise in einem Operational-Laufzeitzustand deaktiviert. Mögliche Ursache(n): Software-Fehler	ID des Aufrufers, von dem die Laufzeitüberwachung deaktiviert wurde.	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1079)- ERRVAL(1081)	Kreuzkommunikation funktioniert nicht (unerwarteter Wert empfangen). Mögliche Ursache(n): - Hardware-Fehler - Software-Fehler	über die Kreuzkommunikation empfangener Wert	erwarteter Wert (= Soll-Wert)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1082)	Das berechnete Ausgangsabbild ist in den beiden Controllern unterschiedlich. Mögliche Ursache(n): - Software-Fehler - Hardware-Fehler	Index der Zelle im Applikationsspeicher, deren Inhalt nicht mit der des anderen Controllers übereinstimmt.	Inhalt der im Reason-Code 0 angeführten Zelle	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1083)	Der andere Controller ist bei der Kreuzkommunikation nicht mehr synchron. Es handelt sich hier noch um keinen Fehler, der den sicheren Zustand herbeiführt. Als Reaktion auf diesen Fehler synchronisieren sich die Controller neu, tauschen Informationen aus und erst dann wird entschieden, ob der sichere Zustand herbeigeführt wird. Mögliche Ursache(n): ein Controller hat einen Fehler festgestellt, der andere (noch) nicht	immer 0	immer 0	Fehler quittieren, System ev. neu starten Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1084)	Der Inhalt der SD-Karte konnte nicht geladen werden, weil die Konfiguration im Flash nicht vorhanden und auch nicht gelöscht ist. Mögliche Ursache(n): Anwenderfehler	immer 0	immer 0	Speicher der Safety-CPU löschen und nochmals versuchen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1085)	Die SD-Karte konnte nicht initialisiert werden. Mögliche Ursache(n): Hardware-Fehler	immer 0	immer 0	Andere SD-Karte verwenden Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1086)	Der Inhalt auf der SD-Karte entspricht nicht dem erwarteten Format. Mögliche Ursache(n): - Hardware-Fehler - Software-Fehler - Anwenderfehler	immer 0	immer 0	SD-Karte nochmals bespielen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1087)	Die Konfigurationsdaten auf der SD-Karte stimmen nicht mit denen im Flash überein. Mögliche Ursache(n): ■ Anwenderfehler	CRC der Konfigurationsdaten auf der SD-Karte	CRC der Konfigurationsdaten im Flash	Entweder ist die Konfiguration im Flash zu löschen, falls die Konfiguration der SD-Karte übernommen werden soll. Soll jedoch die Konfiguration von der SD-Karte übernommen werden, so ist die Konfiguration auf dem Flash zu löschen und das System bei eingesteckter SD-Karte neu zu starten.	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1088)	Die SD-Karte wurde eingesteckt, während das Modul eingeschaltet war und sich nicht im Service Mode befand. Beim Start der Applikation ist ein Dongle angesteckt, der Servicemode kann nicht verlassen werden. Mögliche Ursache(n): ■ Anwenderfehler ■ Dongle eingesteckt	immer 0	immer 0	SD-Karte wieder entnehmen, Fehler quittieren und in den Service Mode wechseln, dann SD-Karte einstecken. Dongle entfernen, Fehler quittieren und erneut starten.	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1089)	Interner Fehler in einer Ablaufabelle.	dient zur Identifizierung der Fehlerursache Mögliche Werte: 0 bis 1	ungültiger Status bzw. Index	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1090)	Der gemessene Wert des Referenzclocks vom FPGA (REFCLK) ist zu klein. Hierbei handelt es sich um einen permanenten Fehler, der nicht quittiert werden kann. Mögliche Ursache(n): ■ Hardware-Fehler ■ CPU ■ FPGA	Anzahl gemessener Flanken	minimale Anzahl der zu messenden Flanken	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1091)	Der gemessene Wert des Referenzclocks vom FPGA (REFCLK) ist zu groß. Hierbei handelt es sich um einen permanenten Fehler, der nicht quittiert werden kann. Mögliche Ursache(n): - Hardware-Fehler - CPU - FPGA	Anzahl gemessener Flanken	maximale Anzahl der zu messenden Flanken	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1092)	Beim Warten auf die SSDO Response ist eine Zeitüberschreitung aufgetreten. Mögliche Ursache(n): - Laufzeit über den Bus - Verarbeitungszeit im Zielmodul	immer 0	immer 0	Laufzeiten verringern	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1093)	Der andere Controller hat einen Fehler festgestellt. Mögliche Ursache(n): Ursache gemäß Fehler-Code des anderen Controllers	Fehler-Code des anderen Controllers	immer 0	Fehler-Code des anderen Controllers prüfen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1094)	Der Konfigurationszustand des anderen Controllers stimmt mit dem eigenen nicht überein. Mögliche Ursache(n): Modul wurde während des Konfigurierens abgeschaltet	Konfigurationsstatus des eigenen Controllers	Konfigurationsstatus des anderen Controllers	Modul neu konfigurieren	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1095)	Eine Konstante im Applikationsspeicher wurde verändert und stimmt nicht mehr mit dem im Flash gespeicherten Wert überein. Mögliche Ursache(n): - Hardware-Fehler (defektes Flash, defektes RAM) - Software-Fehler	Index der Speicherzelle der fehlerhaften Konstante im Applikationsspeicher	Index der Konstante im Flash	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1096)- ERRVAL(1097)	Nicht verwendet				
ERRVAL(1098)	Die Reaktion auf den internen Test konnte nicht ermittelt werden. Mögliche Ursache(n): ■ Software-Fehler	Bit-Maske der digitalen Eingänge	0: Fehler bei einem lokalen Eingang 1: Fehler bei einem Safety-Interface-Modul Eingang	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1099)	Beim Eingangstest (interner Test) ist ein Fehler festgestellt worden. Mögliche Ursache(n): ■ Hardware-Fehler	Zustände der digitalen Eingänge	Bit-Maske der digitalen Eingänge	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1100)	Während des externen Tests ist ein interner Test aktiv. Mögliche Ursache(n): ■ Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1101)	Die Reaktion auf den externen Test konnte nicht ermittelt werden. Mögliche Ursache(n): ■ Software-Fehler	Bit-Maske der digitalen Eingänge	0: Fehler bei einem lokalen Eingang 1: Fehler bei einem Safety-Interface-Modul Eingang	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1102)	Externer Test: Bei einem am Takt angeschlossenen Eingang ist der Taktimpuls nicht erkannt worden. Mögliche Ursache(n): ■ Querschluss	immer 0	immer 0	Verdrahtung des Querschlusses mit der Applikation vergleichen	Error-LED des Sicherheitsmoduls blinkt
ERRVAL(1103)	Externer Test: Bei einem am Takt angeschlossenen Eingang ist der Taktimpuls des anderen Taktausgangs erkannt worden. Mögliche Ursache(n): ■ Querschluss	immer 0	immer 0	Verdrahtung des Querschlusses mit der Applikation vergleichen	Error-LED des Sicherheitsmoduls blinkt

