



EtherNet/IP

EtherCAT

Original-Programmierhandbuch
SmartSPS SafeLine AC4S
mit Feldbusschnittstellen

AC402S

AC412S

AC422S

AC432S

für ISO 13849 bis PL e
für IEC 61508 SIL 3 und 62061 bis SIL CL 3

Firmware-Release: 4.2.5
CODESYS Development System: 3.5.9.73

Deutsch

12/2017

7391150/00

© ifm electronic gmbh

Inhaltsverzeichnis

1	Vorbemerkung	6
1.1	Rechtliche Hinweise.....	6
1.2	Zweck des Dokuments.....	6
1.3	Verwendete Symbole und Formatierungen	7
1.4	Übersicht: Anwender-Dokumentation für AC4S	7
1.5	Übersicht: CODESYS-Dokumentation von 3S	8
1.6	Änderungshistorie	8
2	Sicherheitshinweise	9
2.1	Allgemeine Sicherheitshinweise	9
2.2	Notwendige Vorkenntnisse	9
2.3	Verwendete Warnhinweise	10
3	Systemvoraussetzungen	11
3.1	Hardware.....	11
3.2	Software	11
3.3	Lizenzierung.....	11
4	Installation	12
4.1	Programmiersoftware CODESYS	12
4.1.1	CODESYS Development System installieren.....	12
4.2	ifm AS-i Package.....	13
4.2.1	Komponenten des ifm-Packages.....	13
4.2.2	ifm-Package installieren	14
4.2.3	ifm-Package deinstallieren	14
5	Schnelleinstieg	15
5.1	CODESYS starten.....	15
5.2	CODESYS-Projekt erstellen	16
5.2.1	Neues Projekt mit AC4S erstellen	17
5.2.2	Übersicht: Projektstruktur mit AC4S	19
5.3	CODESYS-Dokumentation nutzen	20
5.4	Nutzerkonto anlegen.....	20
5.5	Programmierschnittstelle konfigurieren	21
5.5.1	Kommunikationspfad der Standard-SPS setzen	21
5.5.2	Kommunikationspfad der fehlersicheren SPS setzen.....	22
6	Systemkonfiguration	23
6.1	Standard-SPS konfigurieren	23
6.2	Fehlersichere SPS konfigurieren	24
6.3	Sichere AS-i Slaves konfigurieren	25
6.3.1	Hinweise.....	25
6.3.2	Sichere AS-i Slaves in Projekt einbinden	27
6.3.3	Sichere AS-i Slaves aus Projekt entfernen.....	31
6.4	Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren.....	32
6.4.1	Hinweise.....	32
6.4.2	Sicheres lokales Gerät in Projekt einbinden	34
6.4.3	Testsignal an lokalem Ausgang erzeugen.....	39

Inhalt

6.4.4	Sicheres lokales Gerät aus Projekt entfernen	39
6.5	Standard-Geräte an lokaler E/A-Schnittstelle konfigurieren	40
6.6	IP-Einstellungen ändern.....	40
6.7	Zusätzliches Gerät zum Projekt hinzufügen	41
6.8	Feldbus-Funktionalität erweitern	42
6.8.1	Verfügbare Feldbus-Stacks	42
6.8.2	EtherCAT Master nutzen	43
6.8.3	Feldbus-Stack hinzufügen	46
7	Programmierung	47
7.1	Objekte einer Standard-SPS-Anwendung	47
7.2	Standard-SPS-Anwendung erstellen	48
7.2.1	Remanente Variablen nutzen	48
7.2.2	Unterstützte Programmiersprachen.....	48
7.2.3	Systemzeit des Geräts ändern	49
7.3	Auf Standard-Ein- und Ausgangsdaten zugreifen	50
7.3.1	Möglichkeiten des Zugriffs auf Ein- und Ausgangsdaten.....	50
7.3.2	Gültigkeit der Schnittstellen-Daten	51
7.3.3	Prozessdaten der AS-i Slaves.....	52
7.3.4	Feldbus-Daten.....	54
7.3.5	Prozessdaten der nicht-sicheren lokalen Geräte.....	56
7.4	Funktionen des ifm-Packages nutzen.....	57
7.4.1	Steuerinterface der ifm-Funktionsbausteine	57
7.4.2	System konfigurieren.....	59
7.4.3	AS-i Master konfigurieren	59
7.4.4	AS-i Slaves konfigurieren	59
7.4.5	AS-i Netzwerk verwalten	60
7.4.6	Kommandos an System und AS-i Master senden	63
7.5	Visualisierungen nutzen	64
7.5.1	Unterstützte Visualisierungstypen	64
7.5.2	Visualisierung zu einem Projekt hinzufügen	65
7.5.3	Visualisierung erstellen.....	66
7.5.4	Visualisierung konfigurieren	67
7.6	Querkommunikation	68
7.6.1	Netzwerkvariablen nutzen	68
7.7	Task-Abarbeitung konfigurieren.....	70
7.7.1	Haupt-Task konfigurieren	70
7.7.2	Visualisierung-Task konfigurieren	70
8	Programmierung der fehlersicheren SPS	71
8.1	Objekte einer sicheren Anwendung	72
8.2	Sichere Anwendung erstellen	73
8.2.1	Unterstützte Programmierbausteine (POUs)	74
8.2.2	Verfügbare Safety-Bibliotheken.....	75
8.2.3	Unterstützte Programmiersprachen.....	75
8.2.4	Mindestanzahl an AS-i Slaves.....	76
8.2.5	Variablendeklaration bei Datentransfer zwischen Standard- und Safety-Bereich.....	76
8.2.6	Verfügbarer Speicherplatz.....	77
8.2.7	Sicheres Projekt pinnen.....	77
8.3	Safety-Funktionen des ifm AS-i Packages nutzen.....	78
8.3.1	Sichere AS-i Slaves: Logisches Gerät rücksetzen	78
8.3.2	Sichere AS-i Slaves: Diagnose-Informationen ausgeben	79
8.3.3	Sichere AS-i Slaves: Hilfssignale HSI_1 und HSI_2 an sichere AS-i Ausgangsmodule senden	79
8.3.4	Lokale E/A-Schnittstelle: Logisches Gerät rücksetzen	79
8.3.5	Lokale E/A-Schnittstelle: Diagnose-Informationen ausgeben.....	80
8.3.6	Lokale E/A-Schnittstelle: Auf nicht sichere Ein-/Ausgänge zugreifen	80
8.3.7	PLCopen Safety: Diagnose-Informationen ausgeben	81
8.4	Safety-Funktionen von CODESYS nutzen	82
8.4.1	SafetyStandard-Bibliothek	82

Inhalt

8.4.2	SafetyPLCopen-Bibliothek.....	83
8.4.3	SafetyFSoEMaster-Bibliothek.....	84
8.5	Datenaustausch zwischen Standard-SPS und fehlersicherer SPS.....	85
8.5.1	Hinweise zur Verwendung von Austauschobjekten.....	85
8.5.2	Werkseitig angelegte Austauschvariablen nutzen.....	86
8.5.3	Zusätzliche Objekte für Datenaustausch anlegen.....	88
8.5.4	Intervall des Datenaustauschs einstellen.....	88
8.6	Sichere Querkommunikation.....	89
8.6.1	Sichere Netzwerkvariablen einrichten.....	89
8.6.2	Sichere Netzwerkvariablen in Anwendung nutzen.....	91
8.7	Sichere Feldbuskommunikation.....	92
8.7.1	FSoE-Verbindung konfigurieren.....	92
8.7.2	Hinweise zu FB FSoEMaster.....	93
8.8	Prozesssicherheitszeit einstellen.....	94
8.8.1	Prozesssicherheitszeit beim AC4S.....	94
8.8.2	Variable Komponenten der Reaktionszeit einstellen.....	100
8.9	Safety-Task konfigurieren.....	101
8.9.1	Zykluszeit der fehlersicheren SPS einstellen.....	101
9	Betrieb	102
9.1	CODESYS-Projekt auf Gerät übertragen.....	102
9.1.1	Standard-Anwendung auf AC4S laden.....	103
9.1.2	Sichere Anwendung auf AC4S laden.....	104
9.1.3	Hinweis: Projekte mit Netzwerkvariablen.....	105
9.1.4	Codefolgen der sicheren AS-i Slaves einlernen (teachen).....	105
9.1.5	Boot-Applikation per SD-Karte löschen.....	106
9.2	Betriebszustände AC4S.....	107
9.2.1	Standard-SPS.....	107
9.2.2	Fehlersichere SPS.....	109
9.3	Reset.....	112
9.3.1	Standard-SPS.....	112
9.3.2	Fehlersichere SPS.....	114
9.4	Web-Visualisierung anzeigen.....	115
10	Anhang	116
10.1	Bibliothek ACnnnn_Utils.library.....	117
10.1.1	Übersicht: AS-i Funktionen (FB_ASi).....	118
10.1.2	Übersicht: System-Funktionen (FB_System).....	166
10.1.3	Aufzählungstypen und komplexe Variablen.....	173
10.2	Bibliothek ACnnnn_SYS_CMD.library.....	182
10.2.1	ACnnnn_SysCmd.....	183
10.3	Bibliothek SF_IO.library.....	190
10.3.1	CtrlASi_InSlave.....	191
10.3.2	CtrlASi_OutSlave.....	193
10.3.3	CtrlASi_ResetAllSlaves.....	196
10.3.4	CtrlLocalInputs.....	198
10.3.5	Ctrl_SetDiagInfo.....	200
10.3.6	GetLocalInput.....	203
10.3.7	SetLocalOutput.....	204
10.4	Bibliothek SF_LogicalInterfaces.library.....	206
10.4.1	Hinweise zu logischen Geräten.....	207
10.4.2	Logische Geräte für sichere AS-i Slaves.....	211
10.4.3	Logische Geräte für die lokale E/A-Schnittstelle.....	242

Inhalt

10.5	Fehler-Codes: Sichere AS-i Slaves	275
10.6	Fehler-Codes: Lokale E/A-Schnittstelle	277
11	Index	278
12	ifm weltweit • ifm worldwide • ifm à l'échelle internationale	281



1 Vorbemerkung

Inhalt	
Rechtliche Hinweise	6
Zweck des Dokuments	6
Verwendete Symbole und Formatierungen.....	7
Übersicht: Anwender-Dokumentation für AC4S.....	7
Übersicht: CODESYS-Dokumentation von 3S.....	8
Änderungshistorie.....	8

14801

1.1 Rechtliche Hinweise

1631

© Alle Rechte bei ifm electronic gmbh. Vervielfältigung und Verwertung dieser Anleitung, auch auszugsweise, nur mit Zustimmung der ifm electronic gmbh.

Alle auf unseren Seiten verwendeten Produktnamen, -Bilder, Unternehmen oder sonstige Marken sind Eigentum der jeweiligen Rechteinhaber:

- AS-i ist Eigentum der AS-International Association, (→ www.as-interface.net)
- CAN ist Eigentum der CiA (CAN in Automation e.V.), Deutschland (→ www.can-cia.org)
- CODESYS™ ist Eigentum der 3S – Smart Software Solutions GmbH, Deutschland (→ www.codesys.com)
- DeviceNet™ ist Eigentum der ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® ist Eigentum der →ODVA™
- EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland
- IO-Link® (→ www.io-link.com) ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- ISOBUS ist Eigentum der AEF – Agricultural Industry Electronics Foundation e.V., Deutschland (→ www.aef-online.org)
- Microsoft® ist Eigentum der Microsoft Corporation, USA (→ www.microsoft.com)
- PROFIBUS® ist Eigentum der PROFIBUS Nutzerorganisation e.V., Deutschland (→ www.profibus.com)
- PROFINET® ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Windows® ist Eigentum der →Microsoft Corporation, USA

1.2 Zweck des Dokuments

21675

Dieses Dokument gilt für folgende Geräte der Produktfamilie "SmartSPS SafeLine AC4S":

- SmartSPS SafeLine AC4S mit Profinet-Device-Schnittstelle (AC402S)
- SmartSPS SafeLine AC4S mit Profibus-Slave-Schnittstelle (AC412S)
- SmartSPS SafeLine AC4S mit EtherNet/IP-Device-Schnittstelle (AC422S)
- SmartSPS SafeLine AC4S mit EtherCAT-Slave-Schnittstelle (AC432S)

Dieses Dokument dient als Ergänzung zu den Gerätehandbüchern der o.g. Geräte.

Diese Anleitung beschreibt die Konfiguration und Programmierung der geräteinternen Standard-SPS und der fehlersicheren SPS des AC4S mithilfe des Programmiersystems CODESYS.

1.3 Verwendete Symbole und Formatierungen

13839

- ▶ ... Handlungsanweisung
- > ... Reaktion, Ergebnis
- ... Querverweis oder Internet-Link
- 123 Dezimalzahl
- 0x123 Hexadezimalzahl
- 0b010 Binärzahl
- [...] Bezeichnung von Tasten, Schaltflächen oder Anzeigen

1.4 Übersicht: Anwender-Dokumentation für AC4S

18655

Für die Modelle der Geräteklasse "SmartSPS SafeLine AC4S" stellt die ifm electronic die folgenden Anwender-Dokumentationen bereit:

Dokument	Inhalt / Beschreibung
Datenblatt	Technische Daten des AC4S in Tabellenform
Betriebsanleitung *	<ul style="list-style-type: none"> ▪ Hinweise zur Montage und elektrischen Installation des AC4S ▪ Inbetriebnahme, Beschreibung der Bedien- und Anzeigeelemente, Wartungshinweise, Maßzeichnung
Gerätehandbuch	<ul style="list-style-type: none"> ▪ Hinweise zur Bedienung des AC4S über GUI und Web-Interface ▪ Beschreibung der zyklischen und azyklischen Datensätze, der Feldbus-Parameter und der Kommandoschnittstelle ▪ Fehlerbeschreibung
Ergänzung Gerätehandbuch	<ul style="list-style-type: none"> ▪ Beschreibung der azyklischen Datensätze und der Kommandoschnittstelle
Programmierhandbuch	<ul style="list-style-type: none"> ▪ Erstellung eines Projekts mit dem Gerät in CODESYS ▪ Konfiguration des Geräts in CODESYS ▪ Programmierung der Standard-SPS des Geräts ▪ Programmierung der fehlersicheren SPS des Geräts ▪ Beschreibung der gerätespezifischen CODESYS-Funktionsbibliotheken

Legende:

*... Die Betriebsanleitung gehört zum Lieferumfang des Geräts.



Alle Dokumente können von der ifm-Webseite heruntergeladen werden.

1.5 Übersicht: CODESYS-Dokumentation von 3S

20259

Die 3S GmbH stellt für die Programmierung der Standard-SPS und der fehlersicheren SPS des AC4S folgende Anwender-Dokumentationen bereit:

Dokument	Inhalt / Beschreibung
Online-Hilfe	<ul style="list-style-type: none"> ▪ Kontextsensitive Hilfe ▪ Beschreibung der Programmiersystems CODESYS und der Safety-Erweiterung
CODESYS Installation und Erste Schritte	<ul style="list-style-type: none"> ▪ Hinweise zur Installation des Programmiersystems CODESYS ▪ Erste Schritte beim Umgang mit dem Programmiersystem CODESYS
Safety Anwenderhandbuch	<ul style="list-style-type: none"> ▪ Hinweise zur sicherheitsgerichteten Projektentwicklung mit CODESYS ▪ Hinweise zur sicherheitsgerichteten Programmierung der fehlersicheren SPS mit CODESYS

Alle Dokumente sind nach der Installation des Programmiersystems CODESYS 3.5 auf der Festplatte des PC/Laptops gespeichert und aufrufbar:

- Online-Hilfe:
... \Programme\3S CoDeSys\CoDeSys\Online-Help
- CODESYS Installation und Erste Schritte:
... \Programme\3S CoDeSys\CoDeSys\Documentation
- Safety Anwenderhandbuch:
... \Programme\3S CoDeSys\CoDeSys\Documentation

1.6 Änderungshistorie

21676

Ausgabe	Thema	Datum
00	Neuerstellung des Dokuments	12/2017

2 Sicherheitshinweise

Inhalt	
Allgemeine Sicherheitshinweise	9
Notwendige Vorkenntnisse	9
Verwendete Warnhinweise	10

213

2.1 Allgemeine Sicherheitshinweise

8516

Dieses Dokument vor Inbetriebnahme des Produktes lesen und während der Einsatzdauer aufbewahren.

Das Produkt nur bestimmungsgemäß verwenden.

Die Missachtung von Anwendungshinweisen oder technischen Angaben kann zu Sach- und / oder Personenschäden führen.

Unsachgemäße oder nicht bestimmungsgemäße Verwendung können zu Funktionsstörungen des Gerätes, zu unerwünschten Auswirkungen in der Applikation oder zum Verlust der Gewährleistungsansprüche führen.

Für Folgen durch Eingriffe in das Gerät oder Fehlgebrauch durch den Betreiber übernimmt der Hersteller keine Haftung.

- ▶ Angaben dieser Anleitung befolgen.
- ▶ Warnhinweise auf dem Gerät beachten.

2.2 Notwendige Vorkenntnisse

10317

Das Dokument richtet sich an Fachkräfte. Dabei handelt es sich um Personen, die aufgrund ihrer einschlägigen Ausbildung und ihrer Erfahrung befähigt sind, Risiken zu erkennen und mögliche Gefährdungen zu vermeiden, die der Betrieb oder die Instandhaltung eines Produkts verursachen kann.

Zum Programmieren der AC4S sollten die Personen zusätzlich mit der Software CODESYS 3.5 und der CODESYS Safety-Erweiterung vertraut sein.

Darüber hinaus sollten die Personen über Kenntnisse zu folgenden Themenkomplexen verfügen:

- Anforderungen an die sicherheitsgerichtete Programmierung
- Normen DIN EN ISO 13849 und DIN EN 62061

2.3 Verwendete Warnhinweise

13685

WARNUNG

Tod oder schwere irreversible Verletzungen sind möglich.

VORSICHT

Leichte reversible Verletzungen sind möglich.

ACHTUNG

Sachschaden ist zu erwarten oder möglich.



Wichtiger Hinweis
Fehlfunktionen oder Störungen sind bei Nichtbeachtung möglich



Information
Ergänzender Hinweis

3 Systemvoraussetzungen

Inhalt	
Hardware	11
Software.....	11
Lizenzierung	11

16903

3.1 Hardware

16904

- Gerät der Produktfamilie SmartSPS SafeLine AC4S mit Firmware V4.2.5
- PC/Laptop für Programmiersystem CODESYS (→ Systemvoraussetzungen CODESYS Development System V3.5)
- Ethernet-Verbindung zwischen CODESYS-PC/Laptop und Konfigurationsschnittstelle (X3) des Geräts

3.2 Software

20253

Um die geräteinterne Standard-SPS und die fehlersichere SPS des AC4S zu programmieren, werden folgende Software-Komponenten benötigt:

Komponente	Bedeutung	Version
CODESYS Development System	Programmiersystem CODESYS für normenkonforme SPS-Programmierung nach die IEC 61131-3	3.5 SP9 Patch 7 Hotfix 3
Package "CODESYS für ifm SmartPLC SafeLine"	<ul style="list-style-type: none"> ▪ Safety-Erweiterung für das Programmiersystem CODESYS ▪ Geräte- und Schnittstellenbeschreibung des AC4S ▪ Funktionsbibliotheken für Programmierung der Standard-SPS ▪ zertifizierte Funktionsbibliotheken für Programmierung der fehlersicheren SPS ▪ zertifizierte Bibliotheken für die sicherheitsgerichtete Programmierung (Safety PLCopen, Safety Standard) 	1.5.2.10



Die in diesem Dokument zugesicherten Eigenschaften und Funktionen sind nur mit den Software-Komponenten in den hier angegebenen Versionen erreichbar!

ifm electronic stellt auf seiner Webseite die benötigten Software-Komponenten zum Herunterladen bereit.

3.3 Lizenzierung

20254

Mit dem Kauf eines Geräts der SmartSPS SafeLine AC4S hat der Käufer gleichzeitig eine gültige Lizenz für die Nutzung des Programmiersystems CODESYS 3.5 erworben.

4 Installation

Inhalt

Programmiersoftware CODESYS	12
ifm AS-i Package	13
	17146

4.1 Programmiersoftware CODESYS

7282

Das CODESYS Development System (kurz: CODESYS) ist eine Entwicklungsumgebung für die Erstellung von SPS-Anwendungen (Applikationen) gemäß Norm IEC 61131-3.

4.1.1 CODESYS Development System installieren

18596

Um die Software "CODESYS Development System" zu installieren:

- ▶ Programmiersystem CODESYS 3.5 SP9 Patch 7 Hotfix 3 installieren (→ CODESYS Installation und Erste Schritte).
- > CODESYS 3.5 SP9 Patch 7 Hotfix 3 ist auf dem PC/Laptop installiert.

4.2 ifm AS-i Package

Inhalt	
Komponenten des ifm-Packages	13
ifm-Package installieren	14
ifm-Package deinstallieren	14

17679



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Package Manager
 - Online-Hilfe > CODESYS Development System > Packages und Lizenzen verwalten

4.2.1 Komponenten des ifm-Packages

10982

ifm electronic stellt für die Programmierung des AC4S das Package "CODESYS für ifm SmartPLC SafeLine" (kurz: ifm-Package) bereit. Das ifm-Package (Datei: ifm_SmartPLC_SafeLine_V1_5_2_10.package) enthält folgende zertifizierte Komponenten:

Komponente	Beschreibung
Plugins	CODESYS-Plugins
Dateien	Projektvorlagen, Projektbeispiele, Safety-Anwenderhandbuch CODESYS, CODESYS-Profile
Online-Hilfe-Dateien	Online-Hilfen (CODESYS Safety, Safety Netzvariablen)
Online-Hilfe-Merge-Dateien	Online-Hilfe-System
Gerätebeschreibung	Beschreibung der Gerätemodule, Schnittstellen und zusätzlicher Geräte
Bibliothek	CODESYS-Bibliotheken (SF_IO, SafetyPLCopen, SF_LogicalInterfaces, SafetyStandard, SafetySystemIO, SafetyNetVar, SafetyFSofEMaster, ACnnt_Utils, ACnntn_SYS_CMD, IoStandard, SysSafetyIoBase_ifm, IoDrvEtherCAT, IoDrvEtherNetIP, IoDrvAL1020, IoDrvAL1030)

4.2.2 ifm-Package installieren

6966

Um das ifm-Package auf dem Programmier-PC/Laptop zu installieren:

Voraussetzungen

- > CODESYS 3.5 SP9 Patch 7 Hotfix 3 ist auf dem Programmier-PC/Laptop installiert.

1 CODESYS starten

- ▶ CODESYS mit Administratorrechten starten.
- > CODESYS-Programmoberfläche erscheint.

2 ifm-Package installieren

- ▶ Mit [Tools] > [Package Manager] den Package Manager starten.
- > Fenster [Package Manager] erscheint.
- ▶ [Installieren...] klicken, um den Installationsdialog zu starten.
- ▶ ifm-Package `ifm_SmartPLC_SafeLine_V1_5_2_10.package` wählen und eine komplette Installation durchführen.
- > Fenster [Package Manager] zeigt installiertes ifm-Package.
- ▶ [Schließen] klicken, um den Package Manager beenden.



- ▶ Hinweis zum korrekten Start des Programmiersystems CODESYS beachten!
→ **CODESYS starten** (→ S. [15](#))

4.2.3 ifm-Package deinstallieren

12270

Um das installierte Package "CODESYS für ifm SmartPLC SafeLine" zu deinstallieren:

1 CODESYS starten

- ▶ CODESYS mit Administratorrechten starten.
- > CODESYS-Programmoberfläche erscheint.

2 ifm package deinstallieren

- ▶ [Tools] > [Package Manager] wählen, um den Package Manager aufzurufen.
- > Fenster [Package Manager] zeigt die installierten Packages.
- ▶ Kontrollfeld [Versionsnummer anzeigen] aktivieren.
- > Fenster zeigt Versionsnummern der installierten Packages.
- ▶ Package-Version wählen, die deinstalliert werden soll.
- ▶ Auf [Entfernen...] klicken, um das gewählte Package zu deinstallieren.
- > Gewählte Package-Version wird entfernt.
- ▶ Auf [Schließen] klicken, um den Package Manager zu beenden.

5 Schnelleinstieg

Inhalt	
CODESYS starten	15
CODESYS-Projekt erstellen	16
CODESYS-Dokumentation nutzen.....	20
Nutzerkonto anlegen	20
Programmierschnittstelle konfigurieren	21

15858

5.1 CODESYS starten

19358

ifm electronic stellt für CODESYS ein spezielles Profil bereit. Das CODESYS-Profil "ifm electronic SmartPLC SafeLine V3.5.9.73" erzeugt eine vorkonfigurierte Umgebung für die Konfiguration und Programmierung eines AC4S-Automatisierungssystems.

Um CODESYS mit dem Profil "ifm electronic SmartPLC SafeLine V3.5.9.73" zu starten:

Voraussetzungen

- > Software-Komponenten wurden korrekt installiert (→ **ifm-Package installieren** (→ S. [14](#)))

1 Desktop-Verknüpfung erstellen

- ▶ Die bei der Installation erzeugte Desktop-Verknüpfung von CODESYS löschen.
- ▶ Desktop-Verknüpfung der folgenden Anwendung erstellen:
[Start] > [Alle Programme] > [3S CODESYS] > [CODESYS] > [CODESYS without Profile]

2 CODESYS mit ifm-Profil starten

- ▶ Doppelklick auf Desktopverknüpfung [CODESYS without Profile]
- > Auswahlfenster erscheint.
- ▶ In Liste [Versionsprofil] den Wert [ifm electronic SmartPLC SafeLine V3.5.9.73] wählen.
- ▶ Mit [Weiter] die Auswahl übernehmen und das Profil laden.
- > CODESYS-Programmiersystem startet mit dem gewählten Profil.

5.2 CODESYS-Projekt erstellen

Inhalt	
Neues Projekt mit AC4S erstellen	17
Übersicht: Projektstruktur mit AC4S	19

17129



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - CODESYS-Projekt erstellen
→ Online-Hilfe > CODESYS Development System > Projekt anlegen und konfigurieren
 - Objekte der Benutzeroberfläche
→ Online-Hilfe > CODESYS Development System > Referenz Benutzeroberfläche

5.2.1 Neues Projekt mit AC4S erstellen

12225



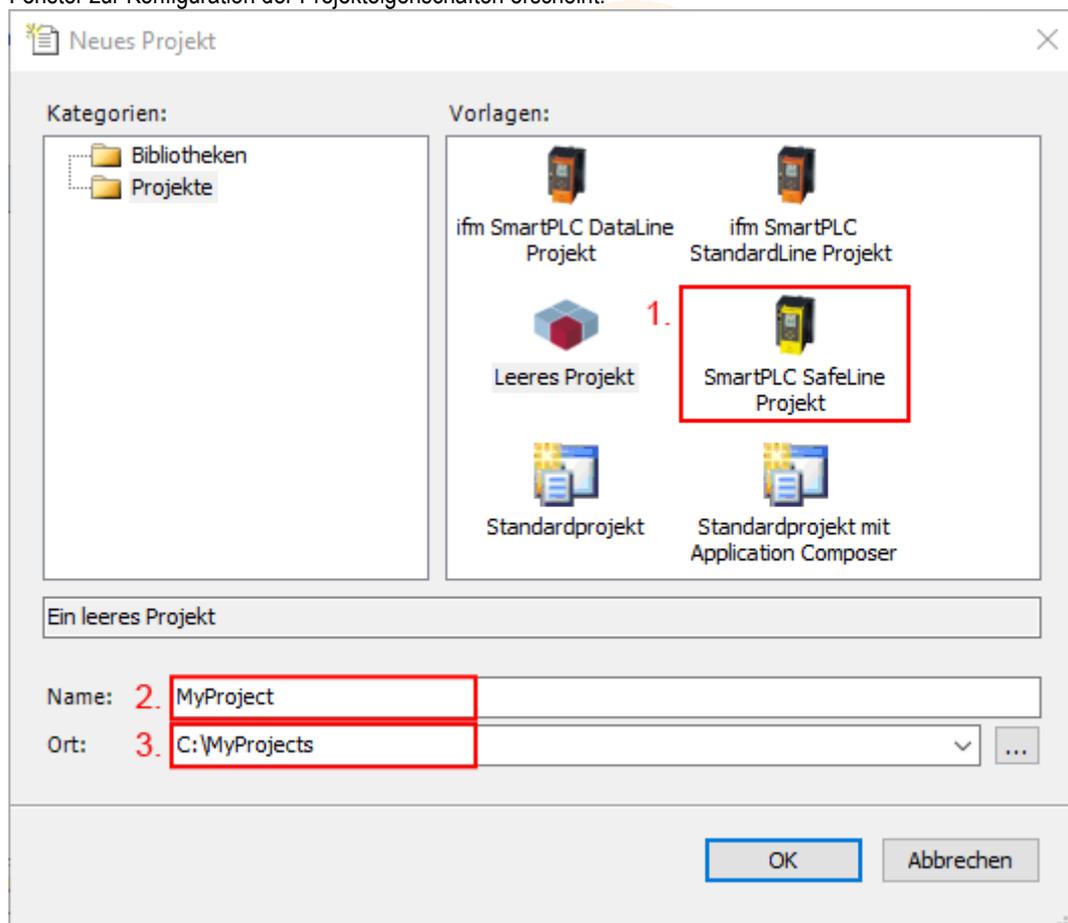
Um Fehler bei einer manuellen Systemkonfiguration zu vermeiden, wird ausdrücklich empfohlen, bei der Erstellung eines SmartSPS SafeLine AC4S-Projekts in CODESYS das Projekt-Template der ifm electronic zu verwenden.

Voraussetzungen:

- > Alle notwendigen Software-Komponenten sind korrekt installiert (→ **ifm-Package installieren** (→ S. 14)).
- > CODESYS wurde fehlerfrei gestartet (→ **CODESYS starten** (→ S. 15)).

1 Neues Projekt anlegen

- ▶ [Datei] > [Neues Projekt ...] wählen.
- > Fenster zur Konfiguration der Projekteigenschaften erscheint.



- ▶ Folgende Werte einstellen:
 1. [Vorlagen]: [ifm_SmartPLC_SafeLine Projekt] wählen.
 2. [Name]: Projektname eingeben
 3. [Ort]: Speicherort für Projektdatei wählen.
- ▶ [OK] klicken, um die eingegebenen Werte zu bestätigen.
- > CODESYS erzeugt ein neues Projekt mit einem AC4S.
- > Fenster [Geräte] zeigt den Gerätebaum des Projekts (→ **Übersicht: Projektstruktur mit AC4S** (→ S. 19)).

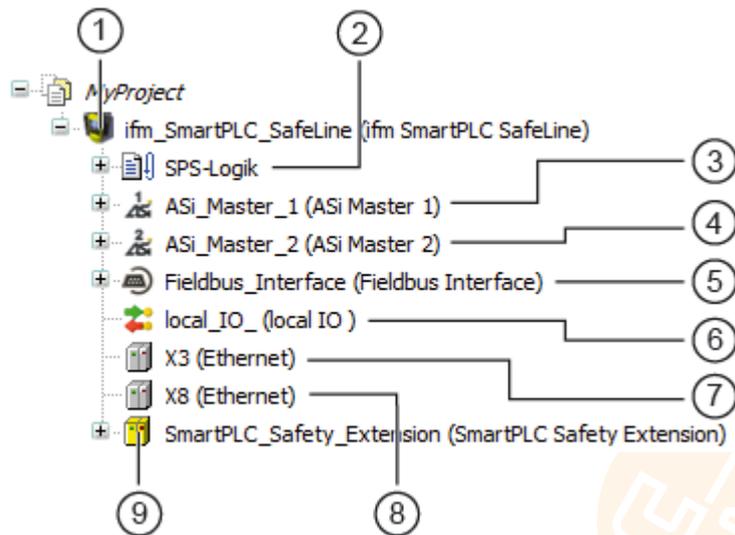
2 Projekt speichern

- ▶ [Datei] > [Projekt speichern] wählen.
- > CODESYS speichert das Projekt.



5.2.2 Übersicht: Projektstruktur mit AC4S

12275



- ① [ifm_SmartPLC_SafeLine (ifm SmartPLC SafeLine)] bietet Zugriff auf die Einstellungen der Standard-SPS des AC4S
→ **Standard-SPS konfigurieren** (→ S. 23)
- ② [SPS-Logik] enthält die Standard-SPS-Anwendungen des AC4S (→ **Objekte einer Standard-SPS-Anwendung** (→ S. 47))
- ③ [ASi_Master_1] enthält folgende Objekte:
 - Prozessdatenabbilder der Standard AS-i Slaves am AS-i Master 1 (→ **Auf Standard-Ein- und Ausgangsdaten zugreifen** (→ S. 50))
 - Konfigurationsknoten für sichere AS-i Slaves am AS-i Master 1 (→ **Sichere AS-i Slaves konfigurieren** (→ S. 25))
- ④ [ASi_Master_2] enthält folgende Objekte:
 - Prozessdatenabbilder der Standard AS-i Slaves am AS-i Master 2 (→ **Auf Standard-Ein- und Ausgangsdaten zugreifen** (→ S. 50))
 - Konfigurationsknoten für sichere AS-i Slaves am AS-i Master 2 (→ **Sichere AS-i Slaves konfigurieren** (→ S. 25))
- ⑤ [Fieldbus_Interface] bietet Zugriff auf die Ein- und Ausgangsdaten der Feldbus-Schnittstelle.
→ **Auf Standard-Ein- und Ausgangsdaten zugreifen** (→ S. 50)
- ⑥ [local_IO_] ist der Konfigurationsknoten für sichere Geräte an der lokalen E/A-Schnittstelle
→ **Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren** (→ S. 32)
- ⑦ [X3] bietet Zugriff auf die Konfigurationsoptionen der Konfigurationsschnittstelle 2 (X3)
→ **Feldbus-Funktionalität erweitern** (→ S. 42) und → **IP-Einstellungen ändern** (→ S. 40)
- ⑧ [X8] bietet Zugriff auf die Konfigurationsoptionen der Konfigurationsschnittstelle 2 (X8)
→ **Feldbus-Funktionalität erweitern** (→ S. 42) und → **IP-Einstellungen ändern** (→ S. 40)
- ⑨ [SmartPLC_Safety_Extension] bietet Zugriff auf folgende Objekte:
 - Einstellungen der fehlersicheren SPS des AC4S (→ **Fehlersichere SPS konfigurieren** (→ S. 24))
 - sichere Anwendung (→ **Objekte einer sicheren Anwendung** (→ S. 72))

5.3 CODESYS-Dokumentation nutzen

6989

Dieses Handbuch beschreibt ausschließlich die Integration, Konfiguration und Programmierung des AC4S mithilfe des Programmiersystems CODESYS.

Für die Beschreibung von Nutzeraktionen und Komponenten der Bedienoberfläche wird die CODESYS-eigene Terminologie verwendet.

Standard-Funktionen und -Mechanismen von CODESYS werden nicht beschrieben. Am Anfang jedes Abschnitts erfolgt ein Verweis auf die entsprechenden Kapitel der Online-Hilfe von CODESYS.

Um die Online-Hilfe von CODESYS aufzurufen:

- ▶ CODESYS starten.
- > Bedienoberfläche von CODESYS erscheint.
- ▶ [F1] drücken.
- > Online-Hilfe des Programmiersystems CODESYS erscheint.



- ▶ Machen Sie sich mit dem Programmiersystem CODESYS vertraut! Dies gilt insbesondere für folgende Themen:
 - Bezeichnungen und Funktionen der Elemente der Bedienoberfläche
 - Grundlegende Menüfunktionen
 - Programmiertechniken und Mechanismen zur Datenhaltung

5.4 Nutzerkonto anlegen

18968



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Safety Benutzerverwaltung und Zugriffskontrolle:
→ Online-Hilfe > Add-Ons > CODESYS Safety > Safety Benutzerverwaltung und Zugriffskontrolle
 - Safety Benutzerkonfiguration:
→ Online-Hilfe > Add-Ons > CODESYS Safety > Safety Benutzerkonfiguration

Safety-relevante Objekten und Funktionen eines Projekts dürfen nur von einem Nutzer ausgeführt werden, der zur Benutzergruppe "Safety" bzw "Safety.ExtendedLevel" gehört.

Um einen Nutzer anzulegen:

- ▶ [Projekt] > [Projekteinstellungen] wählen
- ▶ CODESYS-Benutzerverwaltung erscheint.
- ▶ Nutzer anlegen und der gewünschten Benutzergruppe hinzufügen.
- ▶ Passwort erzeugen.

5.5 Programmierschnittstelle konfigurieren

20231

Um das erstellte Projekt auf das AC4S zu übertragen, muss zwischen dem Programmiersystem CODESYS und dem AC4S ein gültiger Kommunikationspfad konfiguriert werden.

Standard-SPS und fehlersichere SPS des AC4S sind hardwaretechnisch getrennt. Deshalb ist es notwendig, die Kommunikationspfade für beide Steuerungen separat einzustellen.

5.5.1 Kommunikationspfad der Standard-SPS setzen

20233

Folgende Schnittstellen können als Programmierschnittstelle konfiguriert werden:

- Ethernet-Konfigurationsschnittstelle 1 (X3)
- Ethernet-Konfigurationsschnittstelle 2 (X8)

Um die Verbindung zwischen CODESYS-Programmiersoftware und der Standard-SPS des AC4S zu konfigurieren:

1 Vorbereitungen

- ▶ CODESYS-PC/Laptop und Konfigurationsschnittstelle 1 (X3) oder 2 (X8) des Geräts verbinden.
- ▶ Optional: IP-Einstellungen der Ethernet-Schnittstellen anpassen.

2 Kommunikationseinstellungen wählen

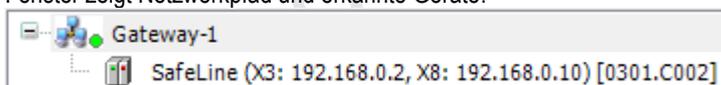
- ▶ Im Gerätebaum: Doppelklick auf Symbol [ifm_SmartPLC_SafeLine]
- ▶ Im Editor-Fenster: Auf [Kommunikationseinstellungen] klicken.
- > Editor-Fenster zeigt Kommunikationseinstellungen der SPS.

3 Gateway wählen

- ▶ In Liste [Gateway] das gewünschte Gateway wählen.
- > Liste zeigt gewählten Gateway.

4 Kommunikationspfad setzen

- ▶ Auf [Netzwerk durchsuchen...] klicken.
- > Fenster [Gerät auswählen] erscheint.
- ▶ Gateway-Knoten wählen und auf [Gerät suchen] klicken, um den Scan-Vorgang zu starten.
- > CODESYS durchsucht Netzwerk nach Geräten.
- > Fenster zeigt Netzwerkpfad und erkannte Geräte.



- ▶ Knoten des AC4S wählen.
- > Informationsfeld zeigt Detailinformationen über gewählten Knoten.
- ▶ Mit [OK] den Kommunikationspfad zur Standard-SPS setzen.
- > CODESYS kann Daten auf die Standard-SPS des AC4S übertragen.

5.5.2 Kommunikationspfad der fehlersicheren SPS setzen

20232

Um die Verbindung zwischen CODESYS-Programmiersoftware und der fehlersicheren SPS des AC4S zu konfigurieren:

Voraussetzungen:

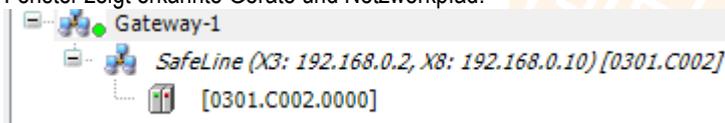
- > Kommunikationspfad zur Standard-SPS ist korrekt eingestellt (→ **Kommunikationspfad der Standard-SPS setzen** (→ S. 21)).

1 Kommunikationseinstellungen wählen

- ▶ Im Projektbaum: Doppelklick auf Symbol [SmartPLC_Safety_Extension]
- > Editor-Fenster zeigt Registerkarte [Kommunikationseinstellungen].

2 Kommunikationspfad setzen

- ▶ [Netzwerk durchsuchen...] aktivieren.
- > Fenster [Gerät auswählen] erscheint.
- ▶ Knoten [Gateway-1] markieren und mit [Gerät suchen] den Scan-Vorgang starten.
- > Gerät durchsucht Netzwerk nach Geräten.
- > Fenster zeigt erkannte Geräte und Netzwerkpfad.



- ▶ Im Netzwerkpfad den Unterknoten des SafeLine-Knotens wählen.
- > Informationsfeld zeigt Detailinformationen über gewählten Knoten.
- ▶ Mit [OK] den Kommunikationspfad zur fehlersicheren SPS setzen.

3 Verbindung zur fehlersicheren SPS bestätigen

- > Fenster [Verbindung zur Sicherheitssteuerung] erscheint.
- ▶ In Feld [Instanzidentifikation] die Seriennummer des Geräts eingeben.
- ▶ Auf [OK] klicken, um die Eingabe zu bestätigen.
- > CODESYS verifiziert die Verbindung zur fehlersicheren SPS.
- > CODESYS kann sich auf die fehlersichere SPS einloggen.

6 Systemkonfiguration

Inhalt	
Standard-SPS konfigurieren.....	23
Fehlersichere SPS konfigurieren.....	24
Sichere AS-i Slaves konfigurieren.....	25
Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren.....	32
Standard-Geräte an lokaler E/A-Schnittstelle konfigurieren.....	40
IP-Einstellungen ändern.....	40
Zusätzliches Gerät zum Projekt hinzufügen.....	41
Feldbus-Funktionalität erweitern.....	42

18964

Dieses Kapitel enthält Informationen über die Konfiguration der geräteinternen SPS, der eingebundenen sicheren AS-i Slaves, der sicheren Sensoren/Aktuatoren an der lokalen E/A-Schnittstelle und der Ethernet-Konfigurationschnittstelle.

6.1 Standard-SPS konfigurieren

18961



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Generischer Geräteeditor
→ Online-Hilfe > CODESYS Development System > Referenz Benutzeroberfläche > Objekte > Objekt 'Gerät' und Generischer Geräteeditor

Die Konfiguration der Standard-SPS erfolgt über den "Generischen Geräteeditor" des CODESYS-Programmiersystems. Der Programmierer kann über folgenden Knoten im Gerätebaum auf den Geräteeditor der Standard-SPS zugreifen:

 ifm_SmartPLC_SafeLine (ifm SmartPLC SafeLine)

Um die geräteinterne Standard-SPS zu konfigurieren:

- ▶ Im Gerätebaum: Doppelklick auf [ifm_SmartPLC_SafeLine]
- > Editor-Fenster zeigt Geräte-Editor der geräteinternen Standard-SPS.
- ▶ Standard-SPS konfigurieren.
- ▶ Projekt speichern, um die Änderungen zu übernehmen.

6.2 Fehlersichere SPS konfigurieren

18960



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Editor der Sicherheitssteuerung
→ Online-Hilfe > Add-Ons > CODESYS Safety > Editoren > Editor der Sicherheitssteuerung

Die Konfiguration der fehlersicheren SPS des AC4S erfolgt über den "Geräteeditor der Sicherheitssteuerung" des CODESYS-Programmiersystems. Der Programmierer kann über folgenden Knoten im Gerätebaum auf den Geräteeditor der fehlersicheren SPS zugreifen:

 SmartPLC_Safety_Extension (SmartPLC Safety Extension)

Um die fehlersichere SPS des AC4S zu konfigurieren:

- ▶ Im Gerätebaum: Doppelklick auf [SmartPLC_Safety_Extension (SmartPLC Safety Extension)]
- > Editor-Fenster zeigt Geräteeditor der fehlersicheren SPS des AC4S.
- ▶ Fehlersichere SPS wie gewünscht konfigurieren.



Die Schaltfläche [Firmware Update] in der Registerkarte [Sicherheitssteuerung] besitzt im Zusammenhang mit dem AC4S keine Funktionalität!

- ▶ Schaltfläche nicht aktivieren!

Ein Update der Firmware des AC4S darf nur durch autorisierte Mitarbeiter der ifm electronic durchgeführt werden!

- ▶ Kontaktieren Sie ihren AS-i Fachberater für weitere Informationen.
- ▶ Nach einem Firmware-Update muss der Anwender die Bootapplikation aus sicherheitstechnischen Gesichtspunkten neu bewerten, übersetzen und auf das Gerät laden.

6.3 Sichere AS-i Slaves konfigurieren

Inhalt	
Hinweise	25
Sichere AS-i Slaves in Projekt einbinden	27
Sichere AS-i Slaves aus Projekt entfernen	31

18974



Die Konfiguration der sicheren Ein- und Ausgänge erfolgt über den "Geräte-Editor für logische E/As" des Programmiersystems CODESYS Safety. In den folgenden Abschnitten werden nur die AC4S-spezifischen Funktionen beschrieben!

- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Geräteeditor für logische E/As
→ Online-Hilfe > Add-Ons > CODESYS Safety > Editoren > Geräteeditor für logische E/As

6.3.1 Hinweise

20227

Konzept der logischen Geräte

20230

Sichere AS-i Slaves werden über sogenannte logische Geräte in ein CODESYS-Projekt eingebunden. Je nach Art des sicheren AS-i Slaves erfüllen die logischen Geräte unterschiedliche Funktionen.

Sichere AS-i Eingangs-Slaves

20228

Abgebildet auf das klassische Verhalten einer Speicherprogrammierbaren Steuerung übernimmt das logische Gerät sowohl Funktionen der Signalerfassung als auch der Signalverarbeitung. Im Einzelnen sind dies folgende Funktionen:

- die vom sicheren AS-i Eingangs-Slaves generierten Code-Halbfolgen erfassen
- die Code-Halbfolgen mithilfe der internen Logik vorverarbeiten
- das Ergebnis der logischen Vorverarbeitung als sicheren Variablenwert bereitstellen

Die logische Vorverarbeitung kann der Programmierer über das Parameter-Interface des logischen Geräts konfigurieren.

Sichere AS-i Ausgangs-Slaves

20229

Sichere AS-i Ausgangs-Slaves werden indirekt über einen virtuellen AS-i Control Slave angesteuert. Der AS-i Control Slave übernimmt folgende Funktionen:

- eine sichere Code-Folge für die Freigabe eines sicheren AS-i Ausgangs-Slaves erzeugen



Der AC4S erkennt den AS-i Control Slave erst nach dem Herunterladen des Safety-Projekts auf den AC4S. Der betreffende AS-i Master erzeugt daraufhin einen Konfigurationsfehler (Unbekannter Slave).

- ▶ Projektierungsabgleich durchführen, um den Fehler zu beseitigen (→ Gerätehandbuch, **Quick-Setup: AS-i Netzwerke projektieren**)

Verfügbare logische Geräte für sichere AS-i Slaves

18987

Für sichere AS-i Slaves stehen folgende logische Geräte zur Verfügung:

Bezeichnung	Typ	Referenz
SF_IN_ASi_forced	Eingang	→ SF_IN_ASi_forced (→ S. 212)
SF_IN_ASi_independent	Eingang	→ SF_IN_ASi_independent (→ S. 216)
SF_IN_ASi_conditionally_dependent	Eingang	→ SF_IN_ASi_conditionally_dependent (→ S. 220)
SF_IN_ASi_dependent	Eingang	→ SF_IN_ASi_dependent (→ S. 224)
SF_IN_ASi_dependent_filter_w_testreq	Eingang	→ SF_IN_ASi_dependent_filter_w_testreq (→ S. 228)
SF_IN_ASi_dependent_filter_ntestreq	Eingang	→ SF_IN_ASi_dependent_filter_ntestreq (→ S. 232)
SF_IN_ASi_dependent_filter_nshutdown	Eingang	→ SF_IN_ASi_dependent_filter_nshutdown (→ S. 236)
SF_OUTcontrol_ASi	Ausgang	→ SF_OUTcontrol_ASi (→ S. 240)



Detaillierte Informationen zu den logischen Geräten:
→ **Hinweise zu logischen Geräten** (→ S. [207](#))

6.3.2 Sichere AS-i Slaves in Projekt einbinden

8964

Sichere AS-i Slaves werden in 2 Schritten in ein CODESYS-Projekt eingebunden:

1. Logisches Gerät des sicheren AS-i Slaves zum Projektbaum hinzufügen:
→ **Logisches Gerät zum Projektbaum hinzufügen** (→ S. 27)
2. Logisches Gerät konfigurieren:
→ **Logisches Gerät konfigurieren** (→ S. 29)



Die Auswahl der passenden logischen Geräte liegt in der Verantwortung des Anwenders. Der Anwender muss sicherstellen, dass die gewählten logischen Geräte die für die zu realisierende Sicherheitsanwendung notwendige Funktionalität bereitstellen.