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1104)	Der Lo-Resolution Timer arbeitet nicht richtig (gemessene Werte sind zu klein). Mögliche Ursache(n): Hardware-Fehler	gemessener Wert des Lo-Resolution Timers	unteres Limit für die Messung	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1105)	Der Lo-Resolution Timer arbeitet nicht richtig (gemessene Werte sind zu groß). Mögliche Ursache(n): Hardware-Fehler	gemessener Wert des Lo-Resolution Timers	oberes Limit für die Messung	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1106)	Der Millisekunden-Timer arbeitet nicht richtig (gemessene Werte sind zu klein). Mögliche Ursache(n): - Hardware-Fehler - Software-Fehler	gemessener Wert des Millisekunden-Timers	unteres Limit für die Messung	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1107)	Der Millisekunden-Timer arbeitet nicht richtig (gemessene Werte sind zu groß). Mögliche Ursache(n): - Hardware-Fehler - Software-Fehler	gemessener Wert des Millisekunden-Timers	oberes Limit für die Messung	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1108)	Nicht verwendet				
ERRVAL(1109)	Das Format des Info-Sektors im Flash stimmt nicht. Mögliche Ursache(n): - Modul wurde noch nie initialisiert (mit dem Clear-All Befehl) - Hardware-Fehler im Flash	immer 0	immer 0	Modul konfigurieren Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1110)- ERRVAL(1114)	Das Format der Konfigurationsdaten im Flash stimmt nicht. Mögliche Ursache(n): - Modul wurde falsch konfiguriert - Hardware-Fehler im Flash	Detailinfo zur Fehlerursache	Detailinfo zur Fehlerursache	Modul konfigurieren Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1115)	Das Format der Konfigurationsdaten im Flash stimmt nicht. Mögliche Ursache(n): - Modul wurde falsch konfiguriert - in den Konfigurationsdaten ist ein nicht unterstütztes Erweiterungsmodul angegeben Hardware-Fehler im Flash	1 nicht unterstütztes Erweiterungsmodul 2-39 Fehlerinformation 40 mehr globale Flags in der Konfiguration als erlaubt 41 mehr als 5 unsicher CAN Empfangsvariablen	bei Reason-Code 0 = 1: Typ des Moduls bei Reason-Code 0 = 2-32: Detailinformation	Modul konfigurieren, prüfen ob ein nicht unterstütztes Erweiterungsmodul angesteckt ist und ggf. eine Safety-CPU mit aktueller Firmware verwenden; Anzahl der CAN Variablen prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1116)- ERRVAL(1128)	Das Format der Konfigurationsdaten im Flash stimmt nicht. Mögliche Ursache(n): - Modul wurde falsch konfiguriert - Hardware-Fehler im Flash	Detailinfo zur Fehlerursache	Detailinfo zur Fehlerursache	Modul konfigurieren Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1129)	Bei der Kommunikation mit dem FPGA ist ein Fehler aufgetreten. Mögliche Ursache(n): Hardware-Fehler (FPGA)	Wert, der genauere Informationen über den Fehler liefert 0: SPI Zugriff 0 fehlgeschlagen 1: SPI Zugriff 1 fehlgeschlagen 2: DMA funktioniert nicht mehr 3: ungültige FPGA-Version 4: FPGA nicht bereit	Wert, der genauere Informationen über den Fehler liefert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1130)	Bei einem Funktionsaufruf wurde ein Buffer als Parameter übergeben. Die Größe des Buffers ist allerdings zu groß. Mögliche Ursache(n): ■ Programmfehler	tatsächliche Größe des Buffers	maximale Größe des Buffers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1131)	Für den SSDO Befehl wurde ein falscher Frametyp verwendet. Mögliche Ursache(n): ■ Fehler im Safety-Tool	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1132)	Bei den Modul-Sicherheitsnummern der entfernten Module ist ein Wert doppelt vorhanden. Mögliche Ursache(n): ■ Beim Ermitteln der Modul-Sicherheitsnummer anhand des Topologiepfads ist ein Fehler aufgetreten	Reason-Code 0: < 1024: Index des Moduls in der Tabelle der entfernten Module Reason-Code 0: = 1024: Eine Sicherheitsnummer eines entfernten Moduls ist gleich wie die eigene Sicherheitsnummer Reason-Code 0: = 1025: Bei Abfrage der eigenen Sicherheitsnummer wurde nicht die eigene Sicherheitsnummer empfangen	doppelt vorhandene Modul-Sicherheitsnummer	- Applikation nochmals starten - Alle Sicherheitsnummern auf Eindeutigkeit prüfen	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1133)	Der Microcontroller befindet sich nicht im korrekten Prozessor-Modus. Entweder sind die Flags für den Prozessor-Modus oder für die Aktivierung der Interrupts im Programm Status-Register nicht korrekt gesetzt, oder es wird der falsche Stack verwendet. Mögliche Ursache(n): Der Wechsel des Prozessor-Modus funktioniert nicht korrekt (aufgrund Hardware- oder Software-Fehler)	gesetzte Flags im Programm Status-Register oder der Wert des Stack-Pointers	Soll-Wert für die Flags im Programm Status-Register oder untere bzw. obere Grenze des zu verwendenden Stack-Bereichs	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1134)	Das berechnete Ausgangsabbild ist in den beiden Controllern unterschiedlich.	Zustände der lokalen Ausgänge des eigenen Controllers	Zustände der lokalen Ausgänge des anderen Controllers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1135)	Die Versorgungsspannung eines Erweiterungsmoduls ist nicht in Ordnung. Mögliche Ursache(n): - Externe Spannungsversorgung nicht angeschlossen - Hardware-Fehler	0: wenn die Versorgungsspannung des CPU Moduls nicht in Ordnung ist 1: wenn bei der Watchdogprüfung ein Fehler festgestellt wurde, welcher wegen einer fehlenden Versorgungsspannung aufgetreten ist	Modulposition (0 = 1. Erweiterungsmodul)	Es ist sicherzustellen, dass die externe Spannungsversorgung der Taktausgänge bei Sicherheitsmodulen mit digitalen Eingängen und die Versorgungsspannung der Ausgänge für Sicherheitsmodule mit digitalen Ausgängen korrekt ist. Nach Anlegen der richtigen Spannung kann der Fehler mit „Quit Error“ beziehungsweise durch einen Wiederanlauf behoben werden.	Error-LED des Sicherheitsmoduls leuchtet rot DCOK-LED leuchtet nicht.

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1136)	Die boolsche Darstellung eines Ausgangs hat einen ungültigen Wert. Mögliche Ursache(n): - Software-Fehler (Speicher wurde an unerlaubter Stelle überschrieben) - Hardware-Fehler (RAM defekt)	0x00000000 bedeutet, dass der Fehler beim Dekodieren einer Zelle im Applikationsspeicher entdeckt wurde. 0xFFFFFFFF bedeutet, dass der Fehler beim Kodieren des SAFE-BOOL-Wertes für den SPDO-Frame entdeckt wurde.	ungültiger Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1137)	Die Werte der zurückgelesenen Ausgänge stimmen mit den Werten der Sollwerte der Ausgänge nicht überein. Mögliche Ursache(n): - Hardware-Fehler (z.B. Ausgangstransistor defekt) - externer Fehler (z.B. Kurzschluss, Versorgungsspannung der Ausgänge fehlt) - Software-Fehler	Sollwerte (Bit 0 = 1. Ausgang)	Istwerte (Bit 0 = 1. Ausgang)	Spannungsversorgung prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1138)	Beim Watchdog-Test ist ein Fehler aufgetreten. Mögliche Ursache(n): Hardware-Fehler	Bit-Maske zur Identifizierung der Testschritte, bei denen der Fehler entdeckt wurde	aktueller Testschritt	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1139)- ERRVAL(1141)	In der Konfiguration sind beim COPY Befehl ungültige Parameter angegeben. Mögliche Ursache(n): - Es wurde eine fehlerhafte Konfiguration eingegeben - Flash defekt	Index der Quelle bzw. des Ziels	maximale Anzahl vorhandener Zellen bzw. unteres oder oberes Index-Limit	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1142)	Nach dem Einschalten der oberen Ausgangshalbleiter (HSS) ist mind. ein Ausgang auf 1 gegangen, obwohl keiner der unteren Ausgangshalbleiter (FET) eingeschaltet war. Hierbei handelt es sich um einen permanenten Fehler, der nicht quitiert werden kann. Mögliche Ursache(n): Ausgangshalbleiter (FET) defekt	Sollwert (Bit 0 = 1. Ausgang) immer 0	Istwert, betroffene Ausgänge (Bit 0 = 1. Ausgang)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1143)	Nach dem Abschalten des HSS Ausgangshalbleiters ist keiner der zuvor gesetzten Ausgänge auf 0 gegangen. Mögliche Ursache(n): - Ausgangshalbleiter (HSS) defekt - unzulässig große kapazitive Last	Werte der Ausgänge nach dem Abschalten des HSS (Bit 0 = 1. Ausgang)	zuvor gesetzte Ausgänge (Bit 0 = 1. Ausgang)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls blinkt
ERRVAL(1144)	Nach dem Abschalten des FET Ausgangshalbleiters ist der zugehörige Ausgang nicht auf 0 gegangen. Mögliche Ursache(n): - Ausgangshalbleiter (FET) defekt - unzulässig große kapazitive Last	betroffener Ausgang (Bit 0 = 1. Ausgang)	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls blinkt
ERRVAL(1145)	Nach dem Abschalten des FET Ausgangshalbleiters hat ein Ausgang der nicht über diesen FET angeschlossen ist, seinen Wert geändert. Mögliche Ursache(n): interner Querschuss im Ausgangskreis	Status der Ausgänge (Bit 0 = 1. Ausgang)	Soll-Status der Ausgänge (Bit 0 = 1. Ausgang)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls blinkt