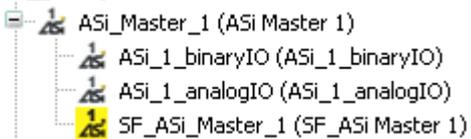
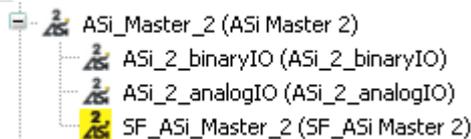
- Dokumentation der logischen Geräte beachten (→ **Logische Geräte für sichere AS-i Slaves** (→ S. 211))!

Ein logisches Gerät mit einer definierten AS-i Adresse darf innerhalb eines Safety-Projekts nur ein einziges Mal eingebunden werden!

Logisches Gerät zum Projektbaum hinzufügen

14806

Sichere AS-i Slaves werden im Standardbereich des Gerätebaums über die Knoten [SF_ASi_Master_1] und [SF_ASi_Master_2] in das CODESYS-Projekts eingebunden:

- Sichere AS-i Slaves an AS-i Master 1:

- Sichere AS-i Slaves an AS-i Master 2:


Um einen sicheren AS-i Eingangs- oder Ausgangs-Slave in das Projekt einzubinden:

Voraussetzungen:

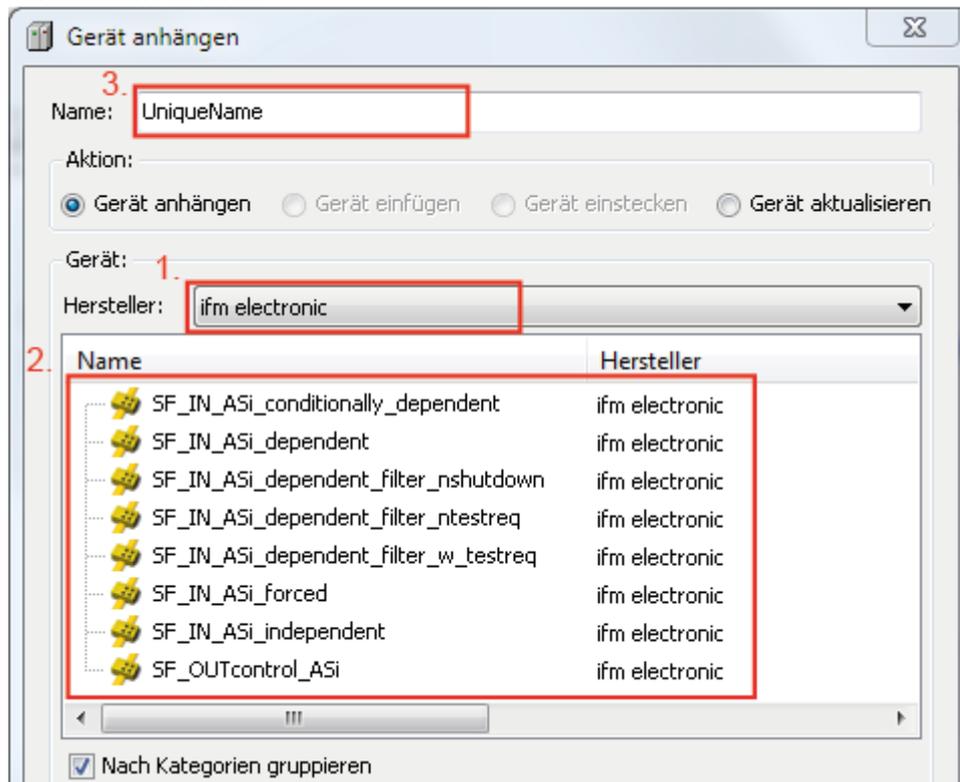
- > Sicherer AS-i Slave ist korrekt installiert und adressiert.
- > AS-i Netzwerk ist projektiert.

1 AS-i Netzwerk wählen

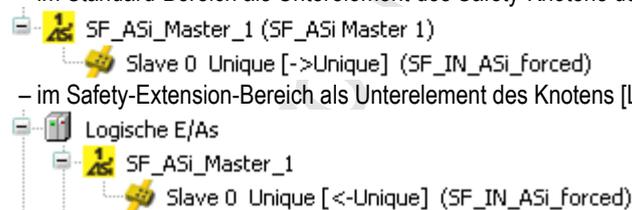
- Im Standardbereich des Gerätebaums: Safety-Knoten des AS-i Netzwerks markieren, in dem der sichere AS-i Slave installiert ist:
 - für AS-i 1: [ASi_Master_1] > [SF_ASi_Master_1]
 - für AS-i 2: [ASi_Master_2] > [SF_ASi_Master_2]

2 Logisches Gerät wählen

- ▶ [Projekt] > [Gerät anhängen...] wählen.
- > Auswahlfenster erscheint.



- ▶ Im Bereich [Gerät] folgende Werte einstellen:
 1. [Hersteller]: ifm electronic
 2. Tabelle: gewünschtes logisches Gerät markieren (→ Hinweis)
 3. [Name]: eindeutige Bezeichnung für logisches Gerät eingeben
- ▶ Mit [Gerät anhängen] die eingestellten Werte übernehmen.
- > CODESYS fügt das logische Gerät an folgenden Positionen des Gerätebaums ein:
 - im Standard-Bereich als Unterelement des Safety-Knotens des gewählten AS-i Netzwerks
 - im Safety-Extension-Bereich als Unterelement des Knotens [Logische E/As]



- ▶ Mit [Schließen] das Auswahlfenster schließen.

3 AS-i Adresse zuweisen

- ▶ Im Standardbereich des Gerätebaums:
 - Doppelklick auf das hinzugefügte logische Gerät
- > Editor-Fenster zeigt Registerkarte mit Konfigurationsoptionen des logischen Geräts.
- ▶ Registerkarte [Safe ASi IO Konfiguration] wählen.
- > Tabelle zeigt Parameter des logischen Geräts.
- ▶ In Spalte [Wert] die Adresse des sicheren AS-i Slaves eingeben.
- > Logisches Gerät ist an den sicheren AS-i Slave gekoppelt.
- > Symbol im Gerätebaum zeigt zugewiesene AS-i Adresse.

4 Optional: Weitere logische Geräte hinzufügen

- ▶ Schritte 1 bis 3 wiederholen, um zusätzliche logische Geräte zum Gerätebaum hinzuzufügen.



Hinweise zur Konfiguration des logischen Geräts: → **Logisches Gerät konfigurieren** (→ S. [29](#))

Hinweise zum Entfernen des logischen Geräts: → **Sichere AS-i Slaves aus Projekt entfernen** (→ S. [31](#))

Logisches Gerät konfigurieren

18978

Die Konfiguration erfolgt über den Knoten [Logische E/As] im Safety-Extension-Bereich des Gerätebaums. Alle in das Projekt eingebundenen sicheren AS-i Slaves sind dort als Unterknoten gelistet.

Um das logische Gerät eines sicheren AS-i Slaves zu konfigurieren:

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
Doppelklick auf das Symbol des logischen Geräts unter [Safety Logik] > [SafetyApp] > [Logische E/As]
- > Editor-Fenster zeigt folgende Optionen:

Registerkarte	Bedeutung
[Sichere Konfiguration]	Parameter-Interface des logischen Geräts (→ Parameter des logischen Geräts einstellen (→ S. 30))
[I/O Abbild]	Variablen-Abbild für den Zugriff aus Safety-Anwendung (→ Sicheres Prozesssignal auf Variable abbilden (→ S. 30))
[Information]	Informationen über den sicheren AS-i Slave (→ Online-Hilfe CODESYS Safety)

Parameter des logischen Geräts einstellen

7096

Die Registerkarte [Sichere Konfiguration] ermöglicht den Zugriff auf das Parameter-Interface des logischen Geräts. Anzahl und Art der verfügbaren Parameter ist abhängig vom gewählten logischen Gerät.



Unsachgemäß eingestellte Parameter können zu unerwünschtem Verhalten des Systems führen.

- ▶ Machen Sie sich vor der Konfiguration mit den Parametern des logischen Geräts vertraut (→ **Logische Geräte für sichere AS-i Slaves** (→ S. [211](#)))!



Alle Zeitangaben besitzen eine Ungenauigkeit von +/- 1 SPS-Zykluszeit. Der Anwender kann die Zykluszeit der geräteinternen fehlersicheren SPS im Rahmen der vorgegebenen Grenzen frei einstellen.

- ▶ Bei der Dimensionierung der folgenden Parameter die eingestellte SPS-Zykluszeit beachten!
 - Stabilisierungszeit (StabilisingTime)
 - Synchronisationszeit (SynchronisingTime)
 - Toleranzzeit (ToleranceTime)

Um das logische Gerät zu konfigurieren:

- > Im Geräte-Editor die Registerkarte [Sichere Konfiguration] wählen.
- > Editor-Fenster zeigt Tabelle mit verfügbaren Parametern des logischen Geräts.
- ▶ In Spalte [Werte] für alle angezeigten Parameter die gewünschten Parameterwerte eingeben.
- > Eingegebenen Werte werden übernommen.

Sicheres Prozesssignal auf Variable abbilden

20246

Das logische Gerät erzeugt ein sicheres Prozesssignal. Um dieses Signal in der Safety-Anwendung nutzen zu können, muss es auf eine Variable abgebildet werden.

Um das erzeugte Prozesssignal eines logischen Geräts auf eine sichere Variable abzubilden:

- ▶ Im Geräte-Editor die Registerkarte [I/O Abbild] wählen.
- > Editor-Fenster zeigt Tabelle mit den Mapping-Einstellungen.
- ▶ Im Variablenfeld von Bit 0 den Namen der Variable eingeben, auf die das sichere Prozesssignal abgebildet werden soll.
- > CODESYS erzeugt mit der eingegebenen Bezeichnung eine globale Variable vom Typ SAFEBOOL.
- > Safety-Anwendung kann über die Variable auf das sichere Prozesssignal zugreifen.



Die erzeugte Variable ist in der Globale-Variablen-Liste (GVL) IoConfigRemote_Globals abgelegt.

- ▶ Bei der Benennung der Variablen immer eindeutige Bezeichnungen wählen!

6.3.3 Sichere AS-i Slaves aus Projekt entfernen

17125

Um einen sicheren AS-i Slave vollständig aus dem Projekt zu entfernen, müssen die 2 Instanzen des logischen Geräts im Projektbaum separat gelöscht werden.

1 Logisches Gerät im Standard-Bereich entfernen

- ▶ Logisches Gerät im Standardbereich des Gerätebaums markieren.
- ▶ [Bearbeiten] > [Löschen] wählen.
- > CODESYS entfernt das logische Gerät aus dem Standard-Bereich.

2 Logisches Gerät im Safety-Extension-Bereich entfernen

- ▶ Logisches Gerät im Safety-Extension-Bereich des Gerätebaums markieren.
- ▶ [Bearbeiten] > [Löschen] wählen.
- > CODESYS entfernt das logische Gerät aus dem Safety-Extension-Bereich.



6.4 Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren

Inhalt	
Hinweise	32
Sicheres lokales Gerät in Projekt einbinden.....	34
Testsignal an lokalem Ausgang erzeugen	39
Sicheres lokales Gerät aus Projekt entfernen	39

18972



Die Konfiguration der sicheren Ein- und Ausgänge erfolgt über den "Geräte-Editor für logische E/As" von CODESYS Safety. In den folgenden Abschnitten werden nur die AC4S-spezifischen Funktionen beschrieben!

- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Geräteeditor für logische E/As
→ Online-Hilfe > Add-Ons > CODESYS Safety > Editoren > Geräteeditor für logische E/As

6.4.1 Hinweise

20227

Konzept der logischen Geräte

18963

Sichere Geräte an der lokalen E/A-Schnittstelle werden über sogenannte logische Geräte in ein CODESYS-Projekt eingebunden. Je nach Art des sicheren Geräts erfüllen die logischen Geräte unterschiedliche Funktionen.

Sichere Geräte an lokalen Eingängen

18984

Abgebildet auf das klassische EVA-Konzept einer Speicherprogrammierbaren Steuerung übernimmt das logische Gerät dabei sowohl Funktionen der Signalerfassung als auch der Signalverarbeitung. Im Einzelnen sind dies folgende Funktionen:

- die von den sicheren Geräten generierten Signale erfassen
- die Signale mithilfe der internen Logik vorverarbeiten
- das Ergebnis der logischen Vorverarbeitung als sicheren Variablenwert bereitstellen

Die logische Vorverarbeitung kann der Programmierer über das Parameter-Interface des logischen Geräts konfigurieren.

Sichere Geräte an lokalen Ausgängen

18966

Sichere Geräte an den lokalen Ausgängen können mit logischen Geräten angesteuert werden. Das logische Gerät übernimmt folgende Funktionen:

- ein sicheres Signal erzeugen
- das sichere Signal an den gewählten Ausgangskanälen der lokalen E/A-Schnittstelle ausgeben

Verfügbare logische Geräte für sichere Ein-/Ausgänge

18983

Für sichere Geräte an der lokalen E/A-Schnittstelle stehen folgende logische Geräte zur Verfügung:

Bezeichnung	E/A-Typ	Referenz
SF_IN_local_forced	Eingang	→ SF_IN_local_forced (→ S. 243)
SF_IN_local_independent	Eingang	→ SF_IN_local_independent (→ S. 247)
SF_IN_local_conditionally_dependent	Eingang	→ SF_IN_local_conditionally_dependent (→ S. 251)
SF_IN_local_dependent	Eingang	→ SF_IN_local_dependent (→ S. 255)
SF_IN_local_dependent_filter_testreq	Eingang	→ SF_IN_local_dependent_filter_testreq (→ S. 259)
SF_IN_local_dependent_filter_ntestreq	Eingang	→ SF_IN_local_dependent_filter_ntestreq (→ S. 263)
SF_IN_local_dependent_filter_nshutdown	Eingang	→ SF_IN_local_dependent_filter_nshutdown (→ S. 267)
SF_OUT_local_single	Ausgang	→ SF_OUT_local_single (→ S. 272)
SF_OUT_local_dual	Ausgang	→ SF_OUT_local_dual (→ S. 273)
SF_OUT_local_testpulse	Ausgang	→ SF_OUT_local_testpulse (→ S. 274)



Detaillierte Informationen zu den logischen Geräten:
→ [Hinweise zu logischen Geräten](#) (→ S. [207](#))

6.4.2 Sicheres lokales Gerät in Projekt einbinden

18979

Sichere Geräte an der lokalen E/A-Schnittstelle werden in 2 Schritten in ein CODESYS-Projekt eingebunden:

1. Logisches Gerät zum Projektbaum hinzufügen:
→ **Logisches Gerät zum Projektbaum hinzufügen** (→ S. [34](#))
2. Logisches Gerät konfigurieren:
→ **Logisches Gerät konfigurieren** (→ S. [37](#))



Die Auswahl der passenden logischen Geräte liegt in der Verantwortung des Anwenders. Der Anwender muss sicherstellen, dass die gewählten logischen Geräte die für die zu realisierende Sicherheitsanwendung notwendige Funktionalität bereitstellen.

- ▶ Dokumentation der logischen Geräte beachten (→ **Logische Geräte für die lokale E/A-Schnittstelle** (→ S. [242](#)))!

Ein logisches Gerät mit definierten Eingangskanälen bzw. Ausgangskanälen darf innerhalb eines Safety-Projekts nur ein einziges Mal eingebunden werden!

Logisches Gerät zum Projektbaum hinzufügen

18982

Sichere Geräte an der lokalen E/A-Schnittstelle werden im Standardbereich des Gerätebaums über den Knoten [local_IO_] in das Projekt eingebunden:

 local_IO_ (local IO)



Die Auswahl der passenden logischen Geräte obliegt dem Anwender. Der Anwender muss sicherstellen, dass die gewählten logischen Geräte die für die zu realisierende Sicherheitsanwendung notwendige Funktionalität bereitstellen.

- ▶ Dokumentation der logischen Geräte beachten (→ **Logische Geräte für die lokale E/A-Schnittstelle** (→ S. [242](#)))!

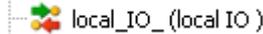
Ein logisches Gerät eines Typs mit 2 definierten Eingangskanälen (Input_Channel_A, Input_Channel_B) darf innerhalb eines Safety-Projekts nur ein einziges Mal eingebunden werden!

Voraussetzung:

- > Geräte sind entsprechend der gewünschten Sicherheitseinstufung mit der lokalen E/A-Schnittstelle verdrahtet.

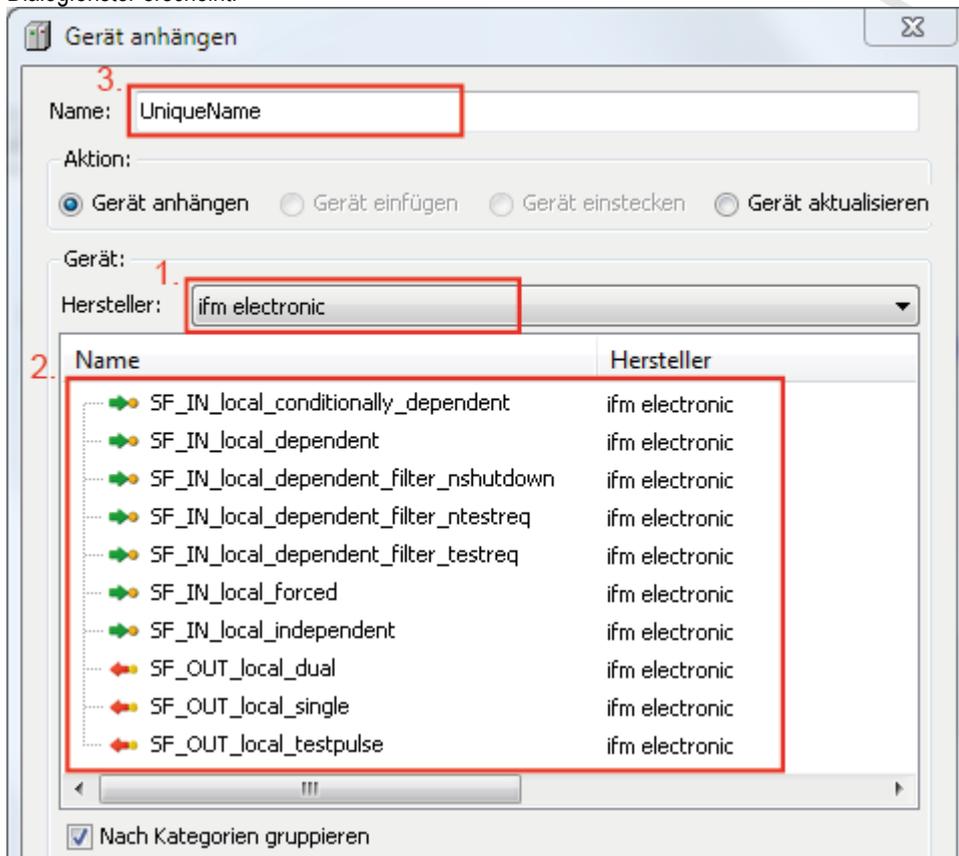
1 Logisches Gerät hinzufügen

- ▶ Im Standardbereich des Gerätebaums: Knoten [local_IO_] wählen:



- ▶ [Projekt] > [Gerät anhängen...] wählen.

- > Dialogfenster erscheint:

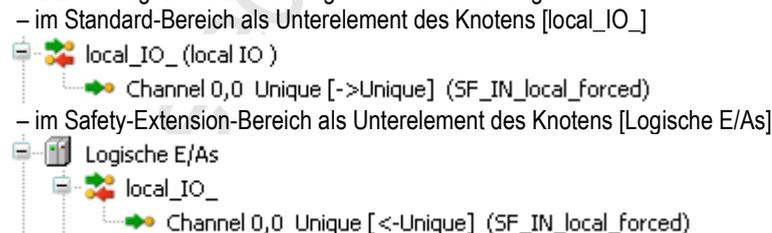


- ▶ In Gruppe [Gerät] folgende Parameter einstellen:

1. [Hersteller]: ifm electronic wählen.
2. Tabelle: gewünschtes logisches Gerät markieren (→ Hinweis)
3. [Name]: Eindeutigen Instanznamen eingeben.

- ▶ Mit [Gerät anhängen] die eingestellten Werte übernehmen.

- > CODESYS fügt Instanzen des logischen Geräts an folgenden Positionen des Gerätebaums ein:



- ▶ Mit [Schließen] das Dialogfenster schließen.



2 Ein- bzw. Ausgangskanäle zuweisen

- ▶ Im Standardbereich des Gerätebaums:
Doppelklick auf das hinzugefügte logische Gerät
- > Editor-Fenster zeigt Konfigurationsoptionen des logischen Geräts.
- ▶ Registerkarte [Safe Local IO Konfiguration] wählen.
- > Tabelle zeigt in Abhängigkeit des gewählten logischen Geräts folgende Parameter:

Schnittstellen-Typ	Parameter	Bedeutung
Eingang (2-kanalig)	Input_Channel_A	Lokaler Eingang, der mit Sensorkanal A verbunden ist.
	Input_Channel_B	Lokaler Eingang, der mit Sensorkanal B verbunden ist.
Ausgang (1-kanalig)	Output_Channel	Lokaler Ausgang, der mit dem Aktuator verbunden ist bzw. an dem das Testsignal ausgegeben werden soll.
Ausgang (2-kanalig)	Output_Channel_A	Lokaler Ausgang, der mit Aktuator kanal A verbunden ist.
	Output_Channel_B	Lokaler Ausgang, der mit Aktuator kanal B verbunden ist.

- ▶ In der Spalte [Wert] für jeden Parameter den gewünschten Wert eingeben.
- > Logisches Gerät ist mit den eingestellten Ein- bzw. Ausgangskanälen gekoppelt.
- > Symbol im Gerätebaum zeigt zugewiesene E/A-Kanäle.

3 Optional: Weitere logische Geräte hinzufügen

- ▶ Schritte 1 bis 2 wiederholen, um zusätzliche logische Geräte zum Gerätebaum hinzuzufügen.



Hinweise zur Konfiguration der logischen Geräte → **Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren** (→ S. [32](#))

Hinweise zum Entfernen des logischen Geräts: → **Sicheres lokales Gerät aus Projekt entfernen** (→ S. [39](#))

Logisches Gerät konfigurieren

18980

Die Konfiguration erfolgt über den Knoten [Logische E/As] im Safety-Extension-Bereich des Gerätebaums. Alle in das Projekt eingebundenen sicheren lokalen Geräte sind dort als Unterknoten gelistet.

Um das logische Gerät eines sicheren Sensors an einem lokalen Eingangskanal zu konfigurieren:

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
Doppelklick auf das Symbol des logischen Geräts unter [Safety Logik] > [SafetyApp] > [Logische E/As]
- > Editor-Fenster zeigt folgende Optionen:

Registerkarte	Bedeutung / Referenz
[Sichere Konfiguration]	Parameter-Interface des logischen Geräts (→ Parameter des logischen Geräts einstellen (→ S. 37))
[I/O Abbild]	Variablen-Abbild für den Zugriff aus Safety-Anwendung (→ Sicheres Prozesssignal auf Variable abbilden (→ S. 38))
[Information]	Informationen über das sichere Peripheriegeräts (→ CODESYS-Hilfe)

Parameter des logischen Geräts einstellen

18973

Die Registerkarte [Sichere Konfiguration] ermöglicht den Zugriff auf das Parameter-Interface des logischen Geräts. Anzahl und Art der verfügbaren Parameter sind abhängig vom gewählten logischen Gerät.



- Unsachgemäß eingestellte Parameter können zu unerwünschtem Verhalten des System führen.
- ▶ Machen Sie sich vor der Konfiguration mit den Parametern des logischen Geräts vertraut (→ **Logische Geräte für die lokale E/A-Schnittstelle** (→ S. [242](#)))!



- Alle Zeitangaben besitzen eine Ungenauigkeit von +/- 1 SPS-Zykluszeit. Der Anwender kann die Zykluszeit der geräteinternen fehlersicheren SPS im Rahmen der vorgegebenen Grenzen frei einstellen.
- ▶ Bei der Dimensionierung der folgenden Parameter die eingestellte SPS-Zykluszeit beachten!
 - Stabilisierungszeit (StabilisingTime)
 - Synchronisationszeit (SynchronisingTime)
 - Toleranzzeit (ToleranceTime)

Um das logische Gerät zu konfigurieren:

- ▶ Im Geräte-Editor die Registerkarte [Sichere Konfiguration] wählen.
- > Editor-Fenster zeigt Tabelle mit verfügbaren Parametern des logischen Geräts.
- ▶ In Spalte [Werte] für alle angezeigten Parameter die gewünschten Werte eingeben.
- > Eingegebene Werte werden übernommen.

Sicheres Prozesssignal auf Variable abbilden

18989

Das logische Gerät erzeugt ein sicheres Prozesssignal. Um dieses Signal in der Safety-Anwendung nutzen zu können, muss es auf eine Variable abgebildet werden.

Um das erzeugte Prozesssignal eines logischen Geräts auf eine sichere Variable abzubilden:

- ▶ Im Geräte-Editor die Registerkarte [I/O Abbild] wählen.
- > Editor-Fenster zeigt Tabelle mit den Mapping-Einstellungen.
- ▶ Im Variablenfeld von Bit 0 den Namen der Variable eingeben, auf die das sichere Prozesssignal abgebildet werden soll.
- > CODESYS erzeugt mit der eingegebenen Bezeichnung eine globale Variable vom Typ SAFEBOOL.
- > Safety-Anwendung kann über die Variable auf das sichere Prozesssignal zugreifen.



Die erzeugte Variable ist in der Globale-Variablen-Liste (GVL) IoConfigRemote_Globals abgelegt.

- ▶ Bei der Benennung der Variablen immer eindeutige Bezeichnungen wählen!

6.4.3 Testsignal an lokalem Ausgang erzeugen

20240

Bestimmte Gerätetypen benötigen als Eingangssignal einen Testimpuls, um einen Querschluss der beiden Eingangskanäle zu erkennen. Mithilfe des logischen Geräts SF_OUT_local_testpulse kann der Programmierer an einem lokalen Ausgang ein Testsignal erzeugen.

Um ein Testsignal an einem lokalen Ausgang zu erzeugen:

- ▶ Logisches Gerät "SF_OUT_loal_testpulse" zum Projektbaum hinzufügen
(→ **Logisches Gerät zum Projektbaum hinzufügen** (→ S. 34))

Die Parameter des Testimpulses sind fest eingestellt, es existieren keine zusätzlichen Konfigurationsoptionen (→ **SF_OUT_local_testpulse** (→ S. 274)).

6.4.4 Sicheres lokales Gerät aus Projekt entfernen

8963

Um ein an der lokalen E/A-Schnittstelle eingebundenes sicheres Gerät vollständig aus dem Projekt zu entfernen, müssen die 2 Instanzen des logischen Geräts im Projektbaum separat gelöscht werden.

1 Logisches Gerät im Standard-Bereich entfernen

- ▶ Logisches Gerät im Standardbereich des Gerätebaums markieren.
- ▶ [Bearbeiten] > [Löschen] wählen.
- > CODESYS entfernt das logische Gerät aus dem Standard-Bereich.

2 Logisches Gerät im Safety-Extension-Bereich entfernen

- ▶ Logisches Gerät im Safety-Extension-Bereich des Gerätebaums markieren.
- ▶ [Bearbeiten] > [Löschen] wählen.
- > CODESYS entfernt das logische Gerät aus dem Safety-Extension-Bereich.

6.5 Standard-Geräte an lokaler E/A-Schnittstelle konfigurieren

6976

Standard-Geräte werden ohne logisches Gerät in das CODESYS-Projekt eingebunden. Nachdem das Gerät ordnungsgemäß mit der lokalen E/A-Schnittstelle verdrahtet ist, kann der Programmierer über Funktionsbausteine direkt aus der sicheren Anwendung heraus auf die Prozessdaten zugreifen.



Der Zugriff auf nicht sichere Ein- und Ausgänge der lokalen E/A-Schnittstelle erfolgt ausschließlich über die fehlersichere SPS des AC4S.

- Um auf die Ein-/Ausgangsdaten der nicht sicheren Geräte an der lokalen E/A-Schnittstelle zuzugreifen:
→ **Lokale E/A-Schnittstelle: Auf nicht sichere Ein-/Ausgänge zugreifen** (→ S. [80](#))
- Um Prozessdaten zwischen Standard-SPS und fehlersicherer SPS des AC4S zu übertragen:
→ **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. [85](#))

6.6 IP-Einstellungen ändern

14524



► Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!

- Ethernet-Adapter-Konfigurator
→ Online-Hilfe > Feldbusunterstützung > Ethernet-Adapter-Konfigurator

Die Ethernet-Konfigurationsschnittstellen 1 (X3) und 2 (X8) dürfen nicht Teilnehmer des selben Ethernet-Subnetzes sein. Eine Nichtbeachtung kann zu Verbindungsproblemen zwischen dem Programmiersystem CODESYS und dem Gerät führen.

- IP-Einstellungen so konfigurieren, dass die Schnittstellen X3 und X8 zu unterschiedlichen Ethernet-Subnetzen gehören.

Um die IP-Einstellungen der Ethernet-Konfigurationsschnittstellen 1 (X3) und 2 (X8) zu ändern:

1 Ethernet-Schnittstelle wählen

- Im Gerätebaum: Doppelklick auf gewünschte Ethernet-Schnittstelle (X3 oder X8)
- > Editor-Fenster zeigt Ethernet-Geräteeditor.

2 IP-Einstellungen ändern

- IP-Einstellungen der Ethernet-Schnittstelle ändern.
- > Projekt speichern, um die Änderungen zu übernehmen.

6.7 Zusätzliches Gerät zum Projekt hinzufügen

18967

Innerhalb eines CODESYS-Projekts können mehrere AC4S verwaltet werden.

Um ein zusätzliches Gerät zu einem CODESYS-Projekt hinzuzufügen:

Voraussetzungen:

- > CODESYS-Projekt wurde korrekt erstellt (→ **Neues Projekt mit AC4S erstellen** (→ S. 17)).

1 Gerät hinzufügen

- ▶ Rechtsklick auf [MyProject]
- > Kontextmenü erscheint.
- ▶ [Gerät anhängen...] wählen.
- > Fenster [Gerät anhängen] erscheint:



- ▶ Folgende Werte einstellen:
 1. [Hersteller]: ifm electronic wählen.
 2. [Gerät]: In Liste [ifm SmartPLC SafeLine] wählen.
 3. [Name]: Eindeutigen Namen für Gerät eingeben
- ▶ Auf [Gerät anhängen] klicken, um das Gerät zum Projekt hinzuzufügen.
- ▶ Auf [Schließen] klicken, um das Fenster zu schließen.
- > CODESYS fügt Gerät zum Gerätebaum hinzu.

2 Gerät konfigurieren

- ▶ Gerät wie gewünscht konfigurieren (→ **Systemkonfiguration** (→ S. 23)).

3 Projekt speichern

- ▶ [Datei] > [Projekt speichern] wählen.
- > CODESYS speichert das Projekt.

6.8 Feldbus-Funktionalität erweitern

Inhalt	
Verfügbare Feldbus-Stacks	42
EtherCAT Master nutzen	43
Feldbus-Stack hinzufügen	46

15782

6.8.1 Verfügbare Feldbus-Stacks

13978

Neben seiner hardwareseitig definierten Feldbus-Funktionalität kann das Gerät softwareseitig als Feldbus-Gerät betrieben werden. Dafür muss der Anwender den Ethernet-Konfigurationsschnittstellen 1 (X3) oder 2 (X8) in CODESYS einen Feldbus-Stack zuweisen und konfigurieren. Das Gerät unterstützt folgende Feldbus-Stacks:

Bezeichnung	Feldbus	Weitere Infomrationen
EtherCAT Master	EtherCAT	→ EtherCAT Master nutzen (→ S. 43)
EtherNet/IP Scanner	EtherNet/IP	→ Feldbus-Stack hinzufügen (→ S. 46)
Modbus TCP Master	Modbus TCP	
Modbus TCP Slave Gerät	Modbus TCP	

6.8.2 EtherCAT Master nutzen

Inhalt	
EtherCAT Master hinzufügen	43
EtherCAT Master konfigurieren	44
EtherCAT Slave hinzufügen und konfigurieren	45

7474

Das Gerät kann zusätzlich als EtherCAT Master betrieben werden. Dafür stellt die ifm electronic einen EtherCAT-Master-Stack für das Programmiersystem CODESYS 3.5 bereit. Der Nutzer muss dem EtherCAT Master eine der beiden Ethernet-Konfigurationsschnittstellen (X3 oder X8) zuweisen.

EtherCAT Master hinzufügen

7475



Pro Gerät kann nur ein EtherCAT Master genutzt werden.

Um den EtherCAT-Master-Stack zum Projekt hinzuzufügen:

Voraussetzungen:

- ▶ Projekt mit AC4S öffnen oder erstellen.

1 EtherCAT-Master-Stack hinzufügen

- ▶ Im Gerätebaum: Rechtsklick auf [ifm_SmartPLC_SafeLine]
- ▶ Im Kontextmenü: [Gerät anhängen...] wählen.
- > Fenster [Gerät anhängen] erscheint.
- ▶ Folgende Werte einstellen:
 1. [Hersteller]: [ifm electronic] wählen.
 2. In Tabelle: [EtherCAT Master] wählen.
- ▶ Auf [Gerät anhängen] klicken, um das gewählten Gerät zum Projekt hinzuzufügen.
- ▶ Auf [Schließen] klicken, um das Fenster zu schließen.
- > CODESYS fügt EtherCAT Master zum Gerätebaum hinzu.



CODESYS fügt automatisch einen EtherCAT-Master-Aufruf zu dem Task mit der kürzesten Zykluszeit hinzu.

Für Hinweise zur Konfiguration der Taskabarbeitung: → **Task-Abarbeitung konfigurieren**
(→ S. [70](#))

EtherCAT Master konfigurieren

7477



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - EtherCAT Master
 - Online-Hilfe > Felddbusunterstützung > EtherCAT Konfigurator > EtherCAT-Master

Der Nutzer muss dem hinzugefügten EtherCAT Master die Ethernet-Konfigurationsschnittstelle 1 (X3) oder 2 (X8) zuweisen.

1 Voraussetzungen

- > CODESYS-Laptop/PC ist korrekt mit dem Gerät verbunden.
- > Netzwerkpfad der Standard-SPS ist eingestellt (→ **Programmierschnittstelle konfigurieren** (→ S. [21](#))).

2 Ethernet-Schnittstelle zuweisen

- ▶ Im Gerätenbaum: Doppelklick auf [EtherCAT_Master (EtherCAT Master)]
- ▶ Auf Registerkarte [Allgemein] klicken.
- > Editor-Fenster zeigt Konfigurationsoptionen des EtherCAT Master.
- ▶ Auf [Durchsuchen...] klicken.
- > Fenster [Netzwerkadapter auswählen] erscheint.
- ▶ Gewünschte Ethernet-Schnittstelle wählen (eth0 = X3 oder eth1 = X8).
- ▶ Auf [OK] klicken, um die gewählte Ethernet-Schnittstelle zu übernehmen.
- > Feld [Quelladresse (MAC)] zeigt die MAC-Adresse der gewählten Ethernet-Schnittstelle.
- > Feld [Netzwerkname] zeigt die Bezeichnung der Ethernet-Schnittstelle.
- > EtherCAT Master ist mit gewählter Ethernet-Schnittstelle gekoppelt.

3 EtherCAT Master konfigurieren

- ▶ Parameter des EtherCAT Master wie gewünscht einstellen.
- > EtherCAT Master ist konfiguriert.

Weiter mit: → **EtherCAT Slave hinzufügen und konfigurieren** (→ S. [45](#))

EtherCAT Slave hinzufügen und konfigurieren

7479



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - EtherCAT Slave
→ Online-Hilfe > Feldbusunterstützung > EtherCAT Konfigurator > EtherCAT-Slave

Um EtherCAT Slaves für den Betrieb mit dem EtherCAT Master des AC4S zu konfigurieren:

Voraussetzungen:

- > EtherCAT Master ist zum Projekt hinzugefügt und konfiguriert (→ **EtherCAT Master hinzufügen** (→ S. 43)).
- > Gerätebeschreibungsdatei des EtherCAT Slaves wurde heruntergeladen (→ Webseite des Herstellers).

1 Optional: EtherCAT Slave zum Geräte-Repository hinzufügen

- ▶ [Tools] > [Geräte-Repository...] wählen.
- > Fenster [Geräte-Repository] erscheint.
- ▶ [Installieren...] klicken.
- > Fenster [Installiere Gerätebeschreibung] erscheint.
- ▶ Gerätebeschreibungsdatei des EtherCAT Slaves wählen und [Öffnen] klicken.
- > CODESYS installiert das Gerät im Geräte-Repository.

2 EtherCAT Slave hinzufügen

- ▶ Im Standardbereich des Gerätebaums: Rechtsklick auf [EtherCAT_Master (EtherCAT Master)]
- ▶ Im Kontextmenü: [Gerät anhängen...] wählen.
- > Fenster [Gerät anhängen] erscheint.
- ▶ Im Bereich [Gerät] den gewünschten EtherCAT Slave wählen.
- ▶ Auf [Gerät anhängen] klicken, um das gewählte Gerät zum Projekt hinzuzufügen.
- ▶ Auf [Schließen] klicken, um das Fenster zu schließen.
- > CODESYS fügt den EtherCAT Slave zum Projekt hinzu.

3 EtherCAT Slave konfigurieren

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf den hinzugefügten EtherCAT Slave.
- > Editor-Fenster zeigt Konfigurationsoptionen des Geräts.
- ▶ EtherCAT Slave wie gewünscht konfigurieren.

6.8.3 Feldbus-Stack hinzufügen

17702



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Modbus-Konfigurator
→ Online-Hilfe > Feldbusunterstützung > Modbus-Konfigurator

Um die Ethernet-Schnittstelle als eine Feldbus-Schnittstelle zu deklarieren:

1 CODESYS-Projekt erstellen/laden

- ▶ CODESYS-Projekt mit AC4S erstellen oder laden.

2 Feldbus-Stack hinzufügen

- ▶ Im Gerätebaum: Rechtsklick auf [X3 (Ethernet)] oder [X8 (Ethernet)].
- ▶ Im Kontextmenü: [Gerät anhängen...] wählen.
- > Fenster [Gerät anhängen] erscheint.
- ▶ Folgende Werte einstellen:
 1. [Hersteller]: [<Alle Hersteller>] wählen.
 2. In Tabelle: In Spalte [Name] den gewünschten Feldbus-Stack wählen.
 3. [Name]: Name des Feldbus-Stacks eingeben.
- ▶ Auf [Gerät anhängen] klicken, um den gewählten Feldbus-Stack zum Projekt hinzuzufügen.
- > Im Gerätebaum: CODESYS fügt gewählten Feldbus-Stack als Unterelement der Ethernet-Schnittstelle ein.

3 Feldbus-Stack konfigurieren

- ▶ Hinzugefügtes Feldbus-Gerät konfigurieren.
- ▶ Projekt speichern, um die Änderungen zu übernehmen.

7 Programmierung

Inhalt	
Objekte einer Standard-SPS-Anwendung.....	47
Standard-SPS-Anwendung erstellen.....	48
Auf Standard-Ein- und Ausgangsdaten zugreifen.....	50
Funktionen des ifm-Packages nutzen	57
Visualisierungen nutzen	64
Querkommunikation	68
Task-Abarbeitung konfigurieren	70

7074

Dieses Kapitel liefert Informationen über die Programmierung der Standard-SPS des Geräts.

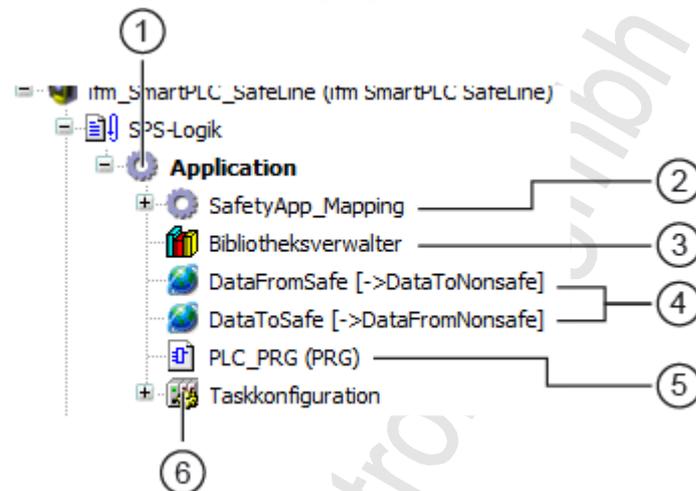


- Machen Sie sich mit der Programmierung nach Norm IEC 61131-3 vertraut!

7.1 Objekte einer Standard-SPS-Anwendung

18965

Alle Objekte einer Standard-SPS-Anwendung sind als Unterelemente des Knotens [Application] im Standardbereich des Gerätebaums gelistet. In der Grundkonfiguration enthält eine Standard-SPS-Anwendung folgende Objekte:



- ① [Application] ist der Container der Standard-SPS-Anwendung
- ② [SafetyApp_Mapping] ist eine Hilfsanwendung für den Datenaustausch zwischen Standard-SPS und fehlersicherer SPS (→ **Intervall des Datenaustauschs einstellen** (→ S. [88](#)))
- ③ [Bibliotheksverwalter] bietet Zugriff auf die Standard- und gerätespezifischen Funktionsbibliotheken
- ④ [DataFromSafe] und [DataToSafe] sind globale Variablenlisten für den Datenaustausch zwischen Standard-SPS und fehlersicherer SPS (→ **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. [85](#)))
- ⑤ [PLC_PRG (PRG)] bietet Zugriff auf den Programm-Editor der Standard-Anwendung (→ **Standard-SPS-Anwendung erstellen** (→ S. [48](#)))
- ⑥ [Taskkonfiguration] bietet Zugriff auf die Einstellungen der Task-Abarbeitung (→ **Haupt-Task konfigurieren** (→ S. [70](#)))

7.2 Standard-SPS-Anwendung erstellen

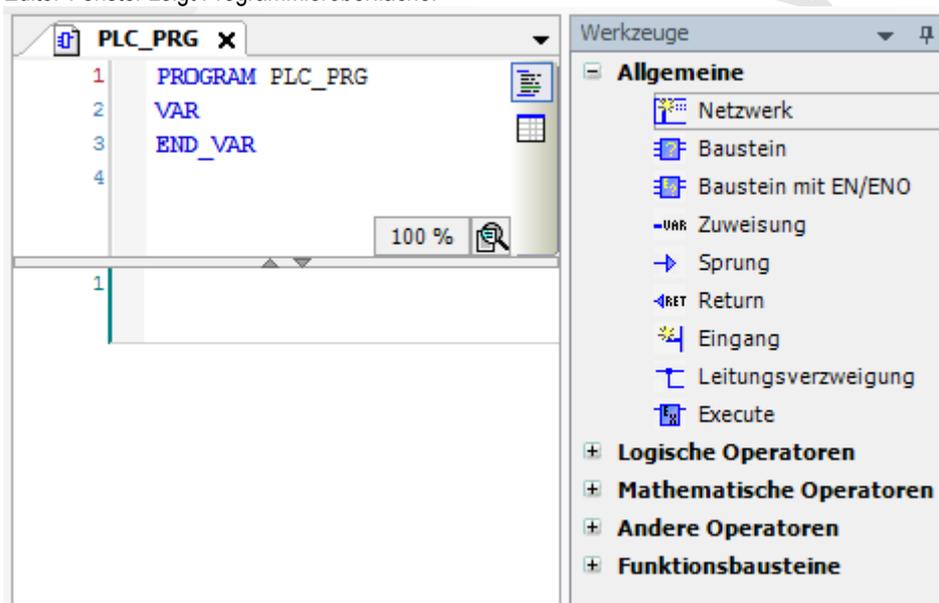
17691



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Applikation programmieren
→ Online-Hilfe > CODESYS Development System > Applikation programmieren
 - Programmierreferenz
→ Online-Hilfe > CODESYS Development System > Referenz Programmierung

Um eine Standard-SPS-Anwendung zu erstellen:

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf [PLC_PRG (PRG)]
- > Editor-Fenster zeigt Programmieroberfläche:



- ▶ Im Editor die Anwendung programmieren.

7.2.1 Remanente Variablen nutzen

18522

Die Standard-SPS des Geräts unterstützt die Verwendung remanenter Variablen. Als VAR RETAIN deklarierte Variablen werden in einem Speicherbereich abgelegt, der auch beim Ausschalten des Geräts erhalten bleibt.

Die Deklaration einer Variable als RETAIN beeinflusst auch deren Verhalten beim Zurücksetzen der Standard-SPS-Anwendung (→ **Unterstützte Reset-Varianten** (→ S. [112](#))).



- Der Speicherbereich für RETAIN-Variablen umfasst 4072 Bytes.
 - ▶ Bei der Deklaration von RETAIN-Variablen die maximale Größe des RETAIN-Speicherbereichs beachten!

7.2.2 Unterstützte Programmiersprachen

18034

Folgende Tabelle zeigt, welche Programmiersprachen nach IEC 61131 durch die ifm-Funktionsbibliotheken unterstützt werden:

Bibliothek	Funktionsbausteinsprache (FUP)	Ablaufsprache (AS)	Anweisungsliste (AWL)	Continuous Function Chart (CFC)	Kontaktplan (KOP)	Strukturierter Text (ST)
ACnnnn_Utils.library	X	X	X	X	X	X
ACnnnn_SYS_CMD.library	X	X	X	X	X	X

Legende:

X ... wird unterstützt

– ... wird nicht unterstützt

7.2.3 Systemzeit des Geräts ändern

18271

WARNUNG

Gefahr von unerwünschtem Systemverhalten!

Die Verwendung der CODESYS-Funktion SysTimeRtcSet zum Einstellen der Uhrzeit kann zu Fehlfunktionen führen.

- ▶ Um die Systemzeit (Datum, Uhrzeit) des Geräts einzustellen, ausschließlich die folgenden gerätespezifischen Kommandos nutzen:
 - Funktionsbaustein Set_DateTime (→ **Set_TimeDate** (→ S. [171](#)))
 - Systemkommando 0x1109 mit Funktionsbaustein ACnnnn_SysCmd (→ **ACnnnn_SysCmd** (→ S. [183](#)))

7.3 Auf Standard-Ein- und Ausgangsdaten zugreifen

Inhalt

Möglichkeiten des Zugriffs auf Ein- und Ausgangsdaten	50
Gültigkeit der Schnittstellen-Daten	51
Prozessdaten der AS-i Slaves	52
Feldbus-Daten	54
Prozessdaten der nicht-sicheren lokalen Geräte	56

11185



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Adressen nach IEC-Standard 61131-3:
 - Online-Hilfe > CODESYS Development System > Programmierreferenz > Operanden > Adressen
 - Zugriff auf IEC-Adresse über AT-Deklaration:
 - Online-Hilfe > CODESYS Development System > Programmierreferenz > Deklaration > AT-Deklaration
 - Definieren eines ALIAS für eine IEC-Adresse:
 - Online-Hilfe > CODESYS Development System > Programmierreferenz > Datentypen > Reference
 - Kopplung einer Programm-Variablen an eine Adresse (Mapping):
 - Online-Hilfe > CODESYS Development System > E/A-Anbindung konfigurieren

7.3.1 Möglichkeiten des Zugriffs auf Ein- und Ausgangsdaten

17621

In einem CODESYS-Projekt besitzt jeder Ein- und Ausgang eine physikalische Adresse nach IEC-Standard (z.B. %IW5). CODESYS bietet folgende Möglichkeiten, um aus einer Standard-SPS-Anwendung heraus auf diese Adresse und somit auf die Ein-/Ausgangsdaten eines Geräts zuzugreifen:

- Direkter Zugriff auf die IEC-Adresse
- Zugriff auf IEC-Adresse über AT-Deklaration
- Definition eines ALIAS für eine IEC-Adresse
- Verknüpfung einer Programm-Variablen mit einer IEC-Adresse (Mapping)



- Die IEC-Adressen (%I, %Q) der Ein- und Ausgänge verändern sich, wenn zusätzliche sichere AS-i Slaves in das CODESYS-Projekt eingebunden werden.
- ▶ Für den Zugriff auf die Ein und Ausgänge immer symbolische Programmierung nutzen!