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1146)	Die zyklische Verarbeitung ist innerhalb der erlaubten Zeit nicht fertig geworden. Mögliche Ursache(n): fehlerhafte Applikation	Caller-ID mit der zuletzt die Runtime-Triggerung erfolgte	Code-Adresse, an der nach Beendigung der Interrupt-Funktion fortgesetzt wird	Zykluszeit der Safety-CPU erhöhen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1147)	Die logische Programmlaufüberwachung hat festgestellt, dass ein Programmteil unzulässig lange nicht mehr aufgerufen wurde. Mögliche Ursache(n): Software-Fehler	Safety-CPU XS-102: <1000 Index des Programmteils ≥1000 ID des Programmteils, wo der Fehler festgestellt wurde 1000-1009	Safety-CPU XS-102: Reason-Code 0: < 1000: einzuhaltende Zeit des Programmteils	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1148)	Die Rücklesung der HS-Transistoren ist nicht in Ordnung. Mögliche Ursache(n): - Hardware-Fehler - keine Versorgungsspannung	Position des Erweiterungsmoduls (0 = 1. Modul)	Wert des HS-Signals	Spannungsversorgung prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1149)	Ein SSI Erweiterungsmodul hat bei einem Eingang einen fehlerhaft kodierten Wert erkannt. Mögliche Ursache(n): Hardware-Fehler	Modulposition und fehlerhaft kodierter Wert	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1150)	Der Eingangswert ist unzulässig lange nicht mehr aktualisiert worden. Mögliche Ursache(n): - Fehler in der Eingangshardware - ständige Änderung des Eingangssignals	betroffener Eingang (Bit 0 = 1. Eingang)	maximales Alter	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1151)	Die reservierten Stackbereiche wurden überschrieben (untere Grenze). Mögliche Ursache(n): - Software-Fehler - Stack-Reserve zu klein dimensioniert	ID des betroffenen Stackbereichs	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1152)	Die reservierten Stack-Bereiche wurden überschrieben (obere Grenze). Mögliche Ursache(n): - Software-Fehler - Stack-Reserve zu klein dimensioniert	ID des betroffenen Stack-Bereichs	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1153)	Das berechnete Ausgangsabbild ist in den Controllern unterschiedlich. Mögliche Ursache(n): Software-Fehler	CRC des Sende SPDO Frames und des App-Speichers des eigenen Controllers	CRC des Sende SPDO Frames und des App-Speichers des anderen Controllers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1154)	Die Information über die gespeicherten Sicherheitsnummern der entfernten Module ist inkonsistent. Mögliche Ursache(n): Programmfehler	Index des gesuchten Moduls	Anzahl der Elemente in der Tabelle der entfernten Module	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1155)	Der CPU-Test hat einen Fehler festgestellt. Mögliche Ursache(n): CPU-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1156)	Es wurde vom Geber (Inkrementalgeber) kein Geberwert eingelesen, da vom CPLD kein Einlesevorgang gestartet wurde. Mögliche Ursache(n): - Standard-Applikation wurde nicht gestartet - Hardware-Klasse für das Inkrementalgeber-Modul ist nicht platziert.	immer 0	immer 0	Applikation neu starten Überprüfung, ob der Hardwareaufbau mit der Applikation übereinstimmt; Applikation entsprechend umbauen.	

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1157)	Der maximale Jitter zwischen zwei Einlesevorgängen des Geberwertes wurde nicht eingehalten. Mögliche Ursache(n): ■ Ungültige Konfiguration durch die Hardware-Klassen ■ Hardware-Fehler	immer 0	immer 0	Überprüfung der Versionen von FPGA und Hardware-Klasse	
ERRVAL(1158)	Das CPLD für den Geber meldete einen Fehler, wodurch kein Geberwert eingelesen werden konnte. Mögliche Ursache(n): ■ Einstellungen für den Geber sind nicht korrekt ■ Geber ist nicht angeschlossen ■ Geber wird nicht unterstützt ■ Geber ist defekt	immer 0	immer 0	Überprüfung, ob die Parameter für die Geber aus deren Datenblatt korrekt in der Standard-Applikation, sowie in der sicheren Applikation übernommen wurden Überprüfung der Kabelverbindung zu den Gebern Überprüfen, ob der verwendete Geber gemäß Datenblatt unterstützt wird Prüfen, ob ein defekter Geber vorliegt	
ERRVAL(1159)	Die vom CPLD eingelesenen Daten sind ungültig. Mögliche Ursache(n): ■ Einstellungen für den Geber sind nicht korrekt ■ Die Einstellungen für den Geber wurden zur Laufzeit geändert. ■ Geber ist defekt	immer 0	immer 0	Überprüfung, ob die Parameter für die Geber aus deren Datenblatt korrekt in der Standard-Applikation, sowie in der sicheren Applikation übernommen wurden Überprüfung der Kabelverbindung zu den Gebern Prüfen, ob ein defekter Geber vorliegt	
ERRVAL(1160)	Die Versorgungsspannung des Gebers ist nicht in Ordnung. Mögliche Ursache(n): ■ Versorgungsspannung für die Geber ist nicht vorhanden ■ Hardware-Fehler	immer 0	immer 0	Überprüfung der Verdrahtung der Versorgungsspannung für die Geber. Die Fehlermeldung wird bei Unter- bzw. Überspannung der extern anzuschließenden 24 V Gebersversorgung am SNC 021/SSI 021(X3) generiert. Die Versorgungsspannungstoleranzen der Versorgung sind der HW-Dokumentation des SNC 021/SSI 021 zu entnehmen.	

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1161)	Es liegt ein Fehler in der Konfiguration für das Modul vor. Mögliche Ursache(n): Geber angeschlossen, obwohl dieser laut Safety-Applikation nicht verwendet wird	immer 0	immer 0	Prüfen, ob für die angeschlossenen Geber in der Safety-Applikation die Parameter korrekt gesetzt wurden.	
ERRVAL(1162)	Beim Test der Signale des Inkrementalgebers wurde durch den ersten RS422-Treiber ein Signalfehler detektiert. Mögliche Ursache(n): - Querschluss der Signalleitungen - Drahtbruch - Kurzschluss	immer 0	immer 0	Verdrahtung des Gebers, sowie den Geber überprüfen	Safe DATA OK LED des betreffenden Geber-Kanals leuchtet nicht Geberstatus LED des betreffenden Gebers blinkt
ERRVAL(1163)	Beim Test der Signale des Inkrementalgebers wurde durch den zweiten RS422-Treiber ein Signalfehler detektiert. Mögliche Ursache(n): - Querschluss der Signalleitungen - Drahtbruch - Kurzschluss	immer 0	immer 0	Verdrahtung des Gebers, sowie den Geber überprüfen	Safe DATA OK LED des betreffenden Geber-Kanals leuchtet nicht Geberstatus LED des betreffenden Gebers blinkt
ERRVAL(1164)	Beim Test der Signale der Z-Spur des Inkrementalgebers wurde ein Signalfehler erkannt. Mögliche Ursache(n): - Querschluss der Signalleitungen - Drahtbruch - Kurzschluss	immer 0	immer 0	Verdrahtung des Gebers, sowie den Geber überprüfen	Safe DATA OK LED des betreffenden Geber-Kanals leuchtet nicht Geberstatus LED des betreffenden Gebers blinkt

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1165)	Beim Test der Signale der Z-Spur des Inkrementalgebers wurde erkannt, dass die Signalleitungen Z+ und Z- vertauscht wurden. Mögliche Ursache(n): ■ Vertausch der Signalleitungen Z+ und Z-	immer 0	immer 0	Verdrahtung des Gebers überprüfen	Safe DATA OK LED des betreffenden Geber-Kanals leuchtet nicht Geberstatus LED des betreffenden Geber blinkt
ERRVAL(1166)	Es wurde kein Z-Impuls erkannt, obwohl aufgrund der Bewegung ein Z-Impuls aufgetreten sein müsste. Mögliche Ursache(n): ■ eingestellte Geber-Auflösung stimmt nicht mit tatsächlicher Auflösung des Gebers überein ■ Geber defekt	immer 0	immer 0	Überprüfung der Auflösung des Gebers in den Parametern des SNC 021 im XN Safety Designer Verdrahtung des Gebers, sowie den Geber überprüfen	Safe DATA OK LED des betreffenden Geber-Kanals leuchtet nicht
ERRVAL(1167)	Es wurden zwischen zwei Z-Impulsen zu wenig Flanken auf der A- und B-Spur gezählt. Mögliche Ursache(n): ■ eingestellte Geber-Auflösung stimmt nicht mit tatsächlicher Auflösung des Gebers überein ■ Geber defekt	immer 0	immer 0	Überprüfung der Auflösung des Gebers in den Parametern des SNC 021 im XN Safety Designer Verdrahtung des Gebers, sowie den Geber überprüfen	Safe DATA OK LED des betreffenden Geber-Kanals leuchtet nicht
ERRVAL(1168)- ERRVAL(1171)	Ein Eingang oder Ausgang eines Erweiterungsmoduls ist im Fehlerzustand. Mögliche Ursache(n): ■ Fehler im Erweiterungsmodul	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1172)	Nicht verwendet				

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1173)	Die Werte der zurückgelesenen Ausgänge stimmen nicht mit den Sollwerten überein. Der Sollwert ist 1, der Istwert ist 0. Mögliche Ursache(n): ■ Hardware-Fehler (z.B. Ausgangstransistor defekt) ■ externer Fehler (z.B. Kurzschluss, Versorgungsspannung der Ausgänge fehlt) ■ Software-Fehler	immer 0	immer 0	Spannungsversorgung prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	LED des sicheren Ausganges blinkt
ERRVAL(1174)	Die Werte der zurückgelesenen Ausgänge stimmen nicht mit den Sollwerten überein. Der Sollwert ist 0, der Istwert ist 1. Mögliche Ursache(n): ■ Hardware-Fehler (z.B. Ausgangstransistor defekt) ■ externer Fehler (z.B. Kurzschluss, Versorgungsspannung der Ausgänge fehlt) ■ Software-Fehler	immer 0	immer 0	Spannungsversorgung prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	LED des sicheren Ausganges blinkt
ERRVAL(1175)	Die Werte der zurückgelesenen Ausgänge stimmen nicht mit den Sollwerten überein. Mögliche Ursache(n): ■ Hardware-Fehler (z.B. Ausgangstransistor defekt) ■ externer Fehler (z.B. Kurzschluss, Versorgungsspannung der Ausgänge fehlt) ■ Software-Fehler	immer 0	immer 0	Spannungsversorgung prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	LED des sicheren Ausganges blinkt