7.3.2 Gültigkeit der Schnittstellen-Daten

413

Um den Zugriff auf Ein- und Ausgänge von AS-i Slaves zu erleichtern, stellen AC4S-Projekte fest definierte Schnittstellen (Interfaces) im Gerätebaum bereit (→ **Übersicht: Projektstruktur mit AC4S** (→ S. 19)).

In Abhängigkeit von der eingestellten Ausgangskontrolle (Manuell, Gateway, SPS) aktualisiert der CODESYS-Datenmapper nur bestimmte Adressbereiche der Schnittstellen. Folgende Tabelle zeigt, an welchen Adressbereichen der E/A-Schnittstellen in den einzelnen Betriebsarten aktuelle Werte anliegen:

Ausgangskontrolle	Schnittstellen	Aktualisierte Adressbereiche/Kanäle
Manuell	[ASi_Master_1]	AS-i 1 Input (%IB, %IW)
	[ASi_Master_2]*	AS-i 2 Input (%IB, %IW)
	[Fieldbus_Interface]	<ul style="list-style-type: none"> ▪ AS-i 1 Output (%IB, %IW) ▪ AS-i 2 Output (%IB, %IW)
Gateway	[ASi_Master_1]	AS-i 1 Input (%IB, %IW)
	[ASi_Master_2]*	AS-i 2 Input (%IB, %IW)
	[Fieldbus_Interface]	<ul style="list-style-type: none"> ▪ AS-i 1 Output (%IB, %IW) ▪ AS-i 2 Output (%IB, %IW)
SPS	[ASi_Master_1]	<ul style="list-style-type: none"> ▪ AS-i 1 Input (%IB, %IW) ▪ AS-i 1 Output (%QB, %QW)
	[ASi_Master_2]*	<ul style="list-style-type: none"> ▪ AS-i 2 Input (%IB, %IW) ▪ AS-i 2 Output (%QB, %QW)
	[Fieldbus_Interface]	<ul style="list-style-type: none"> ▪ AS-i 1 Output (%IB, %IW) ▪ AS-i 2 Output (%IB, %IW)

* ... nur verfügbar bei Geräten mit 2 AS-i Mastern

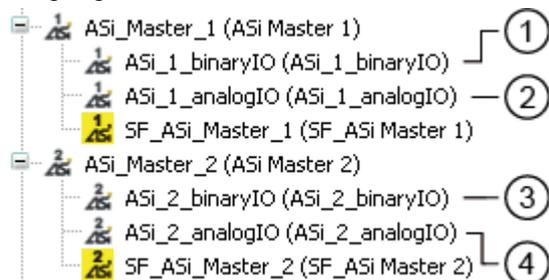


- ▶ Bei der Verknüpfung von Variablen mit Ein- und Ausgängen nur die Schnittstellen im Gerätebaum nutzen, die vom CODESYS-Datenmapper aktualisiert werden!

7.3.3 Prozessdaten der AS-i Slaves

17584

Der Gerätebaum bietet direkten Zugriff auf die zyklisch aktualisierten Prozessabbilder der Ein- und Ausgänge der AS-i Slaves.



- ① Digitale Ein- und Ausgangsdaten der Slave an AS-i Master 1: → **Digitale Ein- und Ausgangsdaten** (→ S. [53](#))
- ② Analoge Ein- und Ausgangsdaten der Slaves an AS-i Master 1: → **Analoge Ein- und Ausgangsdaten** (→ S. [53](#))
- ③ Digitale Ein- und Ausgangsdaten der Slaves an AS-i Master 2: → **Digitale Ein- und Ausgangsdaten** (→ S. [53](#))
- ④ Analoge Ein- und Ausgangsdaten der Slaves an AS-i Master 2: → **Analoge Ein- und Ausgangsdaten** (→ S. [53](#))



Gültigkeit der Schnittstellen-Daten beachten (→ **Gültigkeit der Schnittstellen-Daten** (→ S. [51](#)))!



Die Funktionsbibliothek `ACnnnn_Utils.library` enthält die komplexe Variable `ASi_NET`. Die Variable bildet alle Ein- und Ausgänge eines komplett ausgebauten AS-i Netzwerks ab. Der Programmierer kann diese Datenstruktur nutzen, um die Prozessabbilder der Ein- und Ausgänge eines AS-i Netzwerks zu speichern. (→ **ASi_NET (STRUCT)** (→ S. [178](#)))

Digitale Ein- und Ausgangsdaten

17625

Um auf die digitalen Prozessdaten der Slaves an AS-i Master 1 zuzugreifen:

- ▶ Im Gerätebaum: Doppelklick auf [ASi_1_binaryIO]
- > Editor-Fenster zeigt strukturierte Liste der digitalen Ein- und Ausgänge der AS-i Slaves.

Variable	Mapping	Kanal	Adresse	Typ
		ASi Input bin	%IB1	
		Digital S(A) Slave	%IB1	ARRAY [1..31] OF BYTE
		Digital B Slave	%IB32	ARRAY [1..31] OF BYTE
		ASi Output bin	%QB1	
		Digital S(A) Slave	%QB1	ARRAY [1..31] OF BYTE
		Digital B Slave	%QB32	ARRAY [1..31] OF BYTE

- ▶ In Spalte [Variable]: Auf klicken, um einzelne Variablen sichtbar zu machen.



Um in einem System mit 2 AS-i Mastern auf die digitalen Prozessdaten der Slaves an AS-i Master 2 zuzugreifen:

- ▶ Doppelklick auf [ASi_2_binaryIO]

Analoge Ein- und Ausgangsdaten

17626

Um auf die analogen Prozessdaten der Slaves an AS-i Master 1 zuzugreifen:

- ▶ Im Gerätebaum: Doppelklick auf [ASi_1_analogIO]
- > Editor-Fenster zeigt strukturierte Liste der analogen Ein- und Ausgänge der AS-i Slaves.

Variable	Mapping	Kanal	Adresse	Typ
		ASi Input	%IW32	ARRAY [1..31] OF SLAVEaANaINaTYPE
		ASi Output	%QW32	ARRAY [1..31] OF SLAVEaANaOUTaTYPE

- ▶ In Spalte [Variable]: Auf klicken, um einzelne Variablen sichtbar zu machen.



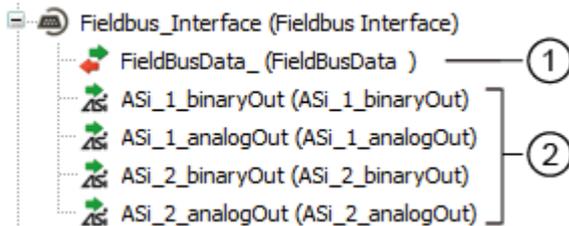
Um in einem System mit 2 AS-i Mastern auf die analogen Prozessdaten der Slaves an AS-i Master 2 zuzugreifen:

- ▶ Doppelklick auf [ASi_1_analogIO]

7.3.4 Feldbus-Daten

17585

Der Gerätebaum bietet direkten Zugriff auf die Daten, die zwischen Feldbus und Gerät übertragen werden.



- ① Daten, die das Gerät über den Feldbus sendet und empfängt.
→ **Ein- und Ausgangsdaten der Feldbus-Schnittstelle** (→ S. 54)
- ② Ausgangsdaten der AS-i Slaves an AS-i Master 1 und AS-i Master 2, die von der übergeordneten Feldbus-SPS gesendet werden.
→ **Ausgangsdaten der AS-i Slaves** (→ S. 55)

! Gültigkeit der Schnittstellen-Daten beachten (→ **Gültigkeit der Schnittstellen-Daten** (→ S. 51))!

Ein- und Ausgangsdaten der Feldbus-Schnittstelle

17619

Die Ein- und Ausgangsdaten der Feldbus-Schnittstelle werden zyklisch in jeweils 120 Worten übertragen. Der Programmierer kann symbolisch (empfohlen) oder über die IEC-Adresse auf diese Daten zugreifen.

Um auf die Ein- und Ausgangsdaten der Feldbus-Schnittstelle zuzugreifen:

- ▶ Im Gerätebaum: Doppelklick auf [FieldBusData_]
- > Editor-Fenster zeigt strukturierte Liste der Ein- und Ausgänge:

Variable	Mapping	Kanal	Adresse	Typ
		Inputs from fieldbus	%IW373	ARRAY [0..119] OF WORD
		Outputs to fieldbus	%QW373	ARRAY [0..119] OF WORD

- ▶ In Spalte [Variable]: Auf klicken, um einzelne Variablen sichtbar zu machen.

Ausgangsdaten der AS-i Slaves

17620

Der Bereich enthält alle Daten, die der übergeordnete Feldbus-Controller zyklisch über den Feldbus an die AS-i Ausgangs-Slaves sendet. Die Daten sind wie ein AS-i Netzwerk strukturiert. Der Programmierer kann symbolisch (empfohlen) oder über die IEC-Adressen auf diese Daten zugreifen.



Die Daten dieses Bereichs kann der Programmierer nutzen, um bei eingestellter Ausgangskontrolle = SPS die Sollwerte der übergeordneten Feldbus-Steuerung in der CODESYS-Standard-SPS zu verarbeiten.

Digitale Ausgangsdaten

17630

Um auf die digitalen Ausgangsdaten der Slaves an AS-i Master 1 zuzugreifen:

- ▶ Im Gerätebaum: Doppelklick auf [ASi_1_binaryIO]
- > Editor-Fenster zeigt die strukturierte Liste der digitalen Ausgangsdaten:

Variable	Mapping	Kanal	Adresse	Typ
		FB ASi Output bin	%IB986	
		Digital S(A) Slave	%IB986	ARRAY [1..31] OF BYTE
		Digital B Slave	%IB1017	ARRAY [1..31] OF BYTE

- ▶ In Spalte [Variable]: Auf klicken, um einzelne Variablen sichtbar zu machen.



Um in einem System mit 2 AS-i Mastern auf die digitalen Ausgangsdaten der Slaves an AS-i Master 2 zuzugreifen:

- ▶ Doppelklick auf [ASi_2_binaryIO]

Analoge Ausgangsdaten

17631

Um auf die analogen Ausgangsdaten der Slaves an AS-i Master 1 zuzugreifen:

- ▶ Im Gerätebaum: Doppelklick auf [ASi_1_analogOut]
- > Editor-Fenster zeigt die strukturierte Liste der analogen Ausgangsdaten:

Variable	Mapping	Kanal	Adresse	Typ
		FB ASi Output ana	%IW524	ARRAY [1..31] OF SLAVEaANaaiNaTYPE

- ▶ In Spalte [Variable]: Auf klicken, um einzelne Variablen sichtbar zu machen.



Um in einem System mit 2 AS-i Mastern auf die analogen Ausgangsdaten der Slaves an AS-i Master 2 zuzugreifen:

- ▶ Doppelklick auf [ASi_2_analogOut]

7.3.5 Prozessdaten der nicht-sicheren lokalen Geräte

8999

Der Zugriff auf nicht sichere Ein- und Ausgänge der lokalen E/A-Schnittstelle erfolgt ausschließlich über die fehlersichere SPS des AC4S.

- Um auf die Ein-/Ausgangsdaten der nicht sicheren Geräte an der lokalen E/A-Schnittstelle zuzugreifen:
→ **Lokale E/A-Schnittstelle: Auf nicht sichere Ein-/Ausgänge zugreifen** (→ S. [80](#))
- Um Prozessdaten zwischen Standard-SPS und fehlersicherer SPS des AC4S zu übertragen:
→ **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. [85](#))



Die Prozessdaten der nicht-sicheren Ein- und Ausgänge der lokalen E/A-Schnittstelle werden am Knoten [local_IO_] nicht automatisch aktualisiert!

7.4 Funktionen des ifm-Packages nutzen

18489

Das CODESYS-Package "CODESYS für ifm SmartPLC SafeLine" stellt verschiedene Funktionen für die Programmierung der geräteinternen CODESYS-Steuerung bereit. In den folgenden Abschnitten werden diese Funktionen kurz beschrieben. Um eine leichtere Orientierung zu ermöglichen, sind die Funktionen thematisch gruppiert und mit einem Querverweis zur ausführlichen Erklärung im Anhang des Handbuchs versehen.

7.4.1 Steuerinterface der ifm-Funktionsbausteine

17660

Alle Funktionsbausteine (FB) der Bibliotheken ACnnnn_Utils.library und ACnnnn_SYS_CMD.library besitzen Ein- und Ausgänge für Steuersignale. Die Eingänge aktivieren die Ausführung des Funktionsbausteins. Die Ausgänge liefern Informationen über den internen Zustand des Funktionsbausteins. Mithilfe der Signale kann der Programmierer eine Steuerungsstruktur für die gezielte Abarbeitung des FB erzeugen und auf eventuell auftretende Fehler reagieren.

Anzahl und Bezeichnung der FB-Steuersignale geben Hinweise auf die Art der FB-Ausführung:

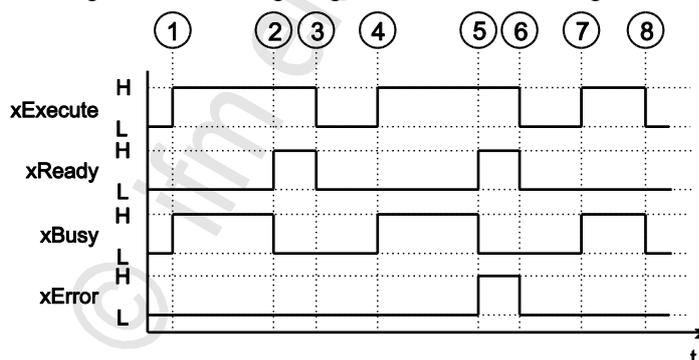
FB mit einmaliger Ausführung

17140

Diese Funktionsbausteine führen nach der Aktivierung ihre Funktion genau einmal aus. Um die Funktion erneut auszuführen, muss der FB wieder aktiviert werden. FBs dieser Art verfügen über ein Steuer-Interface mit folgenden Ein- und Ausgängen:

Bezeichnung	Art	Datentyp	Bedeutung	Mögliche Werte	
xExecute	Eingang	BOOL	FB-Ausführung steuern	FALSE ⇒ TRUE	FB-Ausführung starten
				sonst.	keine Auswirkung
xReady	Ausgang	BOOL	Anzeige, ob FB-Ausführung beendet ist	FALSE	FB-Ausführung noch nicht beendet
				TRUE	FB-Ausführung beendet
xBusy	Ausgang	BOOL	Anzeige, ob FB aktiv ist	FALSE	FB ist inaktiv.
				TRUE	FB ist aktiv.
xError	Ausgang	BOOL	Anzeige, ob bei FB-Ausführung Fehler aufgetreten sind	FALSE	FB fehlerfrei ausgeführt.
				TRUE	Fehler bei FB-Ausführung aufgetreten.
wDiagnostic	Ausgang	WORD	Fehler-Code	FB-spezifisch	

Die folgende Abbildung zeigt den Zusammenhang zwischen den Belegungen der Steuersignale:



- ① xExecute = TRUE: Steigende Flanke (FALSE → TRUE) startet die Ausführung des FB.
xBusy = TRUE: FB-Ausführung ist gestartet, aber noch nicht beendet.
- ② xReady = TRUE: FB-Ausführung ist beendet; an Datenausgängen liegen gültige Werte an.
xBusy = FALSE: FB ist nicht mehr aktiv.
xError = FALSE: FB-Ausführung verlief fehlerfrei.
- ③ xExecute = FALSE: Alle Signalausgänge werden auf FALSE gesetzt und alle internen Zustände werden zurückgesetzt.
- ④ → ①
- ⑤ xReady = TRUE: FB-Ausführung ist beendet.
xBusy = FALSE: FB ist nicht mehr aktiv.
xError = TRUE: Bei FB-Ausführung sind Fehler aufgetreten; wDiagnostic gibt Fehler-Code aus.
- ⑥ → ③
- ⑦ → ①
- ⑧ xExecute = FALSE: FB-Ausführung wird vor Beendigung unterbrochen; Alle Signalausgänge werden auf FALSE gesetzt und alle internen Zustände werden zurückgesetzt.

FB mit zyklischer Ausführung

17141

Funktionsbausteine, die bei Aktivierung ihre Funktion solange zyklisch ausführen, bis sie wieder deaktiviert werden, besitzen folgende Steuer-Ein- und Ausgänge:

Bezeichnung	Art	Datentyp	Bedeutung	Mögliche Werte	
xEnable	Eingang	BOOL	FB-Ausführung steuern	FALSE	FB-Ausführung stoppen
				TRUE	FB-Ausführung starten
xActive	Ausgang	BOOL	Anzeige, ob FB-Ausführung beendet ist	FALSE	FB-Ausführung noch nicht beendet
				TRUE	FB-Ausführung beendet
xError	Ausgang	BOOL	Anzeige, ob bei FB-Ausführung Fehler aufgetreten sind	FALSE	FB fehlerfrei ausgeführt.
				TRUE	Fehler bei FB-Ausführung aufgetreten.
wCycleCount	Ausgang	WORD	Zähler für die FB-Zyklen	Ganzzahliger Wert (Hexadezimaldarstellung)	
wDiagnostic	Ausgang	WORD	Fehler-Code	FB-spezifisch	

7.4.2 System konfigurieren

17450

Um das System des Gerätes zu konfigurieren, die folgenden Funktionsbausteine nutzen:

Name	Beschreibung	Referenz
QuickSetupASi_Master	Quick-Setup-Routine auf einem AS-i Master ausführen	→ QuickSetupASi_Master (→ S. 169)
Set_TimeDate	Systemzeit (Datum, Uhrzeit) des Systems einstellen	→ Set_TimeDate (→ S. 171)
Get_FieldbusInfo	Feldbustyp, den Status der Feldbus-Verbindung und die Parameter der Feldbus-Schnittstelle lesen	→ Get_FieldbusInfo (→ S. 167)

7.4.3 AS-i Master konfigurieren

17448

Um den oder die AS-i Master des Geräts zu konfigurieren, die folgenden Funktionsbausteine nutzen:

Name	Beschreibung	Referenz
Set_Mode	Betriebsart des AS-i Masters einstellen (Projektierungsmodus oder Geschützter Betrieb)	→ Set_Mode (→ S. 152)
Set_ASi_Config	Diagnosefunktionen des AS-i Masters (Doppeladresserkennung, Erdschlussfehlererkennung) einstellen	→ Set_ASi_Config (→ S. 148)
Set_AdressMode	Automatische Adressierung des AS-i Masters einstellen	→ Set_AddressMode (→ S. 146)

7.4.4 AS-i Slaves konfigurieren

17449

Um die AS-i Slaves zu konfigurieren, die mit dem Gerät verbunden sind, die folgenden Funktionsbausteine nutzen:

Name	Beschreibung	Referenz
Set_SlaveAddress	Adresse eines AS-i Slaves ändern	→ Set_SlaveAddress (→ S. 158)
Set_SlaveParameter	E/A-Konfiguration und ID-Codes (IO, ID, ID1, ID2) eines AS-i Slaves ändern	→ Set_SlaveParameter (→ S. 162)
Set_SlaveExtendedID1	Extended ID1 eines AS-i Slaves ändern	→ Set_SlaveExtendedID1 (→ S. 160)

7.4.5 AS-i Netzwerk verwalten

17126

Um die vom AC4S gesteuerten AS-i Netzwerke zu verwalten, die folgenden Funktionsbausteine nutzen:

Komplexe Variablen nutzen

18528

Die Bibliothek stellt dem Programmierer verschiedene komplexe Variablen (STRUCT) zur Verfügung. Sie bündeln logisch zusammengehörige Daten. Sie erleichtern dadurch die Organisation der Datenhaltung in der Anwendung und verringern gleichzeitig die Fehleranfälligkeit bei der Variablendeklaration.

Folgende komplexe Variablen stehen zur Verfügung:

Name	Bedeutung	Referenz
ASI_NET	Die komplexe Variable enthält das komplette Prozessabbild (Ein- und Ausgänge) eines AS-i Netzwerks.	→ ASI_NET (STRUCT) (→ S. 178)
ASI_DATA	Die komplexe Variable enthält folgende Komponenten: <ul style="list-style-type: none"> ▪ Slave-Listen (LPS, LDS, LAS, LPF, LCE, LCEMS, LCEAS, LDAE) ▪ Parameter-Abbilder (PI, PP) ▪ Konfigurationsdaten der AS-i Slaves (CDI, PCD) 	→ ASI_DATA (STRUCT) (→ S. 175) → Get_Asi_Data (→ S. 164)

Netzwerkeinstellungen ändern

17568

Name	Beschreibung	Referenz
Set_ProjectAll	Projektierungsabgleich auf einem AS-i Master ausführen	→ Set_ProjectAll (→ S. 156)
Set_LPS	Liste der projektierten Slaves (LDS) ändern	→ Set_LPS (→ S. 150)
Set_PCD	Permanente Projektierungsdaten (IO, ID, ID1, ID2) aller Slaves am AS-i Master ändern	→ Set_PCD (→ S. 154)

Netzwerkeinstellungen lesen

18532

Um die Netzwerkeinstellungen zyklisch auszulesen und in der Applikation bereitzustellen:

Name	Beschreibung	Referenz
Get_ASi_Data	<p>Folgende Datensätze für Netzwerkverwaltung gebündelt und zyklisch auslesen:</p> <ul style="list-style-type: none"> ▪ Liste der aktivierten Slaves (LAS) ▪ Liste der detektierten Slaves (LDS) ▪ Liste der projektierten Slaves (LPS) ▪ Liste der Peripheriefehler (LPF) ▪ Liste der Konfigurationsfehler (LCE) ▪ Liste der Konfigurationsfehler - fehlende Slaves (LCEMS) ▪ Liste der Konfigurationsfehler - zusätzliche Slaves (LCEAS) ▪ Liste der Doppeladressfehler (LDAE) ▪ Konfigurationsdaten-Abbild (CDI) ▪ Permanente Konfigurationsdaten (PCD) ▪ Eingangsparameter (PI) ▪ Ausgangsparameter (PP) 	→ Get_ASi_Data (→ S. 164)

Alternativ können diese Daten auch separat mit den folgenden FB ausgelesen werden:

Parameter-Abbilder lesen

17569

Name	Beschreibung	Referenz
Get_InputParameter	Parameter der Eingänge der Slaves am AS-i Master lesen (PI)	→ Get_InputParameter (→ S. 142)
Get_OutputParameter	Parameter der Ausgänge der Slaves am AS-i Master lesen (PP)	→ Get_OutputParameter (→ S. 144)

Slave-Listen lesen

18530

Name	Beschreibung	Referenz
Get_LPS	Liste der projektierten Slaves (LPS) lesen	→ Get_LPS (→ S. 126)
Get_LDS	Liste der detektierten Slaves (LDS) lesen	→ Get_LDS (→ S. 124)
Get_LAS	Liste der aktivierten Slaves (LAS) lesen	→ Get_LAS (→ S. 122)
Get_LPF	Liste der Peripheriefehler (LPF) lesen	→ Get_LPF (→ S. 136)
Get_LCE	Liste der Konfigurationsfehler (LCE) lesen	→ Get_LCE (→ S. 128)
Get_LCEMS	Liste der Konfigurationsfehler - Fehlender Slave (LCEMS) lesen	→ Get_LCEMS (→ S. 132)
Get_LCEAS	Liste der Konfigurationsfehler - Zusätzlicher Slave (LCEAS) lesen	→ Get_LCEAS (→ S. 130)
Get_LDAE	Liste der Doppeladressfehler (LDAE) lesen	→ Get_LDAE (→ S. 134)

Konfigurationsdaten der Slaves lesen

18533

Name	Beschreibung	Referenz
Get_CDI	Konfigurationsdaten-Abbild (IO, ID, ID1, ID2) aller Slaves am AS-i Master lesen	→ Get_CDI (→ S. 138)
Get_PCD	Permanente Konfigurationsdaten aller Slaves (IO, ID, ID1, ID2) am AS-i Master lesen	→ Get_PCD (→ S. 140)

Status der Spannungsversorgung lesen

18529

Name	Beschreibung	Referenz
Get_ASi_PHY_Dat	Status der Spannungsversorgung des AS-i Netzwerks ermitteln	→ Get_ASi_PHY_Dat (→ S. 119)

7.4.6 Kommandos an System und AS-i Master senden

17659

Ähnlich der azyklisch übertragenen Kommando-Kanäle und Datensätze des Geräts, kann der Programmierer mit dem FB ACnnnn_SysCmd (→ **ACnnnn_SysCmd** (→ S. 183)) Kommandos an das System oder einen AS-i Master senden.