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1176)	Das Firmware-Update für ein lokales Erweiterungsmodul ist fehlgeschlagen. Mögliche Ursache(n): - falsches Updatefile - Flashen des Images im Erweiterungsmodul aufgrund eines Hardware-Defekts nicht möglich - Fehler im Erweiterungsmodul - Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1177)- ERRVAL(1180)	Nicht verwendet				
ERRVAL(1181)	Der sichere Messbereich des Eingangs wurde unterschritten. Mögliche Ursache(n): - Hardware-Fehler - Verdrahtungsfehler	immer 0	immer 0	Verdrahtung überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1182)	Der sichere Messbereich des Eingangs wurde überschritten. Mögliche Ursache(n): - Hardware-Fehler - Verdrahtungsfehler	immer 0	immer 0	Verdrahtung überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1183)	Die Toleranz des Endwertes zwischen den Controllern würde überschritten. Mögliche Ursache(n): Hardware-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1184)- ERRVAL(1188)	Nicht verwendet				

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1189)	Der sichere Messbereich des Eingangs wurde unterschritten. Fehler wird automatisch quittiert, wenn der Messwert wieder gültig. Mögliche Ursache(n): - Hardware-Fehler - Verdrahtungsfehler	immer 0	immer 0	Verdrahtung überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1190)	Der sichere Messbereich des Eingangs wurde überschritten. Fehler wird automatisch quittiert, wenn der Messwert wieder gültig. Mögliche Ursache(n): - Hardware-Fehler - Verdrahtungsfehler	immer 0	immer 0	Verdrahtung überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1191)- ERRVAL(1196)	Nicht verwendet				
ERRVAL(1197)	Es ist eine falsche CRC2 in der Konfiguration. Der Status des Moduls kann nicht auf Konfiguriert geändert werden. Mögliche Ursache(n): - Hardware-Fehler - falsche Konfiguration	immer 0	immer 0	Konfiguration auf Gültigkeit prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1198)	Der interne Zähler für die Zykluszeit arbeitet falsch. Mögliche Ursache(n): Software-Fehler	letzte Zeitdifferenz	maximale Zeitdifferenz	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1199)	Der entfernte sichere Eingangswert hat das maximale Alter überschritten. Mögliche Ursache(n): ■ Bus-Übertragungsfehler ■ Fehler in der Hardware des entfernten Moduls ■ Software-Fehler im entfernten Modul	immer 0	immer 0	Es ist zu überprüfen, ob die Standardkommunikation zu jedem benötigten Sicherheitsmodul intakt ist. Zudem ist sicherzustellen, dass sich kein benötigtes Sicherheitsmodul im Fehlerzustand befindet. Außerdem kann es sein, dass die maximale Transmission Time zu niedrig konfiguriert wurde. Diese Punkte sind zu überprüfen. Wurde der Fehler behoben, so kann mit einem Quit Error bzw. einem Neustart des Systems der Fehler aufgehoben werden.	Error-LED des Sicherheitsmoduls blinkt
ERRVAL(1200)	Nicht verwendet				
ERRVAL(1201)- ERRVAL(1205)	Es ist eine unerwartete Exception aufgetreten. Mögliche Ursache(n): ■ Software-Fehler	Adresse des unterbrochenen Codes	bei einer Exception außerhalb des FSB-Aufrufs: - immer 0 bei einer Exception während des FSB-Aufrufs: - FSB-ID in den unteren zwei Bytes - Objekt-Index in den oberen zwei Bytes	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1206)	Die Systemzeiten der Controller haben eine zu große Abweichung. Hierbei handelt es sich um einen permanenten Fehler, der nicht quittiert werden kann. Mögliche Ursache(n): ■ Hardware-Fehler	ermittelte Drift zwischen den beiden Controller-Systemzeiten	maximal zulässiger Drift	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1207)	Die Modul-Art und Modul-Variante in der Firmware stimmt nicht mit den in den unveränderlichen Modulparametern gespeicherten Werten überein. Mögliche Ursache(n): ■ falsche Firmware	Modul-Art und -Variante des Bootloaders	Modul-Art und -Variante der Firmware	Wenden Sie sich an Eaton	Error-LED des entsprechenden Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1208)- ERRVAL(1209)	Die unveränderlichen Modulparameter sind in den Controllern unterschiedlich. Mögliche Ursache(n): - Hardware-Fehler - Flash falsch programmiert	Modulparameter des eigenen Controllers	Modulparameter des anderen Controllers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1210)	Die Firmware Versionen sind in den Controllern unterschiedlich. Mögliche Ursache(n): falsche Firmware	Firmware-Version im eigenen Controller	Firmware-Version im anderen Controller	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1211)- ERRVAL(1212)	Beim ROM-Test in einem unveränderlichen Flash Bereich wurde ein Fehler festgestellt. Hierbei handelt es sich um einen permanenten Fehler, der nicht quittiert werden kann. Mögliche Ursache(n): Flash defekt	enthält genauere Informationen über die Ursache des Fehlers	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1213)- ERRVAL(1217)	Beim ROM-Test in einem änderbaren Flash Bereich wurde ein Fehler festgestellt. Mögliche Ursache(n): - Flash defekt - Modul wurde während des Flash Programmierens ausgeschaltet	enthält genauere Informationen über die Ursache des Fehlers	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1218)	Nach dem Einschalten der unteren Ausgangshalbleiter (FET) ist mindestens ein Ausgang auf 1 gegangen, obwohl keiner der oberen Ausgangshalbleiter (HSS) eingeschaltet war. Mögliche Ursache(n): Ausgangshalbleiter (HSS) defekt	Sollwert (Bit 0 = 1. Ausgang) immer 0	Istwert, betroffene Ausgänge (Bit 0 = 1. Ausgang)	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1219)	Der Inhalt einer SSDO Antwort ist in beiden Controllern unterschiedlich. Mögliche Ursache(n): ■ Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1220)	Es wurde versucht, das Flash mit einer ungültigen Länge zu beschreiben. Mögliche Ursache(n): ■ Programmfehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1221)	Es wurde versucht aus einem Datenbuffer zu lesen, wobei der Buffer weniger als die angeforderte Datenmenge enthält. Mögliche Ursache(n): ■ Software-Fehler ■ fehlerhaftes Safety-Tool (SSDO-Befehl)	Anzahl der Bytes, die aus dem Buffer gelesen werden sollten	Größe des Buffers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1222)	Die Zeit für den Testmodus oder die maximale Idle-Zeit ist abgelaufen. Mögliche Ursache(n): ■ Zeit des Testmodus ist abgelaufen, PLC ist im Reset	0: Zeit für den Testmodus ist abgelaufen	immer 0	Bei abgelaufenen Testmodus die Konfiguration erneut runterspielen und neu starten bzw. Fehler quittieren und neu starten	Error-LED des Sicherheitsmoduls leuchtet rot
		1: Zeit für die max. Idle-Zeit ist abgelaufen 2: Die Zeit für die max. Check Config Zeit ist abgelaufen	immer 0	Bei abgelaufener IDLE-Zeit bzw. Check Config Zeit den Fehler quittieren und erneut starten. PLC starten, wenn dieser im Reset	
ERRVAL(1223)	Der zeitliche Ablauf beim internen Test stimmt nicht. Es wurde versucht einen Test zu starten, während der vorherige Test noch nicht zu Ende ist. Mögliche Ursache(n): ■ Software-Fehler	Bit-Maske der Eingänge oder Nummer des fehlerhaften Testschrittes	genauere Informationen über die Ursache des Fehlers	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1224)	Der zeitliche Ablauf beim internen Test stimmt nicht. Es wurde versucht einen Test zu starten während der vorherige Test noch nicht zu Ende ist. Mögliche Ursache(n): Software-Fehler	Bit-Maske der zu testenden Eingänge	Bit-Maske der zuvor getesteten Eingänge	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1225)	Der Konfigurations-Zustand kann nicht auf 'konfiguriert' gesetzt werden, da die CRCs der zur Konfiguration gehörenden Bereiche ungültig sind. Mögliche Ursache(n): - es wurde ein Firmware-Update durchgeführt - Modul wurde falsch konfiguriert - Hardware-Fehler im Flash	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1226)	Der Konfigurations-Zustand kann nicht auf 'konfiguriert' gesetzt werden, da die vom Aufrufer übergebene CRC nicht mit der tatsächlichen übereinstimmt. Das Modul wechselt nicht in den Fehlerzustand. Es wird der Fehler im XN Safety Designer im Konsolenfenster angezeigt. Mögliche Ursache(n): - Modul wurde falsch konfiguriert - Hardware-Fehler im Flash	immer 0	immer 0	Das Sicherheitsmodul muss erneut konfiguriert werden Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1227)	Das Format der Konfigurationsdaten im Flash stimmt nicht. Mögliche Ursache(n): - Modul wurde falsch konfiguriert - Hardware-Fehler im Flash	tatsächlicher Konfigurationszustand	Soll-Konfigurationszustand	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1228)	Der Konfigurations-Zustand kann nicht auf 'konfiguriert' gesetzt werden, da die Konfiguration gelöscht ist. Mögliche Ursache(n): - Modul wurde falsch konfiguriert - Hardware-Fehler im Flash	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1229)	Die Konfiguration ist nicht in dem Zustand, in dem das Verified-Flag gesetzt werden kann. Mögliche Ursache(n): - die Konfiguration ist noch nie verteilt worden - die Konfiguration ist im Developer-Mode in die Safety-CPU geschrieben worden	immer 0	immer 0	Programm nicht im Developer Mode starten Programm erneut starten und somit Verteilung auslösen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1230)	Der SSDO Befehl kann aufgrund fehlender Berechtigung nicht ausgeführt werden. Mögliche Ursache(n): Fehler im XN Safety Designer	immer 0	immer 0	Mit dem XN Safety Designer erneut online gehen und Befehle erneut ausführen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1231)	Beim Onlinegehen konnte keine Session geöffnet werden. Mögliche Ursache(n): ein anderer XN Safety Designer ist bereits online	immer 0	immer 0	Prüfen, ob mehrere XN Safety Designer online sind, es darf nur ein XN Safety Designer zugreifen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1232)	Die max. Anzahl der entfernten Module ist überschritten worden.	