- Übersicht Systemkommandos: → **Tabelle: Systemkommandos** (→ S. 184)
- Übersicht AS-i Master-Kommandos: → **Tabelle: AS-i Master-Kommandos** (→ S. 184)



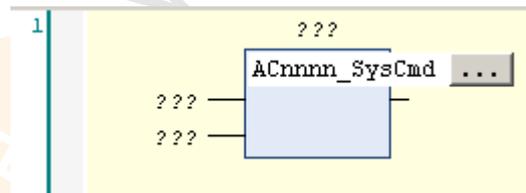
Der FB ACnnnn_SysCmd ist standardmäßig verborgen. Um den FB zu einem Programmbaustein hinzuzufügen:

- ▶ Gewünschtes Netzwerk markieren und mit [FUP/KOP/AWL] > [Leeren Baustein einfügen] einen leeren Funktionsbaustein einfügen.

> Netzwerk zeigt leeren FB.

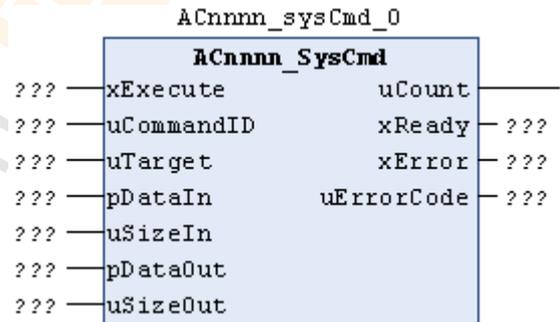
- ▶ Doppelklick auf Namensfeld des FB

- ▶ Bezeichnung ACnnnn_SysCmd eingeben und mit [ENTER] bestätigen.



> FB hat Ein- und Ausgänge des ACnnnn_SysCmd.

- ▶ Ein- und Ausgänge des FB entsprechend des gewünschten Kommandos einstellen.



© ifm electronic gmbh

7.5 Visualisierungen nutzen

Inhalt	
Unterstützte Visualisierungstypen	64
Visualisierung zu einem Projekt hinzufügen.....	65
Visualisierung erstellen.....	66
Visualisierung konfigurieren	67

17059



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Visualisierungen in CODESYS
→ Online-Hilfe > CODESYS Visualization

7.5.1 Unterstützte Visualisierungstypen

11022

Das Gerät unterstützt folgende CODESYS-Visualisierungstypen:

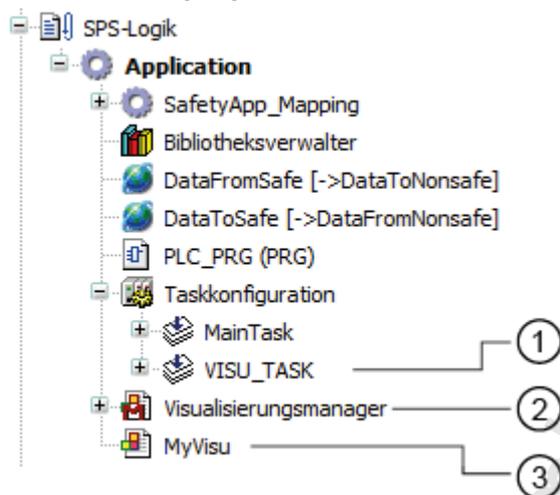
- **Web-Visualisierung (WebVisu)**
Eine WebVisu bietet die Möglichkeit, in einem Webbrowser mithilfe einer anwendungsspezifischen Visualisierung ausgewählte Prozess- und Steuerdaten grafisch darzustellen.

7.5.2 Visualisierung zu einem Projekt hinzufügen

10930

Um eine Visualisierung zu einem CODESYS-Projekt hinzuzufügen:

- ▶ CODESYS-Projekt öffnen.
ODER:
Neues CODESYS-Projekt erstellen. (→ **Neues Projekt mit AC4S erstellen** (→ S. 17))
- ▶ Im Standardbereich des Gerätebaums: [Application] wählen.
- ▶ [Projekt] > [Objekt hinzufügen] > [Visualisierung...] wählen.
- > Fenster [Visualisierung hinzufügen] erscheint.
- ▶ Im Feld [Name] eine Bezeichnung für die Visualisierung eingeben und mit [Hinzufügen] übernehmen.
- > CODESYS fügt folgende Elemente zum Standardbereich des Gerätebaums hinzu:



- ① [VISU_TASK]:
Eigenschaften des Visualisierungs-Tasks einstellen: → **Visualisierung-Task konfigurieren** (→ S. 70)
- ② [Visualisierungsmanager]:
Eigenschaften der Visualisierungen einstellen: → **Eigenschaften der Web-Visualisierung ändern** (→ S. 67)
- ③ [MyVisu]:
Bereich für die Erstellung der Visualisierungsobjekte: → **Visualisierung erstellen** (→ S. 66)

7.5.3 Visualisierung erstellen

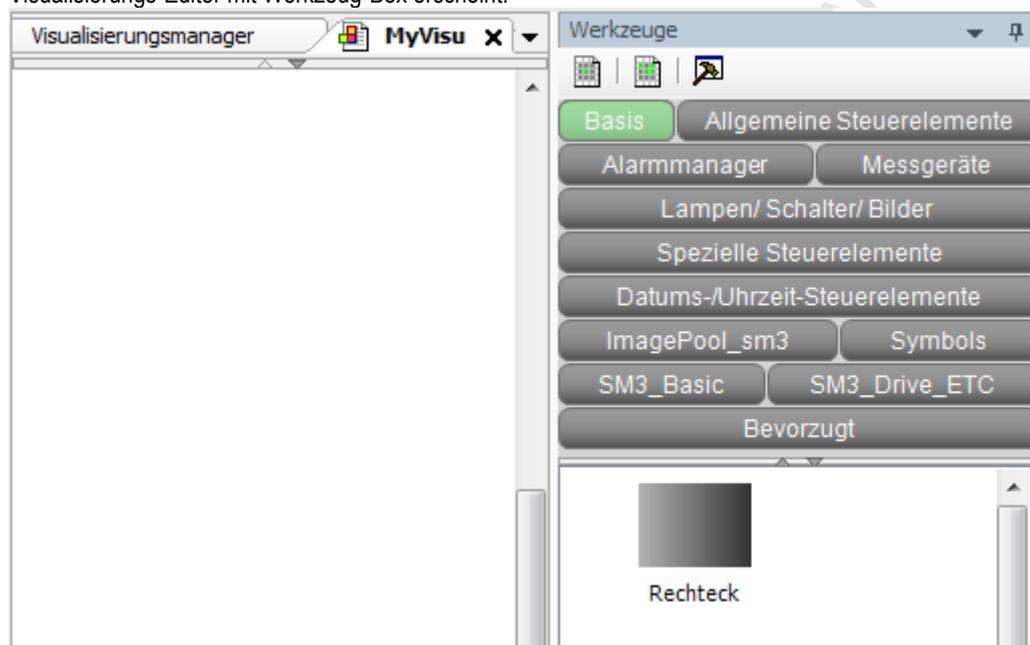
17061



Für Target- und Web-Visualisierung jeweils eine separate Visualisierung erstellen.

Um eine Visualisierung für eine Standard-SPS-Anwendung zu erstellen:

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf [MyVisu]
- > Visualisierungs-Editor mit Werkzeug-Box erscheint:



- ▶ Mithilfe der Werkzeuge die Visualisierung erstellen.
- ▶ Projekt speichern, um die Änderungen zu übernehmen.

7.5.4 Visualisierung konfigurieren

11027

Um die Eigenschaften der erstellten Visualisierung zu konfigurieren, eine der folgenden Optionen wählen:

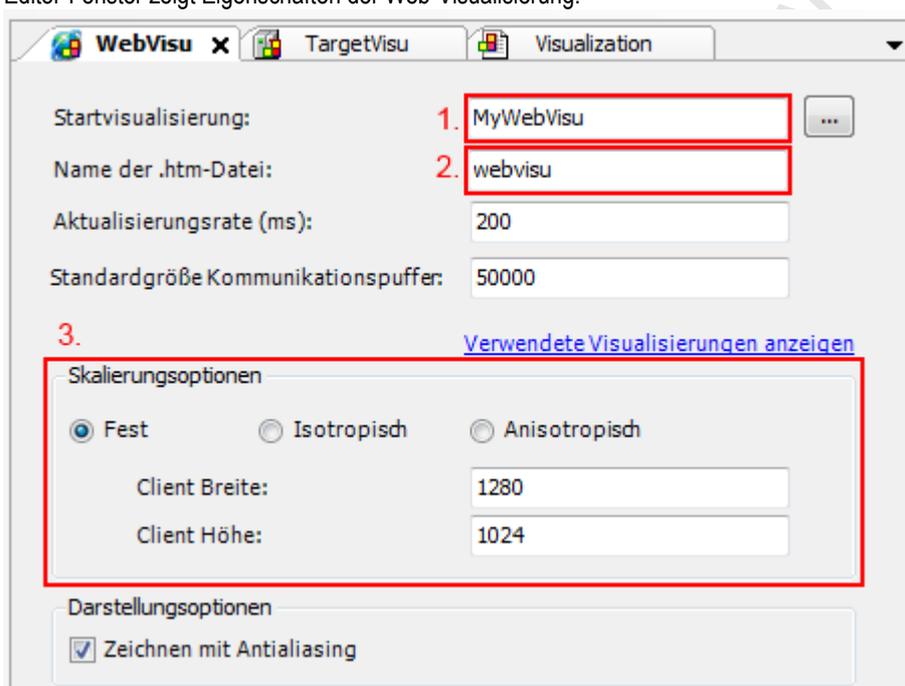
- **Eigenschaften der Web-Visualisierung ändern** (→ S. 67)

Eigenschaften der Web-Visualisierung ändern

17065

Um die Eigenschaften der Web-Visualisierung zu ändern:

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf [Web-Visualisierung]
- > Editor-Fenster zeigt Eigenschaften der Web-Visualisierung:



- ▶ Folgende Werte einstellen:
 1. Feld [Startvisualisierung]: Die erstellte Web-Visualisierung wählen.
 2. Feld [Name der .htm-Datei]: Bezeichnung für HTML-Datei eingeben (→ Hinweis).
 3. Bereich [Skalierungsoptionen]: Feste Breite und Höhe wie abgebildet eingeben.



Im Feld [Name der .htm-Datei] die Bezeichnung eingeben, unter der die Web-Visualisierung im Webbrowser erreichbar sein soll (→ **Web-Visualisierung anzeigen** (→ S. 115)).

- ▶ Bei der Eingabe der Bezeichnung nur Kleinschreibung nutzen!
- ▶ Projekt speichern, um geänderte Werte zu übernehmen.

7.6 Querkommunikation

Inhalt	
Netzwerkvariablen nutzen	68
	10293



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - → Online-Hilfe > CODESYS Development System > Daten im Netzwerk austauschen > Netzwerkvariablen

Das Gerät unterstützt die CODESYS-Funktion "Netzvariablen". Der Anwender kann damit Daten zwischen den Standard-SPSen eines Projekts übertragen.

Netzvariablen werden nach dem Broadcast-Prinzip übertragen. Ein Sender kann Daten an mehrere Empfänger verschicken.

7.6.1 Netzwerkvariablen nutzen

16189



- ▶ Hinweise zur Konfiguration der Ethernet-Konfigurationsschnittstellen 1 (X3) und 2 (X8) beachten: → **IP-Einstellungen ändern** (→ S. 40)

Die Netzwerkvariablen dürfen nur im Subnetz der Ethernet-Schnittstelle gesendet werden, über die das Gerät querkommuniziert. Der Programmierer muss die Broadcast-Adresse der Netzwerkvariablenliste entsprechend ändern.

Beispiel:

- Ethernet-Schnittstelle: X3
- IP-Adresse der Ethernet-Schnittstelle: 192.168.1.10
- Einzustellende Broadcast-Adresse: 192.168.1.255

Um die Querkommunikation zwischen Standard-SPSen zu erzeugen:

Voraussetzungen:

- > AC4S 1 und AC4S 2 sind Teilnehmer des selben CODESYS-Projekts (→ **Zusätzliches Gerät zum Projekt hinzufügen** (→ S. 41)).
- > Die AC4S-Geräte sind über Ethernet-Netzwerk gekoppelt.

1 Netzwerkvariablenliste des Senders erstellen

- ▶ Im Standardbereich des Gerätebaums des AC4S 1: [SPS-Logik] > [Application] wählen.
- ▶ [Projekt] > [Objekt hinzufügen] > [Netzwerkvariablenliste (Sender)...] wählen.
- > Dialogfenster erscheint.
- ▶ Folgende Parameter wie gewünscht einstellen
 1. [Name]: Eindeutige Bezeichnung der Variablenliste
 2. [Netzwerktyp]: UDP
 3. [Einstellungen]: Broadcast-Adresse (→ Hinweis)
 4. [Task]: Gewünschter Task
- ▶ Auf [Hinzufügen] klicken, um die eingegebenen Werte zu übernehmen.
- > CODESYS erzeugt Netzwerkvariablenliste des Senders.
- > Netzwerkvariablenliste erscheint im Gerätebaum als Unterknoten der Standard-Anwendung.

2 Netzwerkvariablen deklarieren

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf Netzwerkvariablenliste des Senders
- > Editor-Fenster zeigt Variablendeklaration.
- ▶ Netzwerkvariablen deklarieren.

3 Netzwerkvariablenliste des Empfängers erstellen

- ▶ Im Standardbereich des Gerätebaums des AC4S 2: [SPS-Logik] > [Application] wählen.
- ▶ [Projekt] > [Objekt hinzufügen] > [Netzwerkvariablenliste (Empfänger)...] wählen.
- > Dialogfenster erscheint.
- ▶ Folgende Parameter wie gewünscht einstellen
 1. [Name]: Eindeutige Bezeichnung der Variablenliste
 2. [Task]: Gewünschter Task
 3. [Sender]: Variablenliste des Senders
- ▶ Auf [Hinzufügen] klicken, um die eingegebenen Werte zu übernehmen.
- > CODESYS erzeugt Netzwerkvariablenliste des Empfängers.
- > Netzwerkvariablenliste erscheint im Gerätebaum als Unterknoten der Standard-Anwendung.

4 Optional: Weitere Empfängerlisten erstellen

- ▶ Schritt 3 wiederholen, um zusätzliche Geräte mit der Variablenliste des Senders zu verbinden.

7.7 Task-Abarbeitung konfigurieren

4109



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Taskkonfiguration
 - Online-Hilfe > CODESYS Development System > Applikation programmieren > Taskkonfiguration

Die Abarbeitung der Tasks wird durch Parameter gesteuert. Der Anwender kann jeden Task separat parametrieren.

CODESYS legt bei der Erstellung des Projekts und der Visualisierungen automatisch folgende Tasks an:

Name	Beschreibung	Verweis
[MainTask]	Konfiguration des Haupt-Tasks (u.a. für Hauptprogramm [PLC_PRG (PRG)])	→ Haupt-Task konfigurieren (→ S. 70)
[VISU_TASK]	Konfiguration des Tasks für die Abarbeitung der Visualisierung	→ Visualisierung-Task konfigurieren (→ S. 70)

7.7.1 Haupt-Task konfigurieren

18412

Die Grundeinstellungen der Task-Eigenschaften decken die Anforderungen vieler Anwendungen ab. Bei nicht optimaler Geräte-Performance muss der Anwender mithilfe von Tests die optimalen Task-Eigenschaften eigenverantwortlich ermitteln und einstellen.

Um die Parameter eines Tasks zu ändern:

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf [Taskkonfiguration] > [MainTask]
- > Editor-Fenster zeigt die Konfiguration des Haupt-Tasks.
- ▶ Parameter wie gewünscht einstellen.
- > Eingestellter Wert ist gültig.

7.7.2 Visualisierung-Task konfigurieren

17066



- ▶ Um die Abarbeitung anderer, für die Kernfunktionalität der Applikation wichtigen Tasks nicht zu unterbrechen, den Visualisierungs-Task (VISU_TASK) mit einer möglichst niedrigeren Priorität ausführen.
- ▶ Um die Ressourcen der geräteinternen SPS und des Feldbus-Netzwerks zu schonen, den VISU_TASK in angemessenen zyklischen Intervallen ausführen.

Jede Visualisierung wird getrennt vom Programm-Code in einem separaten Task ausgeführt. Um die Eigenschaften des Visualisierung-Tasks einzustellen:

- ▶ Im Standardbereich des Gerätebaums: Doppelklick auf [Taskkonfiguration] > [VISU_TASK]
- > Editor-Fenster zeigt Parameter des Visualisierungs-Tasks.
- ▶ Parameter wie gewünscht einstellen.
- ▶ Projekt speichern, um geänderte Werte zu übernehmen.

8 Programmierung der fehlersicheren SPS

Inhalt	
Objekte einer sicheren Anwendung	72
Sichere Anwendung erstellen.....	73
Safety-Funktionen des ifm AS-i Packages nutzen	78
Safety-Funktionen von CODESYS nutzen	82
Datenaustausch zwischen Standard-SPS und fehlersicherer SPS	85
Sichere Querkommunikation	89
Sichere Feldbuskommunikation	92
Prozesssicherheitszeit einstellen	94
Safety-Task konfigurieren.....	101

8689

Dieses Kapitel liefert Informationen über die Programmierung der fehlersicheren SPS des AC4S.

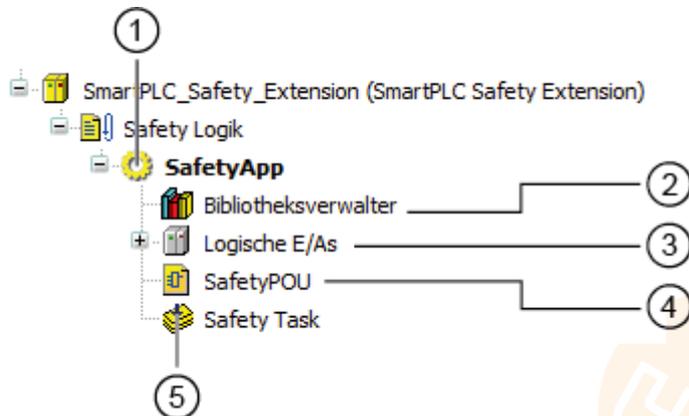


- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - sichere Anwendungen
→ Online-Hilfe > Add-Ons > CODESYS Safety > Gerätetopologie > Sicherheitsapplikation
 - Programmierreferenz
→ Online-Hilfe > Add-Ons > CODESYS Safety > Programmierreferenz
 - Konzepte
→ Online-Hilfe > Add-Ons > CODESYS Safety > Konzepte

8.1 Objekte einer sicheren Anwendung

7133

Alle Objekte einer sicheren Anwendung sind als Unterelemente des Knotens [SafetyApp] im Safety-Bereich des Gerätebaums gelistet. In der Grundkonfiguration enthält die sicherheitsgerichtete Anwendung folgende Objekte:



- ① [SafetyApp] ist der Container für Objekte einer sicheren Anwendung und bietet Zugriff auf den Editor der Safety Applikationsobjekte
→ **Sicheres Projekt pinnen** (→ S. [77](#))
- ② [Bibliotheksverwalter] bietet Zugriff auf die zertifizierten Safety-Bibliotheken
→ **Safety-Funktionen von CODESYS nutzen** (→ S. [82](#))
→ **Safety-Funktionen des ifm AS-i Packages nutzen** (→ S. [78](#))
- ③ [Logische E/As] enthält folgende Objekte:
 - logische Austauschgeräte für den Datenaustausch zwischen Standard-SPS und fehlersicherer SPS
→ **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. [85](#))
 - logische Geräte der eingebundenen sicheren AS-i Slaves
→ **Sichere AS-i Slaves konfigurieren** (→ S. [25](#))
 - logische Geräte der sicheren Geräte an der lokalen E/A-Schnittstelle
→ **Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren** (→ S. [32](#))
- ④ [SafetyPOU] enthält den Programm-Code der sicheren Anwendung
→ **Sichere Anwendung erstellen** (→ S. [73](#))
- ⑤ [Safety Task] bietet Zugriff auf die Konfigurationsoptionen des Safety-Tasks
→ **Safety-Task konfigurieren** (→ S. [101](#))

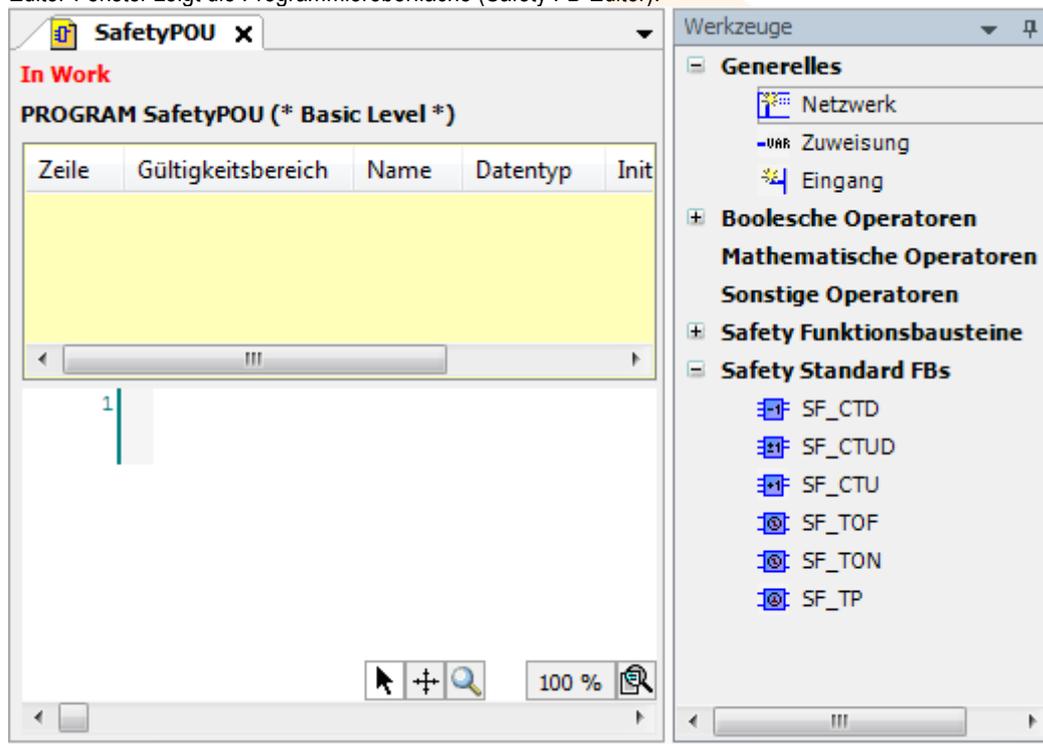
8.2 Sichere Anwendung erstellen

Inhalt	
Unterstützte Programmierbausteine (POUs).....	74
Verfügbare Safety-Bibliotheken.....	75
Unterstützte Programmiersprachen	75
Mindestanzahl an AS-i Slaves.....	76
Variablendeklaration bei Datentransfer zwischen Standard- und Safety-Bereich	76
Verfügbarer Speicherplatz.....	77
Sicheres Projekt pinnen.....	77

6956

Um eine sichere Anwendung zu erstellen:

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
Doppelklick auf [SafetyPOU]
- > Editor-Fenster zeigt die Programmieroberfläche (Safety FB-Editor):



© ifm elec

8.2.1 Unterstützte Programmierbausteine (POUs)

19566

Die Eingabe des Programmcodes erfolgt in den sogenannten POU (= Program Organization Unit). CODESYS Safety bietet 2 POU-Typen für die sichere Programmierung. Sie unterscheiden sich im Umfang der nutzerbaren Funktionen.