Anzahl der entfernten Module	maximale Anzahl möglicher entfernter Module	Anzahl der entfernten Module prüfen und entsprechend ändern	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1233)	Der in den CFG-Pins kodierte Modultyp passt nicht zur Firmware oder es ist eine nicht unterstützte I/O-Erweiterungskarte angesteckt. Mögliche Ursache(n): ■ falsche Firmware	Bit 31 = 0: CFG-Pins laut Hardware Bit 31 = 1: ID der nicht unterstützten Erweiterungskarte Bit 30 = 1: Vendor-ID der nicht unterstützten Erweiterungskarte Bit 29 = 1: ID des nicht unterstützten Safety-Interface- Moduls	Wenn Reason Code 0 Bit31 = 0: CFG-Pins laut Firmware Wenn Reason Code 0 Bit31 = 1: immer 0 Wenn Reason Code 0 Bit30 = 1: Device-ID der nicht unterstützten Erweiterungskarte	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1234)	Der SSDO Befehl kann in diesem Laufzeit-Zustand nicht ausgeführt werden. Mögliche Ursache(n): ■ Fehler im Safety-Tool	immer 0	immer 0	Mit dem XN Safety Designer erneut Online gehen und Befehle erneut ausführen	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1235)	Die Parameter im Firmware-Header der neuen Firmware passen nicht zu diesem Modul. Mögliche Ursache(n): ■ falsche Firmware	Parameter in der Firmware	Parameter im Bootcode	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1236)	Beim Zugriff auf die SD-Karte ist ein Fehler aufgetreten. Mögliche Ursache(n): ■ SD-Karte wurde entfernt ■ defekte oder inkompatible SD-Karte	Identifikation der Fehlerursache	zuletzt aufgetretener Fehler	Es ist zu überprüfen, ob die SD-Karte entfernt wurde. Wurde die SD-Karte nicht entfernt, so ist die SD-Karte auszutauschen. Der Fehler kann mit dem Herausnehmen der SD-Karte und mit einem „Quit Error“-Befehl zurückgesetzt werden Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1237)	Schreib-/Lesefehler im Interpretercode der Konfiguration. Bei der Prüfung der Konfiguration wurde einer der folgenden Fehler festgestellt: a) Es wurde versucht eine Zelle zu beschreiben, die bereits von einer anderen Quelle beschrieben worden ist (z.B. wenn zwei COPY Befehle mit demselben Ziel existieren, oder wenn ein COPY Befehl eine nicht beschreibbare Zelle als Ziel angibt). b) Es wurde versucht, von einer nicht lesbaren Zelle zu lesen. Eine Zelle ist nicht lesbar, wenn es keine zugehörige Quelle gibt, von der sie beschrieben wurde (nicht initialisiert), oder wenn sie nicht als Quelle eines COPY Befehls verwendet werden darf (z.B. der Header des Objektspeichers). Mögliche Ursache(n): - es wurde eine fehlerhafte Konfiguration eingespielt - Flash defekt	Index der fehlerhaften Zelle	0 Schreibfehler 1 Lesefehler 2 bei FSoE: ein entfernter Eingang wird von mehr als einer Quelle beschrieben	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1238)	Der POST (PowerOn Selbsttest) ist nicht vollständig abgearbeitet worden. Mögliche Ursache(n): Software-Fehler	Bit-Maske mit den noch ausstehenden POST-Test-Schritten	immer 0	Während des Hochlauftests wurde ein Fehler festgestellt. Über den XN Safety Designer wird angezeigt, um welchen Fehler es sich handelt. Danach kann der Fehler behoben werden Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1239)	Der auf die SD-Karte zu schreibende Sektorpuffer ist in beiden Controllern unterschiedlich. Mögliche Ursache(n): Software-Fehler	Nummer des zu schreibenden Sektorpuffers	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1240)	Der Force-Modus wurde unzulässig beendet. Mögliche Ursache(n): ■ Auftreten eines Session-Timeout bzw. Login-Timeout, z.B. aufgrund einer Unterbrechung der Online-Kommunikation mit dem XN Safety Designer ■ Wechsel des Login-Levels (z.B. Abmelden) ■ Schließen der Session im XN Safety Designer	immer 0	immer 0	Bei einer Unterbrechung der Kommunikation zwischen XN Safety Designer und Modul, den Fehler quittieren	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1241)	Das Format der Passwörter auf der SD-Karte stimmt nicht. Mögliche Ursache(n): ■ SD-Karte defekt ■ Inhalt wurde nachträglich geändert	immer 0	immer 0	Versuche, die SD-Karte neu zu bespielen. Tritt der Fehler immer noch auf, ist die SD-Karte zu tauschen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1242)	Interner Fehler bei der Verarbeitung der STD0s. Mögliche Ursache(n): ■ Software-Fehler	Länge des zu schreibenden STD0s	maximale Länge eines STD0s	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(1243)	Die Information über eine Modul-Sicherheitsnummer eines entfernten Moduls wird benötigt, ist aber nicht vorhanden bzw. ungültig. Mögliche Ursache(n): ■ Software-Fehler	Wert der ungültigen Modul-Sicherheitsnummer (0 oder 0xFFFFfff)	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1244)	Im Interpretercode der Konfiguration ist eine FSB-ID angegeben, für die kein entsprechender Code existiert. Mögliche Ursache(n): ■ Fehler im Safety-Tool	FSB-ID	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1245)	Ein in den Konfigurationsdaten angegebener Wert ist zu groß. Mögliche Ursache(n): ■ Eingabe eines zu großen Wertes im XN Safety Designer	gibt an welcher Parameter zu groß ist 0 Testzeit im temporären Operational Zustand 1 Zykluszeit für die Laufzeitüberwachung 2 maximales Alter eines entfernten Eingangs 3 Filterzeit eines lokalen Eingangs 4 Watchdog-Zeit ein Safety-Interface-Eingangs 5 Optional-Flag eine Safety-Interface-Eingangs 6 Cfg-Major Version 7 unbekannte Daten nach dem Container Header 8 am Ende des Funktionsblockbereichs steht kein 0 9 ungültige Länge für die FSoE-Slave-Parameter 10 ungültiger Wert in der FSoE-Konfiguration 11 ungültige FSoE-Connection ID 12 ungültige FSoE-Slave Adresse 13 gespeicherten FSoE-Adressen in den Controllern unterschiedlich oder benötigte Anzahl der	Limit	Wert in der Konfiguration richtig stellen (kleineren Wert eingeben)	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
		Adressen stimmt nicht 14 Slave-Adresse, die zur Beantwortung des C_GET_SLAVEADDR Befehls verwendet wird, konnte nicht eindeutig ermittelt werden			
ERRVAL(1246)	Fehler in den Funktionsblöcken. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1247)	Fehler in den Funktionsblöcken, Überlauf bei internen Berechnungen. Mögliche Ursache(n): ■ Software-Fehler ■ Falsche Parameter ■ Safedint-Werte außerhalb des zulässigen Bereichs			Überprüfung der Parameter und des eingestellten Wertebereichs, Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	
ERRVAL(1248)	Fehler in den Funktionsblöcken, falsche Safedint-Werte. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1249)	Fehler in den Funktionsblöcken, falscher interner Status. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1250)	Fehler in den Funktionsblöcken, falsche Parameter. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(1251)	Fehler in den Funktionsblöcken, falsche Safebool-Werte. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1252)	Fehler in den Funktionsblöcken, falsche ID des Funktionsblocks. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1253)	Fehler in den Funktionsblöcken, falsche Ende-Kennung. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1254)	Fehler in den Funktionsblöcken, falsche Start-Kennung. Mögliche Ursache(n): ■ Software-Fehler			Wenden Sie sich an Eaton	
ERRVAL(1255)	Funktionsblock hat einen Fehler-Code zurückgegeben. Möglicher Ursache(n): ■ Software-Fehler	immer 0	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10000)	Die externe Versorgungsspannung der Geber ist nicht in Ordnung. Mögliche Ursache(n): ■ Externe Versorgungsspannung nicht angeschlossen, ■ Hardware-Fehler	Byte 0 = 0; Beim Hochlauf wurde festgestellt, dass die Versorgungsspannung nicht in Ordnung ist Byte 1: Position des Moduls hinter der Safety CPU	immer 0	Versorgungsspannung prüfen. Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10001)	Der maximal zulässige Strom eines Gebers wurde überschritten (siehe „Technische Daten“ im Produktdatenblatt des SNC 021 Moduls). Mögliche Ursache(n): ▪ Verdrahtungsfehler ▪ Querschuss ▪ Kurzschluss ▪ Verwendung eines falschen Gebers	Byte 0: 0: max. Strom des ersten Gebers überschritten 1: max. Strom des zweiten Gebers überschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Max. zulässiger Strom in 10 µA Byte 2 und 3: Gemessener Strom des Gebers in 10 µA	Strom des Gebers prüfen Verdrahtung des Gebers, sowie den Geber überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10002)	Der für einen Geber gemessene Strom liegt unter dem erlaubten Limit. Mögliche Ursache(n): ▪ Verdrahtungsfehler ▪ Kabelbruch ▪ Geber defekt ▪ Hardware-Fehler	Byte 0: 0: das Limit für den Geber 1 ist unterschritten 1: das Limit für den Geber 2 ist unterschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Unteres Limit für den erlaubten Strom in 10 µA Byte 2 und 3: Gemessener Strom des Gebers in 10 µA	Verdrahtung des Gebers, sowie den Geber überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10003)	Der für einen Geber gemessene Strom liegt über dem erlaubten Limit. Mögliche Ursache(n): ▪ Verdrahtungsfehler ▪ Querschuss ▪ Kurzschluss ▪ Geber defekt ▪ Hardware-Fehler	Byte 0: 0: das Limit für den Geber 1 ist überschritten 1: das Limit für den Geber 2 ist überschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Oberes Limit für den erlaubten Strom in 10 µA Byte 2 und 3: Gemessener Strom des Gebers in 10 µA	Verdrahtung des Gebers, sowie den Geber überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10004)	Der für einen Geber gemessene Spannungsoffset für die Stromüberwachung liegt bei abgeschalteter Versorgungsspannung unter dem erlaubten Limit Mögliche Ursache(n): ■ Hardware-Fehler	Byte 0: 0: das Limit für den Geber 1 ist unterschritten 1: das Limit für den Geber 2 ist unterschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Unteres Limit für den erlaubten Spannungsoffset in mV Byte 2 und 3: Gemessener Spannungsoffset des Gebers in mV	SNC Modul tauschen	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10005)	Der für einen Geber gemessene Spannungsoffset für die Stromüberwachung liegt bei abgeschalteter Versorgungsspannung über dem erlaubten Limit. Mögliche Ursache(n): ■ Hardware-Fehler	Byte 0: 0: das Limit für den Geber 1 ist überschritten 1: das Limit für den Geber 2 ist überschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Oberes Limit für den erlaubten Spannungsoffset in mV Byte 2 und 3: Gemessener Spannungsoffset des Gebers in mV	SNC Modul tauschen	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10006)	Die für einen Geber gemessene Versorgungsspannung liegt unter dem erlaubten Limit. Mögliche Ursache(n): ■ Verdrahtungsfehler ■ Kabelbruch ■ Querschluss ■ Hardware-Fehler	Byte 0: 0: das Limit für den Geber 1 ist unterschritten 1: das Limit für den Geber 2 ist unterschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Unteres Limit für die erlaubte Spannung in mV Byte 2 und 3: Gemessene Spannung des Gebers in mV	Verdrahtung der externen Spannungsversorgung sowie der Geber überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10007)	Die für einen Geber gemessene Spannung liegt über dem erlaubten Limit. Mögliche Ursache(n): - Verdrahtungsfehler - Hardware-Fehler	Byte 0: 0: das Limit für den Geber 1 ist überschritten 1: das Limit für den Geber 2 ist überschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Oberes Limit für die erlaubte Spannung in mV Byte 2 und 3: Gemessene Spannung des Gebers in mV	Verdrahtung der externen Spannungsversorgung sowie der Geber überprüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10008)	Die Parameter für die Sollstromvorgabe sind ungültig. Mögliche Ursache(n): - Software-Fehler - Safety CPU nicht kompatibel (siehe Produktdatenblatt des SNC 021 Moduls)	Byte 0: Fehlerinformation Byte 1: Position des Moduls hinter der Safety CPU	immer 0	Safety-CPU prüfen Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10009)	Beim Test der Prüfsumme über die konstanten Elemente für die Stromüberwachung im RAM wurde ein Fehler entdeckt. Mögliche Ursache(n): - Software-Fehler - RAM defekt	Byte 0: Fehlerinformation Byte 1: Position des Moduls hinter der Safety CPU	immer 0	SNC Modul tauschen	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10010)	Die beiden RS422-Treiber liefern für zumindest ein Signal der Geber unterschiedliche Ergebnisse. Mögliche Ursache(n): Hardware-Fehler	Byte 0: Information, welches Signal betroffen ist (Bit 0 = Signal A, Bit 1 = Signal B, Bit 2 = Signal Z) Byte 1: Position des Moduls hinter der Safety CPU	immer 0	SNC Modul tauschen	Error-LED der Safety-CPU leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10011)	Die max. zulässige Eingangs bzw. Zählerfrequenz wurde überschritten (siehe Produktdatenblatt des SNC 021 Moduls). Mögliche Ursache(n): - Geber hat eine zu hohe Auflösung - Geber wird mit einer zu hohen Geschwindigkeit betrieben	Byte 0: 0: die maximale Eingangs- bzw. Zählerfrequenz für den Geber 1 wurde überschritten 1: die maximale Eingangs- bzw. Zählerfrequenz für den Geber 2 wurde überschritten Byte 1: Position des Moduls hinter der Safety CPU	Byte 0 und 1: Maximal zulässige Zählerfrequenz in Inkremente/ms Byte 2 und 3: Gemessene Zählerfrequenz in Inkremente/ms	Überprüfung des Gebers hinsichtlich Geberauflösung und Geschwindigkeit. Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	Error-LED der Safety-CPU leuchtet rot
ERRVAL(10012)	Die konfigurierte Encoder-Auflösung liegt außerhalb der definierten Grenzen (siehe Produktdatenblatt des Moduls SNC 021). Mögliche Ursache(n): Die konfigurierte Encoder-Auflösung ist zu hoch	Byte 0: 0 die maximal konfigurierte Auflösung für den Geber 1 wurde überschritten 1 die maximal konfigurierte Auflösung für den Geber 2 wurde überschritten Byte 1: Position des Moduls hinter der Safety-CPU	Byte 0 und 3: Konfigurierte Geberauflösung	Überprüfung des Gebers hinsichtlich Geberauflösung. Falls der Fehler weiterhin auftritt: Wenden Sie sich an Eaton	
ERRVAL(10013)	Die 3V3 Versorgungsspannung ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10014)	Die 3V3 Versorgungsspannung ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10015)	Die 5V0 Versorgungsspannung ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10016)	Die 5V0 Versorgungsspannung ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10017)	Die 5V1 Versorgungsspannung ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10018)	Die 5V1 Versorgungsspannung ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10019)	Die 15V0 Versorgungsspannung ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10020)	Die 15V0 Versorgungsspannung ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10021)	Die 24V1 Versorgungsspannung ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10022)	Die 24V1 Versorgungsspannung ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10023)	Die analoge Spannungsüberwachung hat einen zu niederen Wert gemeldet. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10024)	Die analoge Spannungsüberwachung hat einen zu hohen Wert gemeldet. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: Index des ADC Kanals Byte 1: Position des Moduls	erlaubter Grenzwert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10025)	Abgleichdaten sind falsch oder konnten nicht gelesen werden. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache 0 Lesefehler 1 falsche Länge 2 falsche CRC 3 falsche Version 4 falsche Modul-ID 5 ungültige Controller-ID	dient zur Identifizierung der Fehlerursache	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10026)	Die 24 V-Sensorversorgung ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: 0=Fehler durch Messung ausgelöst, 1=Fehler durch DCOK-Pin ausgelöst Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Spannungslimit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10027)	Die 24 V-Sensorversorgung ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Spannungslimit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10028)	Die Common-Mode-Spannung (CMVR) ist über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: CMVR Index (0-3) Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Spannungslimit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10029)	Die Common-Mode-Spannung (CMVR) ist unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: CMVR Index (0-3) Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Spannungslimit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10030)	Die Temperatur oder eine ADC-Referenz liegt über dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: 0=ADC-Referenz, 1=Temperatur Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Limit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10031)	Die Temperatur oder eine ADC-Referenz liegt unter dem erlaubten Grenzwert. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 0: 0=ADC-Referenz, 1=Temperatur Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Limit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10032)	Die Kalibrier- oder Offsetspannung liegt unter dem erlaubten Limit	dient zur Identifizierung der Fehlerursache Byte 0: Bit 0-3 Kanalindex Byte 1: Position des Moduls Byte 2-3: erweiterter Fehler-Code	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Limit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