Nutzerbare Elemente	Basic POU	Extended POU
Generelle Elemente	Netzwerk, Zuweisung, Eingang	Netzwerk, Zuweisung, Eingang, Sprung, Return
Boolesche Operatoren	AND, OR	AND, OR, XOR, NOT
Sonstige Operatoren	–	SEL, MUX
Mathematische Operatoren	–	ADD, SUB, MUL, DIV, EQ, NE LT, LE, GT, GE
Safety-Standard-Funktionen (safetystandard.lib)	SF_CTD, SF_CTUD, SF_CTU, SF_TOF, SF_TON, SF_TP	SF_RS, SF_SR, SF_CTD, SF_CTUD, SF_CTU, SF_TOF, SF_TON, SF_TP, SF_F_TRIG, SF_R_TRIG
Safety-PLCopen-Funktionen (safetyplopen.lib)	alle	alle



Für die Verwendung von Extended POU's muss der Entwickler Mitglied der Benutzergruppe "Safety.ExtendedLevel" sein (→ **Nutzerkonto anlegen** (→ S. 20)).

Die Verwendung von Extended POU's erfordert aufgrund der höheren Komplexität des Programmcodes einen aufwendigeren Verifikationsprozess.

- Nach Möglichkeit ausschließlich Basic POU's verwenden, um eine sichere Anwendung zu erstellen!



Der während der Erzeugung des Safety-Projekts angelegte "SafetyPOU" ist vom Typ Basic POU (→ **Objekte einer sicheren Anwendung** (→ S. 72)).

8.2.2 Verfügbare Safety-Bibliotheken

419

Für die Erstellung sicherer Anwendungen stehen dem Programmierer folgende Bibliotheken mit zertifizierten Safety-Funktionen zur Verfügung:

Bibliothek	Beschreibung	Referenz
sf_io.library	gerätespezifische Safety-Funktionsbausteine	→ Safety-Funktionen des ifm AS-i Packages nutzen (→ S. 78)
safetystandard.library	sichere Standard-Funktionsbausteine (Bistabile Funktionsbausteine, Zähler, Timer, Trigger)	→ SafetyStandard-Bibliothek (→ S. 82)
safetyplcopen.library	Safety Funktionsbausteine der PLCopen Safety	→ SafetyPLCopen-Bibliothek (→ S. 83)
safetysoemaster.library	Funktionsbausteine für FSoE-Übertragung	→ SafetyFSoEMaster-Bibliothek (→ S. 84)

8.2.3 Unterstützte Programmiersprachen

19471

Folgende Tabelle zeigt, welche Programmiersprachen nach IEC 61131-3 von den zertifizierten Safety-Bibliotheken unterstützt werden:

Bibliothek	Funktionsbausteinsprache (FUP)	Ablaufsprache (AS)	Anweisungsliste (AWL)	Continuous Function Chart (CFC)	Kontaktplan (KOP)	Strukturierter Text (ST)
sf_io.library	X	-	-	-	-	-
safetystandard.library	X	-	-	-	-	-
safetyplcopen.library	X	-	-	-	-	-
safetysoemaster.library	X	-	-	-	-	-

Legende:

X ... wird unterstützt

- ... wird nicht unterstützt

8.2.4 Mindestanzahl an AS-i Slaves

18462

WARNUNG

Gefahr von Datenverlust

In jedem vom AC4S gesteuerten AS-i Netz müssen insgesamt mindestens 5 AS-i Slaves installiert sein (Standard- oder Safety-AS-i Slaves).

- ▶ Wenn weniger als 5 AS-i Slaves zur Konfiguration eines AS-i Netzes gehören, die notwendige Anzahl an AS-i Slaves mithilfe funktionsloser virtueller AS-i Control-Slaves bereitstellen (→ **SF_OUTcontrol_ASi** (→ S. [240](#))).

8.2.5 Variablendeklaration bei Datentransfer zwischen Standard- und Safety-Bereich

7075

Um in einer Anwendung den Variablenwert aus dem jeweils anderen Bereich nutzen zu können, muss die Variable immer zuerst als Austauschvariable deklariert werden. Anschließend kann der Entwickler die so erzeugte Variable direkt im Programmcode der Standard-Anwendung bzw. der sicheren Anwendung einsetzen.



Wenn der Entwickler zuerst eine Variable im Deklarationsteil des POU erzeugt und anschließend eine gleichnamige Austauschvariable definiert, generiert CODESYS eine Fehlermeldung beim Kompilieren der Anwendung.

Beispiel:

Aufgabe:

In der sicheren Anwendung soll ein nicht sicherer boolescher Wert aus der Standard-Anwendung verwendet werden.

Vorgehensweise:

1. Austauschvariable vom Typ BOOL im Austauschgerät [DataFromNonsafe] deklarieren und Variablenliste in [DataToSafe] aktualisieren (→ **Werkseitig angelegte Austauschvariablen nutzen** (→ S. [86](#))).
2. In der Standard-Anwendung der Variable einen Wert zuweisen.
3. In der sicheren Anwendung die Austauschvariable als Quelle für FB oder Zuweisung nutzen.

8.2.6 Verfügbarer Speicherplatz

11571

Für die Programmierung sicherer Anwendungen auf der fehlersicheren SPS steht dem Anwender folgender Speicherplatz zur Verfügung:

- IEC-Programm-Code: 384 KBytes
- IEC-Daten: 128 KBytes

8.2.7 Sicheres Projekt pinnen

17169



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Projekt pinnen
→ Online-Hilfe > Add-Ons > CODESYS Safety > Pinnen

Nach der Fertigstellung einer sicheren Anwendung muss das Projekt gepinnt werden. Dabei werden die Zustände aller Objekte der sicheren Anwendung gespeichert und mit einer Prüfsumme gegen Änderungen gesichert.

8.3 Safety-Funktionen des ifm AS-i Packages nutzen

Inhalt	
Sichere AS-i Slaves: Logisches Gerät rücksetzen	78
Sichere AS-i Slaves: Diagnose-Informationen ausgeben	79
Sichere AS-i Slaves: Hilfssignale HSI_1 und HSI_2 an sichere AS-i Ausgangsmodule senden.....	79
Lokale E/A-Schnittstelle: Logisches Gerät rücksetzen.....	79
Lokale E/A-Schnittstelle: Diagnose-Informationen ausgeben	80
Lokale E/A-Schnittstelle: Auf nicht sichere Ein-/Ausgänge zugreifen	80
PLCopen Safety: Diagnose-Informationen ausgeben.....	81

9009

ifm electronic stellt die Bibliothek `SF_IO.lib` bereit. Sie enthält zertifizierte Funktionsbausteine, mit denen der Programmierer sichere Anwendungen für die fehlersichere SPS des AC4S erstellen kann. Die Bibliothek wird bei der Erstellung eines Safety-Projekts mithilfe der AC4S-Vorlage automatisch geladen. Der Anwender kann über den Bibliotheksverwalter im Safety-Extension-Bereich des Gerätebaums auf die Elemente der Bibliothek zugreifen.



Um Funktionsbausteine aus der ifm-Bibliothek `sf_io.lib` zu einem Netzwerk hinzuzufügen:

- ▶ Netzwerk wählen.
- ▶ [Strg]+[B] aktivieren.
- > Fenster [Eingabehilfe] erscheint.
- ▶ Unter [SF_IO] den gewünschten FB wählen und mit [OK] in das Netzwerk einfügen.

8.3.1 Sichere AS-i Slaves: Logisches Gerät rücksetzen

601

Um das logische Gerät eines sicheren AS-i Eingangs-Slaves aus dem verriegelten Fehlerzustand zurückzusetzen und in den Initialisierungszustand zu wechseln, stehen folgende Funktionsbausteine bereit:

Funktionsbaustein	Beschreibung	Referenz
CtrlASi_InSlave	Sicheren AS-i Eingangs-Slave zurücksetzen, der sich im verriegelten Fehlerzustand befindet	→ CtrlASi_InSlave (→ S. 191)
CtrlASi_ResetAllSlaves	Alle sicheren AS-i Slaves eines AS-i Netzwerks zurücksetzen, die sich im verriegelten Fehlerzustand befinden.	→ CtrlASi_ResetAllSlaves (→ S. 196)



Der sichere AS-i Slave, der mithilfe des FB `CtrlASi_InSlave` gesteuert werden soll, muss im Gerätebaum eingebunden sein!

8.3.2 Sichere AS-i Slaves: Diagnose-Informationen ausgeben

429

Um die Ausgabe der internen Diagnose-Informationen eines logischen Geräts zu steuern, stehen folgende Funktionsbausteine bereit.

Funktionsbaustein	Beschreibung	Referenz
CtrlASi_InSlave	Diagnose-Informationen eines sicheren AS-i Eingangslaves ausgeben	→ CtrlASi_InSlave (→ S. 191)
CtrlASi_OutSlave	Diagnose-Informationen eines AS-i Control Slaves für sichere Ausgänge ausgeben	→ CtrlASi_OutSlave (→ S. 193)



Der sichere AS-i Slave, der mithilfe des FB gesteuert werden soll, muss im Gerätebaum eingebunden sein!

8.3.3 Sichere AS-i Slaves: Hilfssignale HSI_1 und HSI_2 an sichere AS-i Ausgangsmodule senden

427

Um die Hilfssignale HSI_1 und HSI_2 an das logische Gerät eines sicheren AS-i Control Slaves zu senden, steht der folgende Funktionsbaustein bereit:

Funktionsbaustein	Beschreibung	Referenz
CtrlASi_OutSlave	Hilfssignale HS1 und HS2 an AS-i Control Slave für sicheren Ausgänge senden	→ CtrlASi_OutSlave (→ S. 193)



Der sichere AS-i Control-Slave, der mithilfe des FB gesteuert werden soll, muss im Gerätebaum eingebunden sein!

Eine einmal ausgelöste Übertragung der beiden Hilfssignale HSI_1 und HSI_2 kann nicht unterbrochen werden! Hinweise über den Status der Signalübertragung liefert der FB-Ausgang DiagCode.

8.3.4 Lokale E/A-Schnittstelle: Logisches Gerät rücksetzen

434

Um ein logisches Gerät eines sicheren Sensors an der lokalen E/A-Schnittstelle aus dem verriegelten Fehlerzustand zurückzusetzen, stehen folgende Funktionsbausteine bereit:

Funktionsbaustein	Beschreibung	Referenz
CtrlLocalInputs	Sicheres 2-kanaliges Peripheriegerät an der lokalen E/A-Schnittstelle zurücksetzen, das sich im verriegelten Fehlerzustand befindet	→ CtrlLocalInputs (→ S. 198)



Das sichere Gerät, das mithilfe des FB gesteuert werden soll, muss im Gerätebaum eingebunden sein!

8.3.5 Lokale E/A-Schnittstelle: Diagnose-Informationen ausgeben

428

Um die Ausgabe der internen Diagnose-Informationen des logischen Geräts eines sicheren Sensors an der lokalen E/A-Schnittstelle zu steuern, stehen folgende Funktionsbausteine bereit.

Funktionsbaustein	Beschreibung	Referenz
CtrlLocalInputs	Diagnose-Informationen von 2 sicheren Eingängen der lokalen E/A-Schnittstelle ausgeben	→ CtrlLocalInputs (→ S. 198)



Das sichere Gerät, das mithilfe des FB gesteuert werden soll, muss im Gerätebaum eingebunden sein!

8.3.6 Lokale E/A-Schnittstelle: Auf nicht sichere Ein-/Ausgänge zugreifen

426

Um auf die nicht sicheren Ein- und Ausgänge der lokalen E/A-Schnittstelle zuzugreifen, stehen folgende Funktionsbausteine bereit:

Funktionsbaustein	Beschreibung	Referenz
GetLocalInput	Prozessdaten eines nicht sicheren Eingangs der lokalen E/A-Schnittstelle lesen	→ GetLocalInput (→ S. 203)
SetLocalOutput	Prozessdaten an einem nicht sicheren Ausgang der lokalen E/A-Schnittstelle ausgeben	→ SetLocalOutput (→ S. 204)



Die mit dem FB GetLocalInput erfassten Prozessdaten der lokalen Eingänge sind nicht sicher (Datentyp: BOOL).

- ▶ Aus nicht sicheren Prozessdaten implizit keine sichere Funktion ableiten!

Um mit der Standard-SPS des AC4S auf die nicht sicheren Prozessdaten der lokalen Ein- und Ausgänge zugreifen zu können:

- ▶ Globale Austauschvariablen nutzen! (→ **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. [85](#)))

8.3.7 PLCopen Safety: Diagnose-Informationen ausgeben

597

Die Funktionsbausteine der PLCopen-Safety-Bibliothek `SafetyPLCopen.lib` geben am Ausgang `DiagCode` Informationen über ihren inneren Zustand aus. Um diese Diagnose-Informationen in der sicheren Anwendung nutzen zu können, steht der folgende Funktionsbaustein bereit:

Funktionsbaustein	Beschreibung	Referenz
<code>Ctrl_SetDiagInfo</code>	Diagnose-Informationen der Funktionsbausteine der Safety-PLCopen-Bibliothek <code>SafetyPLCopen.lib</code> im OSC ausgeben	→ Ctrl_SetDiagInfo (→ S. 200)



Der FB generiert für jeden Zustandswechsel eine separate Meldung im Online Support Center (OSC). Der Ringspeicher des OSC stößt deshalb nach kurzer Zeit an seine Kapazitätsgrenzen.

- ▶ Den FB `Ctrl_SetDiagInfo` nur für die Fehlersuche nutzen (Debugging)!

8.4 Safety-Funktionen von CODESYS nutzen

15821

3S stellt in der Programmierumgebung CODESYS Safety mehrere Bibliotheken bereit. Sie enthalten zertifizierte Funktionsbausteine (FB), mit denen der Programmierer sichere Anwendungen für die fehlersichere SPS des AC4S erstellen kann. Die Bibliotheken werden bei der Erstellung eines Safety-Projekts mithilfe der AC4S-Vorlage automatisch geladen (→ **Neues Projekt mit AC4S erstellen** (→ S. 17)). Der Anwender kann über den Bibliotheksverwalter im Safety-Extension-Bereich des Gerätebaums auf die Elemente der Bibliotheken zugreifen.

Folgende Bibliotheken stehen zur Verfügung:

- **SafetyStandard-Bibliothek** (→ S. 82)
- **SafetyPLCopen-Bibliothek** (→ S. 83)
- **SafetyFSOEMaster-Bibliothek** (→ S. 84)

8.4.1 SafetyStandard-Bibliothek

20249

Die Bibliothek `SafetyStandard.lib` bietet sichere Varianten von Standard-Funktionen (Zähler, Timer, Trigger, Bistabile Funktionen). Die Bibliothek enthält folgende Funktionen:

Funktionsbaustein	Bedeutung
SF_RS	Sicheres bistabiler Baustein mit dominantem Setzen
SF_SR	Sicheres bistabiler Baustein mit dominantem Rücksetzen
SF_CTD	Sicherer Zähler (rückwärts)
SF_CTUD	Sicherer Zähler (vorwärts)
SF_TOF	Sicherer Timer mit Ausschaltverzögerung
SF_TON	Sicherer Timer mit Einschaltverzögerung
SF_TP	Sicherer Timer mit Pulsgeber
SF_F_TRIG	Sicherer Trigger mit Erkennung fallender Flanken
SF_R_TRIG	Sicherer Trigger mit Erkennung steigender Flanken



Die Verwendung der in CODESYS bereitgestellten Safety-Standard-Funktionsbausteine liegt in der Verantwortung des Entwicklers!

- ▶ Verfügbarkeit der FBs beachten (→ **Unterstützte Programmierbausteine (POUs)** (→ S. 74))!

Die vollständige Dokumentation der `SafetyStandard.lib` kann über die Online-Hilfe von CODESYS Safety abgerufen werden:

→ Online-Hilfe > Add-Ons > CODESYS Safety > Bibliotheken > SafetyStandard

8.4.2 SafetyPLCopen-Bibliothek

20248

Die Bibliothek SafetyPLCopen.lib enthält folgende zertifizierte Funktionsbausteine:

Funktionsbaustein	Bedeutung
SF_Antivalent	Logische UND-Verknüpfung von 2 antivalenten Signalen; Überwachung der Diskrepanzzeit
SF_EDM	Überwachung des Grundzustands und des Schaltzustands von Aktoren, die von sicheren Ausgangsgeräten gesteuert werden
SF_EmergencyStop	Überwachung eines NOT-HALT-Tasters
SF_EnableSwitch	Auswertung der Signale eines handbetätigten, dreistufigen Zustimmungsschalters
SF_Equivalent	Logische UND-Verknüpfung von 2 äquivalenten Signalen; Überwachung der Diskrepanzzeit
SF_ESPE	Überwachung einer berührungslos wirkenden Schutzeinrichtung
SF_GuardLocking	Überwachung einer Schutzeinrichtung mit vierstufiger Verriegelung
SF_GuardMonitoring	Überwachung einer Schutzeinrichtung mit zweistufiger Verriegelung
SF_ModeSelector	Auswahl einer Betriebsart
SF_MutingPar	Paralleles Muting mit 4 Muting-Sensoren
SF_MutingPar_2Sensor	Paralleles Muting mit 2 Muting-Sensoren
SF_MutingSeq	Sequentielles Muting mit 4 Muting-Sensoren
SF_OutControl	Steuerung eines sicheren Ausgangs mit einem nicht-sicheren Signal und optionaler Anlaufsperr
SF_SafetyRequest	Schnittstelle zwischen Benutzerprogramm und Systemumgebung
SF_TestabelSafetySensor	periodisches Testen eines Sensors / einer berührungslos wirkenden Schutzeinrichtung (ESPE) vom Typ 2
SF_TwoHandControlTypeII	Funktion für "Zweihandschaltung Typ 2"
SF_TwoHandControlTypeIII	Funktion für "Zweihandschaltung Typ 3"



Die Verwendung der in CODESYS bereitgestellten Safety-PLCopen-Funktionsbausteine liegt in der Verantwortung des Entwicklers!

Die vollständige Dokumentation der Bibliothek SafetyPLCopen.lib kann über die Online-Hilfe von CODESYS Safety abgerufen werden:

→ Online-Hilfe > Add-Ons > CODESYS Safety > Bibliotheken > SafetyPLCopen

Detaillierte Informationen zu den Spezifikationen: → www.plcopen.org > PLCopen Safety

8.4.3 SafetyFSoEMaster-Bibliothek

11817

Die Bibliothek SafetyFSoEMaster.lib bietet Zugriff auf Funktionen für die sichere Kommunikation über EtherCAT (FailSafe over EtherCAT - FSoE). Die Bibliothek enthält folgende zertifizierte Funktionsbausteine:

Funktionsbaustein	Beschreibung
FSoEMaster	Überwachung der sicheren Übertragung von Daten zwischen EtherCAT-Master und EtherCAT-Slave (Senden / Empfangen)



Die Verwendung der in CODESYS bereitgestellten SafetyFSoEMaster-Funktionsbausteine liegt in der Verantwortung des Entwicklers!

Die vollständige Dokumentation der Bibliothek SafetyFSoEMaster.lib kann über die Online-Hilfe von CODESYS Safety abgerufen werden:

→ Online-Hilfe > Add-Ons > CODESYS Safety > Feldbusse > FSoE

8.5 Datenaustausch zwischen Standard-SPS und fehlersicherer SPS

7085

Standard-SPS und fehlersichere SPS des AC4S sind hardwareseitig getrennt. Der Datenaustausch zwischen beiden Bereichen erfolgt deshalb über einen speziellen Speicherbereich, auf den beide SPSen unabhängig voneinander zugreifen können.

Der Programmierer kann den Datenaustausch zwischen den Bereichen mit dem CODESYS-Mechanismus der logischen Austauschvariablen organisieren.



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - GVL für logischen Datenaustausch
→ Online-Hilfe > Add-Ons > CODESYS Safety > Gerätetopologie > Logische E/As > GVL für Logischen Austausch
 - Logisches E/A für Datenaustausch mit der Hauptsteuerung
→ Online-Hilfe > Add-Ons > CODESYS Safety > Gerätetopologie > Logische E/As > Logisches E/A für Datenaustausch mit der Hauptsteuerung

8.5.1 Hinweise zur Verwendung von Austauschobjekten

18482



Die Variablenwerte zwischen Standard-SPS und fehlersicherer SPS werden grundsätzlich als nicht sichere Daten übertragen!

- ▶ Aus nicht sicheren Variablenwerten implizit keine sicheren Funktionen ableiten!

Befindet sich die Standard-SPS im STOP-Zustand, werden die deklarierten Austauschvariablen für den Datentransfer in Richtung Safety-SPS mit dem Ersatzwert "0" beschrieben.

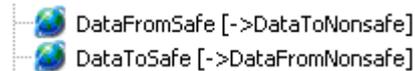
- ▶ Im fehlersicheren Bereich erzeugte sichere Informationen immer unverändert in den Standardbereich übertragen! Im Standardbereich kann das Signal dann weiterverarbeitet werden (z.B. Negation).

8.5.2 Werkseitig angelegte Austauschvariablen nutzen

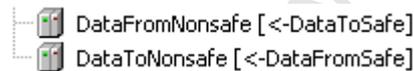
8961

Bei der Erstellung eines AC4S-Projekts mithilfe der AC4S-Vorlage legt CODESYS automatisch Objekte für den Datenaustausch zwischen Standard-SPS und fehlersicherer SPS an. Pro Übertragungsrichtung wird jeweils ein gekoppeltes Paar aus Globaler Variablenliste (GVL) und logischem Austauschgerät generiert.

GVL im Standard-Bereich:



Logische Austauschgeräte im Safety-Bereich:



Mit diesen Austauschobjekten vom Datentyp BYTE können pro Richtung jeweils 8 boolesche Werte gebündelt übertragen werden.

Um Zugriffskonflikte auszuschließen, besitzt pro Austauschrichtung nur eine SPS die Schreibrechte auf das logische Austauschgerät.

Folgende Tabelle zeigt die Zusammenhänge:

Globale Variablenliste (im Standardbereich des Gerätebaums)	Logisches Austauschgerät (im Safety-Extension Bereich des Gerätebaums)	Schreibrechte
[DataToSafe]	[DataFromNonsafe]	Standard-SPS
[DataFromSafe]	[DataToNonsafe]	fehlersichere SPS

Datentransfer Standard-Bereich >>> Safety-Bereich

8962

Um Daten vom Standard-Bereich des Geräts in den Safety-Bereich des Geräts zu übertragen:

1 Austauschvariablen im Safety-Bereich deklarieren

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
Doppelklick auf [DataFromNonsafe]
- > Editor-Fenster zeigt Tabelle für Deklaration der Austauschvariablen.
- ▶ In Spalte [Variable] zeilenweise die Bezeichnungen für Austauschvariablen vom Datentyp BOOL eingeben.

2 Austauschvariablen mit dem Standard-Bereich verknüpfen

- ▶ Im Standardbereich des Gerätebaums:
Doppelklick auf [DataToSafe]
- > Editorfenster zeigt Deklarationsteil der Globalen Variablenliste.
- ▶ [Aktualisieren] aktivieren.
- > CODESYS übernimmt die deklarierte Austauschvariablen aus [DataFromNonsafe] (Schritt 1).
- > Deklarationsteil zeigt die deklarierte Austauschvariablen.



Die logischen Austauschgeräte dürfen ausschließlich im Safety-Bereich definiert werden. Eine Änderung im Standard-Bereich ist nicht erlaubt!

Wenn im logischen Austauschgerät Variablen hinzugefügt oder geändert werden, muss der Entwickler die entsprechende Variablenliste zwingend aktualisieren (Schritt 2)!

Datentransfer Safety-Bereich >>> Standard-Bereich

11187

Um Daten vom Safety-Bereich des Geräts in den Standard-Bereich des Geräts zu übertragen:

1 Austauschvariablen im Safety-Bereich deklarieren

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
Doppelklick auf [DataToNonsafe]
- > Editor-Fenster zeigt Tabelle mit aktuell deklarierten Variablen.
- ▶ In Spalte [Variable] zeilenweise die Bezeichnungen für Austauschvariablen vom Datentyp BOOL eingeben.

2 Austauschvariablen mit dem Standard-Bereich verknüpfen

- ▶ Im Standard-Bereich des Gerätebaums:
Doppelklick auf [DataFromSafe]
- > Editorfenster zeigt Deklarationsteil der Globalen Variablenliste.
- ▶ [Aktualisieren] aktivieren.
- > CODESYS übernimmt die deklarierte Austauschvariable aus [DataToNonsafe] (Schritt 1).
- > Deklarationsteil zeigt die deklarierten Austauschvariablen.



Die logischen Austauschgeräte dürfen ausschließlich im Safety-Bereich definiert werden. Eine Änderung im Standard-Bereich ist nicht erlaubt!