15. Fehler-Codes

Error-Variable	Beschreibung	Reason-Code 0	Reason-Code 1	Maßnahme	LED-Anzeige
ERRVAL(10033)	Die Kalibrier- oder Offsetspannung liegt über dem erlaubten Limit	dient zur Identifizierung der Fehlerursache Byte 0: Bit 0-3 Kanalindex Byte 1: Position des Moduls Byte 2-3: erweiterter Fehler-Code	dient zur Identifizierung der Fehlerursache Byte 0-1: erlaubtes Limit Byte 2-3: gemessener Wert	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10034)	Vom ADC wurden unerwartete Daten empfangen. Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 1: Position des Moduls	dient zur Identifizierung der Fehlerursache Byte 0-3: empfangene Daten	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot
ERRVAL(10035)	Es ist ein Brownout Fehler aufgetreten Mögliche Ursache(n): ■ Hardware-Fehler	dient zur Identifizierung der Fehlerursache Byte 1: Position des Moduls Byte 2-3: erweiterter Fehler-Code	immer 0	Wenden Sie sich an Eaton	Error-LED des Sicherheitsmoduls leuchtet rot

Anhang

A.1 Weitere Nutzungsinformationen	447
A.2 Shortcuts - Tastenkombinationen und Sondertasten	448
Stichwortverzeichnis	449

A.1 Weitere Nutzungsinformationen

Hardware

Weitere Informationen und Informationen zu ergänzenden XN Safety-Modulen finden Sie in folgenden Dokumenten:

	XN Safety Systemhandbuch	MN050020
	Montageanweisung XS-102-C00-000, XS-322-...	IL050036ZU
	Handbuch XN Safety CPU-Modul XS-102-C00-000	MN050019
	Handbuch XN Safety-Modul XS-322-10DI-PF	MN050021
	Handbuch XN Safety-Modul XS-322-8DO-P20	MN050022
	Handbuch XN Safety-Modul XS-322-8DIO-PF20	MN050023
	Handbuch XN Safety-Modul XS-322-2INC	MN050024
	Handbuch XN Safety-Modul XS-322-4AI-I2	MN050025
	Handbuch XN Safety-Modul XS-322-2DO-RNOAC	MN050026
	Handbuch XN Safety-Modul XS-322-2DO-RNODC	MN050027

Download Center, Eaton Online-Katalog

Mit der Eingabe "xnsafety" in das Suchfeld gelangen Sie gezielt zu dieser Produktgruppe aus dem Bereich Automatisierung, Steuern und Visualisieren.

 Eaton.com/documentation

 Eaton.com/ecat

Produktinformation

Aktuelle Informationen finden Sie auf der Produktseite.

 Eaton.com/xnsafety

A.2 Shortcuts - Tastenkombinationen und Sondertasten

Häufig durchzuführende Aufgaben können Sie mit Funktionstasten und Tastenkombinationen beschleunigen. Im Folgenden finden Sie eine Übersicht dazu.

A.2.1 Funktionstasten und Tastenkombinationen

A.2.1.1 Tastenkombinationen im Menü FILE

Zweck	Tastenkombination
Neues Projekt öffnen	Strg + N
Bestehendes Projekt öffnen	Strg + O
Datei drucken	Strg + P
Projekt speichern	Strg + S

A.2.1.2 Tastenkombinationen im Menü EDIT

Zweck	Tastenkombination
Aktion rückgängig machen	Strg + Z
Aktion wiederherstellen	Strg + Y

A.2.1.3 Funktionstaste im Menü BUILD

Zweck	Taste / Tastenkombination
Projekt generieren	F9

A.2.1.4 Funktionstasten und Tastenkombination im Menü DEBUG

Zweck	Taste / Tastenkombination
Online	Alt + F6
Reset geforcete Werte	F8
Online Statusanzeige	F6

Stichwortverzeichnis

A

ABS (Absolute value)	173
ABS_NS (Absolutwert, not safe)	173
Abschätzung der Zykluszeit des Anwen- derprogramms	100
ADD (Addition)	173
ADD_NS (Addition, not safe)	174
After Sales Service	2
Allgemeine Arbeitsweise	67
Analog-Eingangsmodul;Beschreibung	379
AND_5I-Funktion	199
AND_BW (UND, bitweise)	179
AND_NS-Funktion	199
Ändern der Kommentargröße	74
Ändern der Spaltenbreite	78
AND-Funktion	198
Arbeiten im Netzwerkeditor	68
Archivierung	122
Aufruf der Definition eines Elements	75
Aufruf der Startpunkte/Endpunkte einer Ver- bindung	76
Aufruf der temp. Variablen	77

B

Berechnung der Monitoring Settings	97
Beschreibung	378
BOOL_TO_SAFEBOOL	169

C

Collision_Detection	211
Copyright	2
Counter-Funktionsbloেকে	206
Cross Circuit Detection	81

CTD	207
CTU	207
CTUD	208

D

Debuggen	101
Digital-Ausgangsmodul;Beschreibung	381
Digital-Ein/Ausgangsmodul;Beschreibung	382
Digital-Eingangsmodul;Beschreibung	380
DINT_Compare	197
DINT_TO_SAFEDINT	170
Download Center	447
Download Wizard	154
Drucken	114

E

ecat	447
Editor	58
Einfügen von Kommentaren	73-74
Einfügen von Netzwerkelementen	68
Einstellparameter im Projekt	81
Enable Signal von sicheren Ausgängen	85
EQ (gleich)	192
EQ_NS (gleich, not safe)	192
Ersetzen von bestehenden Elementen	75
Ersetzen von Platzhaltern	75
Export der Downloaddaten für eine Safety-CPU	160

F

F_TRIG	206
Fehlercodes	386
Fehlermeldungen beim Einfügen eines Ele- ments ins Netzwerk	79
Filter Time	81
FlipFlop-Funktionsbloেকে	208
Forcen	102

Füllen der Ausgangsspalte	67
Füllen der Eingangsspalte	67
Funktionsblock-spezifische Fehler und Status-Codes	221

G

GE (größer gleich)	190
GE_NS (größer gleich, not safe)	190
Gesamtreaktionszeit	100
Get Module State	159
GT (größer als)	189
GT_NS (größer als, not safe)	190

H

Handbücher	447
Hardware Revision	82

I

Impressum	2
Inkrementalgeber-Modul;Beschreibung	385

K

Komplexe Funktionsblöcke	210
Kopieren, Ausschneiden und Einfügen von Elementen	74
Kopierschutz	2

L

Länge der PDO-Frames	100
LE (kleiner gleich)	191
LE_NS (kleiner gleich, not safe)	192
LIMIT (innerhalb von Grenzen)	195
LIMIT_NS (innerhalb Grenzen, not safe)	195
Logical-Funktionsblöcke	198
Löschen von Kommentaren	73
Löschen von Netzwerkelementen	69

Löschen von Verbindungen	71
LT (kleiner als)	191
LT_NS (kleiner als, not safe)	191

M

Markennamen	
Produktnamen	2
MAX (Maximum)	194
Max. Cycle Time	81
MAX_NS (Maximum, not safe)	194
Maximum Transmission Time	81
Maximum Transmission Time (Output as Input)	81
MIN (Minimum)	193
MIN_NS (Minimum, not safe)	193
MULDIV (Multiplikation und Division)	176
MULDIV_NS (Multiplikation und Division, not safe)	175

N

Netzwerkeditor	63
Netzwerkelemente	64
NOT_BW (NOT, bitweise)	180
NOT-Funktion	200

O

Oberfläche	32
Online-Aktionen	152
Online-Aktionen und Download	149
Online-Katalog	447
Optionale Module	87
OR_5I-Funktion	201
OR_BW (ODER, bitweise)	179
OR-Funktion	200
Originalbetriebsanleitung	2

P

Parameter Revision	82
Parametrierung von Netzwerkelementen	69
Passwörter	157
Passwortmechanismus	123
Path Speed	183

R

R_TRIG	206
Relais-Ausgangsmodul;Beschreibung	384
Relais-DC-Ausgangsmodul;Beschreibung	383
Ressourcenverbrauch	99
Ressourcenverbrauch und Reaktionszeit	98
Restart_Interlock	181
RS_Flipflop	209

S

SAFEBOOL_TO_BOOL	170
SAFEDINT_TO_DINT	170
SAFEDINTERROR_TO_SAFEDINT	171
SDINT_Compare	196
Selektion von Netzwerkelementen	68
Selektion von Verbindungen	71
SERROR_TO_SFALSE	171
SF_Antivalent-Funktion	220
SF_Antivalent-Funktion Fehlererkennung	221
SF_Antivalent-Funktion Fehlerverhalten	221
SF_Antivalent-Funktion Funktionsblock-spezifische Fehler und Status-Codes	221
SF_Antivalent-Funktion Status Diagramm	222
SF_Antivalent-Funktion Zeit-Diagramm	223
SF_DirectionMonitor	354
SF_DirectionMonitor - Fehlerverhalten	356
SF_DirectionMonitor - Status Diagramm	357
SF_EDM Besonderheiten	228
SF_EDM Fehlererkennung	225

SF_EDM Fehlerverhalten	225
SF_EDM Status Diagramm	229
SF_EDM Zeit-Diagramm	230
SF_EDM-Funktion	224
SF_EmergencyStop Fehlererkennung	232
SF_EmergencyStop Fehlerverhalten	233
SF_EmergencyStop Status Diagramm	234
SF_EmergencyStop Zeit-Diagramm	235
SF_EmergencyStop-Funktion	231
SF_EnableSwitch Fehlererkennung	238
SF_EnableSwitch Fehlerverhalten	238
SF_EnableSwitch Status Diagramm	240
SF_EnableSwitch Zeit-Diagramm	241
SF_EnableSwitch-Funktion	236
SF_Equivalent Fehlererkennung	243
SF_Equivalent Fehlerverhalten	243
SF_Equivalent Status Diagramm	244
SF_Equivalent Zeit-Diagramm	245
SF_Equivalent-Funktion	242
SF_ESPE Fehlererkennung	247
SF_ESPE Fehlerverhalten	248
SF_ESPE Status Diagramm	249
SF_ESPE Zeit-Diagramm	250
SF_ESPE-Funktion	246
SF_GuardLocking Betriebssequenz	252
SF_GuardLocking Fehlererkennung	253
SF_GuardLocking Fehlerverhalten	254
SF_GuardLocking Status Diagramm	256
SF_GuardLocking Zeit-Diagramm	257
SF_GuardLocking-Funktion	251
SF_GuardMonitoring Fehler- und Reset-Verhalten	260
SF_GuardMonitoring Fehlererkennung	259
SF_GuardMonitoring Status-Diagramm	262
SF_GuardMonitoring Zeit-Diagramm	263
SF_GuardMonitoring-Funktion	258

SF_LimitComparator	341	SF_OutControl Zeit-Diagramm	310
SF_LimitComparator - Fehlerverhalten	343	SF_OutControl-Funktion	306
SF_LimitComparator Status Diagramm	344	SF_RampMonitor	350
SF_ModeSelector Besonderheiten	267	SF_RampMonitor - Fehlerverhalten	352
SF_ModeSelector Fehlererkennung	266	SF_RampMonitor - Formel zur Berechnung der Geschwindigkeitskurve	351
SF_ModeSelector Fehlerverhalten	266	SF_RampMonitor - Status Diagramm	353
SF_ModeSelector Status-Diagramm	268	SF_SafetyRequest Fehlererkennung	312
SF_ModeSelector Zeit-Diagramm	268	SF_SafetyRequest Fehlerverhalten	312
SF_ModeSelector-Funktion	264	SF_SafetyRequest Status Diagramm	314
SF_MutingPar Fehlererkennung	272	SF_SafetyRequest Zeit-Diagramm	315
SF_MutingPar Fehlerverhalten	272	SF_SafetyRequest-Funktion	311
SF_MutingPar Status Diagramm	277	SF_SampleHold	176
SF_MutingPar Zeit-Diagramm	278	SF_SkewMonitor	358
SF_MutingPar_2Sensor	279	SF_SkewMonitor - Fehlerverhalten	360
SF_MutingPar_2Sensor Fehlererkennung	281	SF_SkewMonitor - Status Diagramm	361
SF_MutingPar_2Sensor Fehlerverhalten	281	SF_TestableSafetySensor Fehlererkennung	318
SF_MutingPar_2Sensor Status Diagramm	284	SF_TestableSafetySensor Fehlerverhalten	318
SF_MutingPar_2Sensor Zeit-Diagramm	284	SF_TestableSafetySensor Status Diagramm	323
SF_MutingPar-Funktion	270	SF_TestableSafetySensor Zeit-Diagramm	324
SF_MutingSeq Fehlererkennung	287	SF_TestableSafetySensor-Funktion	316
SF_MutingSeq Fehlerverhalten	287	SF_TwoHandControlTypell Fehlererkennung	325
SF_MutingSeq Status-Diagramm	290	SF_TwoHandControlTypell Fehlerverhalten	325
SF_MutingSeq Zeit-Diagramm	291	SF_TwoHandControlTypell Status Diagramm	328
SF_MutingSeq-Funktion	285	SF_TwoHandControlTypell Zeit-Diagramm	329
SF_NumericSelector	346	SF_TwoHandControlTypell-Funktion	325
SF_NumericSelector - Fehlerverhalten	347	SF_TwoHandControlTypelll Besonderheiten	332
SF_NumericSelector - Status Diagramm	349	SF_TwoHandControlTypelll Fehlererkennung	330
SF_Optional_Pwd Fehlererkennung	293	SF_TwoHandControlTypelll Fehlerverhalten	330
SF_Optional_Pwd Fehlerverhalten	293	SF_TwoHandControlTypelll Status Diagramm	333
SF_Optional_Pwd-Funktion	292	SF_TwoHandControlTypelll Zeit-Diagramm	334
SF_Optional_Pwdll Fehlererkennung	298	SF_TwoHandControlTypelll-Funktion	330
SF_Optional_Pwdll Fehlerverhalten	298	SHL (Shift-Left)	177
SF_Optional_Pwdll-Funktion	297	SHL_NS (Shift-Left, not safe)	177
SF_OutControl Fehlererkennung	307	SHR (Shift-Right)	178
SF_OutControl Fehlerverhalten	307	SHR_NS (Shift-Right, not safe)	178
SF_OutControl Status-Diagramm	309		

Soll-Istvergleich der Hardware	151
Spalte für Ausgänge	62
Spalte für Eingänge	59
Speed	185
Speichern als XML-Datei	122
SR_Flipflop	209
Status-Diagramm SF_Optional_Pwd	295
Status-Diagramm SF_Optional_PwdII	301
SUB (Subtraktion)	174
SUB_NS (Subtraktion, not safe)	175

T

Tastenkombinationen	448
Programmierung	448
Temporary Operation Time	81
Timer-Funktionsblöcke	204
TOFF	204
TON	205
TP	205
Trigger-Funktionsblöcke	205

U

Übersicht über Aktionen abhängig vom Pass- wortlevel	158
Upload Log-Data	158

V

Validierung und Generierung der Dow- nloaddaten	89
Vergleichende Funktionsblöcke	189
Verschieben von Kommentaren	73
Verschieben von Netzwerkelementen	69
Versionierung	122
Versionskontrolle und Archivierung	122

W

Watch Fenster	104
Weiterführende Literatur	447
Werkzeug-Symbolleiste	38

X

XOR_BW (XOR, bitweise)	180
XOR-Funktion	201

Z

Zeit-Diagramm SF_Optional_Pwd	296
Zeit-Diagramm SF_Optional_PwdII	301
Ziehen von Verbindungen	70

Eaton ist ein intelligentes Energiemanagementunternehmen, das sich dem Ziel verschrieben hat, für mehr Lebensqualität zu sorgen und die Umwelt zu schützen. Wir handeln verantwortlich und nachhaltig und unterstützen unsere Kunden beim Energiemanagement – heute und in Zukunft. Wir setzen auf die globalen Wachstumstrends Elektrifizierung und Digitalisierung und beschleunigen so die Umstellung der Welt auf erneuerbare Energien, tragen zur Lösung der weltweit dringendsten Herausforderungen im Energiemanagement bei und setzen uns für das Beste für unsere Stakeholder und die ganze Gesellschaft ein.

Weitere Informationen finden Sie unter [Eaton.com](https://www.eaton.com).



Powering Business Worldwide

Eaton Industries GmbH

Hein-Moeller-Str. 7–11

D-53115 Bonn

© 2025 Eaton

Alle Rechte vorbehalten.

06/25 MN050028