Wenn im logischen Austauschgerät Variablen hinzugefügt oder geändert werden, muss der Entwickler die entsprechende Variablenliste zwingend aktualisieren (Schritt 2)!

8.5.3 Zusätzliche Objekte für Datenaustausch anlegen

19570

Reichen die Kapazitäten der werkseitig angelegten Austauschvariablen nicht aus, kann der Programmierer zusätzliche Austauschvariablen erstellen.

- ▶ Hinweise beachten (→ **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. [85](#)))!



Für den Datenaustausch zwischen Standard-SPS und Safety-SPS dürfen nur folgende Datentypen verwendet werden: BYTE, WORD, LONG

Mehrere sichere Werte vom Datentyp BOOL können gebündelt in einer Variable vom Datentyp BYTE oder WORD übertragen werden.

Die Übertragung von Arrays von Austauschvariablen wird nicht unterstützt!

In jedem Zyklus der Safety-SPS können pro Richtung (Standard-SPS >>> Safety-SPS, Safety-SPS >>> Standard-SPS) maximal 256 Bytes an Daten übertragen werden.

- ▶ Bei der Erzeugung zusätzlicher Objekte für den Datenaustausch die maximal übertragbare Datenmenge beachten!
- ▶ In die Berechnung der Transferdaten folgende Objekte einbeziehen:
 - sichere Querkommunikation (Safety NetVars): → **Sichere Netzwerkvariablen einrichten** (→ S. [89](#))
 - sichere Feldbuskommunikation (FSOE): → **Sichere Feldbuskommunikation** (→ S. [92](#))

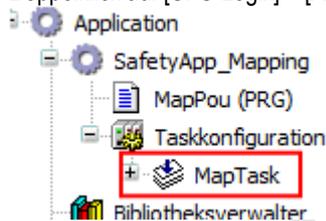
8.5.4 Intervall des Datenaustauschs einstellen

8472

Der Anwender kann einstellen, in welchen Intervallen die Daten zwischen fehlersicherer SPS und Standard-SPS übertragen werden.

Um das Intervall des Variablen-Mappings einzustellen:

- ▶ Im Standardbereich des Gerätebaums:
Doppelklick auf [SPS-Logik] > [Application] > [SafetyApp_Mapping] > [Taskkonfiguration] > [MapTask]



- > Editorfenster zeigt Konfigurationsoptionen für das Variablen-Mapping.
- ▶ In Liste [Typ] den Wert [Zyklisch] wählen.
- ▶ Im Feld [Intervall] das gewünschte Intervall eingeben.
- ▶ Projekt speichern, um die Änderungen zu übernehmen.

8.6 Sichere Querkommunikation

Inhalt	
Sichere Netzwerkvariablen einrichten	89
Sichere Netzwerkvariablen in Anwendung nutzen	91

11525



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Fehlersichere Netzwerkvariablen
→ Online-Hilfe > Add-Ons > CODESYS Safety > CODESYS Safety NetVars

Die Gerätefamilie SmartSPS SafeLine AC4S unterstützt die CODESYS-Funktion Safety NetVars (fehlersichere Netzwerkvariablen). Safety NetVars ermöglichen das Senden und Empfangen von sicheren Daten zwischen 2 fehlersicheren SPS in einem CODESYS-Projekt.

Safety NetVars arbeiten nach dem Sender-Empfänger-Prinzip. Für jede gewünschte Übertragungsrichtung muss ein Sender-Empfänger-Paar erzeugt werden. Ein Sender kann mehrere Empfänger besitzen.

8.6.1 Sichere Netzwerkvariablen einrichten

21673



- ▶ Hinweise zur Konfiguration der Ethernet-Konfigurationsschnittstellen 1 (X3) und 2 (X8) beachten: → **IP-Einstellungen ändern** (→ S. 40)

Maximal 10 Verbindungen werden unterstützt (Sender >>> Empfänger). Pro Sender/Empfänger-Paar dürfen maximal 32 Bytes übertragen werden.

- ▶ Um die Größe der Transferdaten pro Sender/Empfänger-Paar zu bestimmen, folgende Formeln nutzen.
 - Safety NetVars (Sender):
Transferdaten = $\text{Summe}_{\text{SafetyNetVars}(\text{Sender})} * 2 + 10 \text{ Bytes}$
 $\text{Summe}_{\text{SafetyNetVars}(\text{Sender})}$: Größe aller deklarierten Variablen addieren (bis zu 16 SAFEBOOL = 2 Bytes, jedes SAFEINT = 2 Bytes, jedes SAFEWORD = 2 Bytes)
 - Safety NetVars (Empfänger):
Transferdaten = $\text{Summe}_{\text{SafetyNetVars}(\text{Empfänger})} * 2 + 10 \text{ Bytes}$
 $\text{Summe}_{\text{SafetyNetVars}(\text{Empfänger})}$: Größe aller deklarierten Variablen addieren (bis zu 16 SAFEBOOL = 2 Bytes, jedes SAFEINT = 2 Bytes, jedes SAFEWORD = 2 Bytes)

In jedem Zyklus der fehlersicheren SPS können pro Richtung maximal 256 Bytes zwischen Standard-SPS und Safety-SPS übertragen werden.

- ▶ Bei der Dimensionierung der Safety NetVars die maximal übertragbare Datenmenge beachten!
- ▶ In die Berechnung der Transferdaten folgende Objekte einbeziehen:
 - logische Austauschobjekte: → **Datenaustausch zwischen Standard-SPS und fehlersicherer SPS** (→ S. 85)
 - sichere Feldbuskommunikation (FSoE): → **Sichere Feldbuskommunikation** (→ S. 92)

Um sichere Daten von AC4S 1 an AC4S 2 zu senden:

Voraussetzungen

- > AC4S 1 und AC4S 2 sind Teilnehmer des selben CODESYS-Projekts (→ **Zusätzliches Gerät zum Projekt hinzufügen** (→ S. 41)).
- > Die AC4S-Geräte 1 und 2 sind über Ethernet-Netzwerk gekoppelt.

1 Netzwerkvariablenliste im Sender erzeugen

- ▶ In der Safety-Extension des Gerätebaums des AC4S 1: [Safety Logik] > [SafetyApp] wählen.
- ▶ [Project] > [Objekt hinzufügen] > [Safety Netzwerkvariablenliste (Sender)] wählen.
- > Dialogfenster erscheint.
- ▶ Im Feld [Name] eine eindeutige Bezeichnung für die Liste eingeben.
- ▶ Auf [Hinzufügen] klicken.
- > Editor-Fenster zeigt Konfiguration der Variablenliste des Senders.

2 Sichere Variablen deklarieren

- ▶ Rechtsklick in Editor-Fenster.
- ▶ Im Kontextmenü [Variablendeklaration einfügen] wählen.
- > Dialogfenster erscheint
- ▶ Parameter der sicheren Variable wie gewünscht einstellen.
- ▶ Auf [OK] klicken, um die sichere Variable zur Netzwerkvariablenliste hinzuzufügen.
- > Editor-Fenster zeigt deklarierte Variable.
- ▶ Optional: Schritt 2 wiederholen, um zusätzliche sichere Variablen zur Netzwerkvariablenliste hinzuzufügen.

3 Netzwerkvariablenliste im Empfänger erzeugen

- ▶ In der Safety-Extension des Gerätebaums des AC4S 2: [Safety Logik] > [SafetyApp] wählen.
- ▶ [Project] > [Objekt hinzufügen] > [Safety Netzwerkvariablenliste (Empfänger)] wählen.
- > Dialogfenster erscheint.
- ▶ Im Feld [Name] eine eindeutige Bezeichnung für die Liste eingeben.
- ▶ Auf [Hinzufügen] klicken.
- > Editor-Fenster zeigt Konfiguration der Variablenliste des Empfängers.

4 Netzwerkvariablenlisten koppeln

- ▶ Doppelklick auf erzeugte Netzwerkvariablenliste (Empfänger)
- ▶ Editor-Fenster zeigt Konfiguration der Variablenliste.
- ▶ In Liste [Zugeordnete Sender] die gewünschte Netzwerkvariablenliste (Sender) wählen.
- ▶ Auf [Aktualisieren] klicken.
- ▶ CODESYS koppelt die Netzwerkvariablenlisten von Sender und Empfänger.
- > Editor-Fenster zeigt die sicheren Variablen der gekoppelten Netzwerkvariablenliste.

5 Optional: Weitere sichere Querkommunikation einrichten

- ▶ Schritte 1 bis 3 für entgegengesetzte Übertragungsrichtung wiederholen.

8.6.2 Sichere Netzwerkvariablen in Anwendung nutzen

10973

WARNUNG

Gefahr von Personen- und Sachschäden

Im Debug-Modus führt das Stoppen einer fehlersicheren SPS weder zur Unterbrechung der sicheren Querkommunikation noch zum Rücksetzen von gerade aktiven Signalen führt, d.h. empfangenden fehlersicheren SPSen reagieren nicht. Dieses Verhalten ist gewollt und dient dem Test und der Fehlerbeseitigung des Systems.

- ▶ Im Debug-Modus sicherstellen, dass sich zu keinem Zeitpunkt Personen im Gefahrenbereich der Anlage aufhalten.

Um die eingerichteten Safety NetVars nutzen zu können, müssen sie in der sicheren Anwendung der AC4S-Geräte aktiviert werden.

Voraussetzungen:

- > Netzwerkvariablenlisten in Sender und Empfänger wurden erzeugt und korrekt konfiguriert (→ **Sichere Netzwerkvariablen einrichten** (→ S. 89)).

1 Netzerkvariablenliste des Senders aktivieren

- ▶ Sichere Anwendung des AC4S (Sender) öffnen
- ▶ FB NetVarSender einfügen
- ▶ Als Instanznamen die Bezeichnung der Netzwerkvariablenliste (Sender) wählen.
- ▶ FB konfigurieren (→ CODESYS-Hilfe)

2 Netzwerkvariablenliste des Empfängers aktivieren

- ▶ Sichere Anwendung des AC4S (Empfänger) öffnen.
- ▶ FB NetVarReceiver einfügen.
- ▶ Als Instanznamen die Bezeichnung der Netzerkvariablenliste (Empfänger) wählen.
- ▶ FB konfigurieren (→ CODESYS-Hilfe)

3 Optional: Weitere Netzwerkvariablenlisten aktivieren

- ▶ Schritte 1 und 2 wiederholen, um zusätzliche Netzwerkvariablenlisten zu aktivieren.

8.7 Sichere Feldbuskommunikation

Inhalt	
FSoE-Verbindung konfigurieren	92
Hinweise zu FB FSoEMaster	93

19865



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Fail Safe over EtherCAT
 - Online-Hilfe > Add-Ons > CODESYS Safety > Feldbusse > FSoE
- ▶ Hinweise zur Konfiguration der Ethernet-Konfigurationsschnittstellen 1 (X3) und 2 (X8) beachten: → **IP-Einstellungen ändern** (→ S. [40](#))

Das Gerät unterstützt folgende Protokolle für die sichere Kommunikation über Feldbusse:

- Fail Safe over EtherCAT (FSoE)

8.7.1 FSoE-Verbindung konfigurieren

21674



- ▶ Für jedes fehlersichere FSoE-Ein-/Ausgangsmodul eine Instanz des FB FSoEMaster nutzen!
- ▶ Hinweise zum FB FSoEMaster beachten! (→ **Hinweise zu FB FSoEMaster** (→ S. [93](#)))
 In einem FSoE-Netzwerk müssen FSoE-Adressen und Connection-IDs eindeutig sein. CODESYS erkennt nicht, wenn eine FSoE-Adresse oder eine Connection-ID mehrfach vergeben wird.

Um über eine FSoE-Verbindung auf die sicheren Ein- /Ausgangsdaten des EtherCAT Slaves zuzugreifen:

Voraussetzungen

- > AC4S ist über Ethernet-Netzwerk mit EtherCAT Slave verbunden.
- > AC4S ist als EtherCAT Master konfiguriert (→ **EtherCAT Master hinzufügen** (→ S. [43](#))).

1 Sicheren EtherCAT Slave hinzufügen

- ▶ Sicheren EtherCAT Slave hinzufügen (→ **EtherCAT Slave hinzufügen und konfigurieren** (→ S. [45](#)))
- > Sicheres Modul des EtherCAT Slaves erscheint im sicheren Bereich des Gerätebaums unter [Logische E/As].

2 Sicheren EtherCAT Slave konfigurieren

- ▶ Im sicheren Bereich des Gerätebaums unter [Logische E/As]: Doppelklick auf sicheres Modul des EtherCAT Slaves
- ▶ Registerkarte [Sichere Konfiguration] wählen.
- > Geräteeditor zeigt Konfiguration des sicheren Moduls des EtherCAT Slaves.
- ▶ Folgende Parameter wie gewünscht einstellen:

Parameter	Beschreibung	Mögliche Werte
[FSoE address]	FSoE-Adresse des sicheren EtherCAT Slaves	1...255 (abhängig von FSoE-Slave)
[Connection ID]	Nummer der Verbindung zum sicheren EtherCAT Slave	1...65535
[Watchdog Time]	Watchdog-Zeit	

3 Sichere Ein-/Ausgangssignale des EtherCAT Slaves auf Variablen mappen

- ▶ Im sicheren Bereich des Gerätebaums unter [Logische E/As]: Doppelklick auf sicheres Modul des EtherCAT Slaves
- ▶ Registerkarte [I/O Abbild] wählen.
- > Geräteeditor zeigt Variablen-Mapping der sicheren Ein-/Ausgangssignale.
- ▶ Sichere Ein-/Ausgangssignale des EtherCAT Slaves auf Variablen mappen.

4 Auf sichere Daten des EtherCAT Slaves zugreifen

- ▶ Sichere Anwendung des AC4S öffnen.
- ▶ FB FSoEMaster einfügen.
- ▶ FB FSoEMaster wie gewünscht konfigurieren.

8.7.2 Hinweise zu FB FSoEMaster

18586



- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Funktionsbaustein FSoEMaster
 - Online-Hilfe > Add-Ons > CODESYS Safety > Feldbusse > FSoE > FSoEMaster

Die Dokumentation des FB FSoEMaster in der CODESYS-Online-Hilfe ist nicht korrekt.

- ▶ Beim Einsatz des FB FSoEMaster folgendes beachten:
 - Der Parameter S_ActivateIN muss immer auf TRUE gesetzt werden, unabhängig davon, ob es sich um ein EtherCAT-Eingangs- oder Ausgangsmodul handelt.
 - Der Diagnosecode 0x80nn zeigt den aktuellen Zustand des FSoEMaster. Bits nn können folgende Werte einnehmen:

Wert	Zustand	Beschreibung	
0x64	Reset	Die Verbindung sind zurückgesetzt (Ausgänge sind im sicheren Zustand)	
0x65	Session	Es wird die Session-ID übertragen (Ausgänge sind im sicheren Zustand)	
0x66	Connection	Es wird die Verbindungs-ID übertragen (Ausgänge sind im sicheren Zustand)	
0x67	Parameter	Es werden die Parameter übertragen (Ausgänge sind im sicheren Zustand)	
0x68	Data	Prozessdaten oder Ersatzwerte werden übertragen. Der Ausgang S_ReadyIN zeigt, welche Daten übertragen werden:	
		FALSE	Übertragung eines Ersatzwerts bedingt durch einen erkannten Fehler im IO-Modul
		TRUE	Übertragung Prozessdaten

8.8 Prozesssicherheitszeit einstellen

Inhalt	
Prozesssicherheitszeit beim AC4S	94
Variable Komponenten der Reaktionszeit einstellen.....	100
	16681

8.8.1 Prozesssicherheitszeit beim AC4S

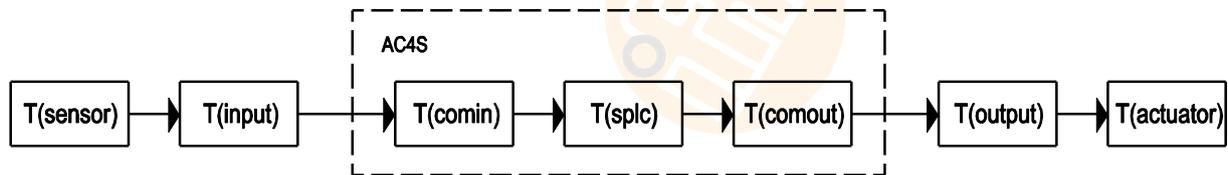
17130



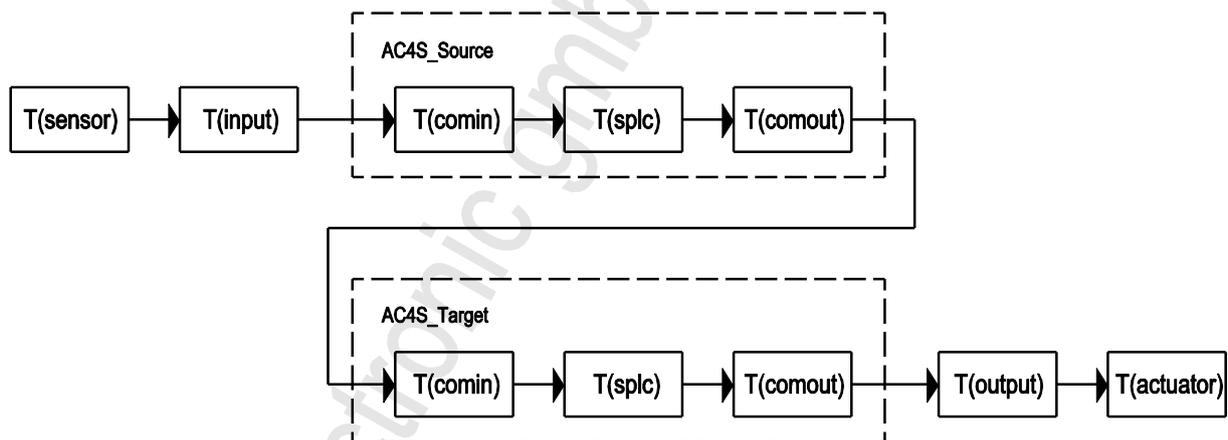
Grundsätzliche Informationen: → Abschnitt "Prozesssicherheitszeit" im Original-Gerätehandbuch

Bei der Berechnung der Prozesssicherheitszeit des Gesamtsystems müssen folgenden Szenarien unterschieden werden:

- Ohne sichere Querkommunikation



- Mit sicherer Querkommunikation



Die folgende Tabelle zeigt die Bedeutung der einzelnen Komponenten des AC4S:

Komponente	Beschreibung	Mögliche Werte	
T(sensor)	Reaktionszeit des Safety Sensors	→ Betriebsanleitung / Datenblatt	
T(input)	Reaktionszeit der Safety Eingangsgeräts	→ Betriebsanleitung / Datenblatt ▪ bei lokalem Eingang: 0 ms	
T(comin)	Reaktionszeit des Eingangskommunikationskanals (ab Klemme)	AS-i	30ms
		Local IO	16ms
		FSoE	32ms + 4*T(MapTask) + 4*T(MainTask)
		Safety NetVars	32ms + 4*T(MapTask) + 4*T(MainTask)
T(splc)	Verarbeitungszeit des fehlersicheren SPS	2*T(SafetyTask)	
T(comout)	Reaktionszeit des Ausgangskommunikationskanals (bis Klemme)	AS-i	15ms
		Local IO	8ms
		FSoE	32ms + 4*T(MapTask) + 4*T(MainTask)
		Safety NetVars	32ms + 4*T(MapTask) + 4*T(MainTask)
T(output)	Reaktionszeit der Ausgangsverarbeitungseinheit	→ Betriebsanleitung / Datenblatt ▪ bei lokalem Ausgang: 0ms	
T(actuator)	Reaktionszeit des Aktuators	→ Betriebsanleitung / Datenblatt	



Die Prozesssicherheitszeit verlängert sich um 1 SPS-Zyklus, wenn der sichere AS-i Eingangs-Slave bzw. die sicheren lokalen Eingangskanäle in CODESYS mithilfe der folgenden logischen Geräte konfiguriert wird:

- **SF_IN_ASi_dependent_filter_nshutdown** (→ S. [236](#))
- **SF_IN_local_dependent_filter_nshutdown** (→ S. [267](#))

Schritt 1: Typische Reaktionszeiten berechnen

10931

► Typische Reaktionszeiten des Sicherheitsfunktion des Geräts berechnen:

- Ohne sichere Querkommunikation

	AS-i Ausgang	Lokaler Ausgang	FSoE Slave Ausgang
AS-i Eingang	30 ms + 2*T(SafetyTask) + 15 ms	30 ms + 2*T(SafetyTask) + 8 ms	30 ms + 2*T(SafetyTask) + 4*T(MainTask) + 4*T(MapTask) + 32 ms
Lokaler Eingang	16 ms + 2*T(SafetyTask) + 15 ms	16 ms + 2*T(SafetyTask) + 8 ms	16 ms + 2*T(SafetyTask) + 4*T(MainTask) + 4*T(MapTask) + 32 ms
FSoE Slave Eingang	32 ms + 4*T(MainTask) + 4*T(MapTask) + 2*T(SafetyTask) + 15 ms	32 ms + 4*T(MainTask) + 4*T(MapTask) + 2*T(SafetyTask) + 8 ms	32 ms + 4*T(MainTask) + 4*T(MapTask) + 2*T(SafetyTask) + 4*T(MainTask) + 4*T(MapTask) + 32 ms

- Mit sicherer Querkommunikation (Safety NetVars)

	nach Gerät 2		
von Gerät 1	AS-i Ausgang	Lokaler Ausgang	FSoE Slave Ausgang
AS-i Eingang	30 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 15 ms	30 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 8 ms	30 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms
Lokaler Eingang	16 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 15 ms	16 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 8 ms	16 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms
FSoE Slave Eingang	4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 15 ms	4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 8 ms	4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms



Die Formeln in der Tabelle bestimmen nur die typische Reaktionszeit des Geräts. Bei der Berechnung der typischen Reaktionszeit des Prozesses müssen zusätzlich die Verzögerungszeiten der Safety AS-i Sensoren/Aktuatoren und der jeweiligen Eingangs-Ausgangsgeräte addiert werden (T(sensor) + T(input) und T(output) + T(aktuator)).

Schritt 2: Reaktionszeit der Sicherheitsfunktion berechnen

11231



- ▶ Messungen für an realer Anlage durchführen.
- ▶ Watchdog-Zeiten und Tasküberwachungen ausreichend groß dimensionieren.

- ▶ In allen Abschnitten des Signallaufpfads mit sicherer Feldbuskommunikation oder sicherer Querkommunikation die maximale Reaktionszeit messen.
 - sichere Feldbuskommunikation (FSoE): Ausgang tMaxRespTimeMS des FB FSoEMaster
 - sichere Querkommunikation (Safety NetVars): Ausgang tMaxRespTime MS des FB NetVarReceiver
- ▶ Mit gemessenen Werten die Reaktionszeit der Sicherheitsfunktion berechnen:
 - ohne sichere Querkommunikation:

	AS-i Ausgang	Lokaler Ausgang	FSoE Slave Ausgang
AS-i Eingang	-	-	30 ms + 2*T(SafetyTask) + tMaxRespTimeMS(FSoE_out)
Lokaler Eingang	-	-	16 ms + 2*T(SafetyTask) + tMaxRespTimeMS(FSoE_out)
FSoE Slave Eingang	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask) + 15 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask) + 8 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask) + tMaxRespTimeMS(FSoE_out)

- mit sicherer Querkommunikation:

nach Gerät 2			
von Gerät 1	AS-i Ausgang	Lokaler Ausgang	FSoE Slave Ausgang
AS-i Eingang	30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 15 ms	30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 8 ms	30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out)
Lokaler Eingang	16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 15 ms	16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 8 ms	16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out)
FSoE Slave Eingang	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 15 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 8 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out)



Die Formeln in der Tabelle bestimmen nur die maximale Reaktionszeit der Geräte. Bei der Berechnung der maximalen Reaktionszeit des Prozesses müssen zusätzlich die Verzögerungszeiten der Safety AS-i Sensoren/Aktuatoren und der jeweiligen Eingangs-Ausgangsgeräte addiert werden (T(sensor) + T(input) und T(output) + T(aktuator)).

Schritt 3: Watchdog-Zeiten definieren

10918

Sichere Querkommunikation und sichere Feldbuskommunikation werden mithilfe von Watchdogs überwacht. Die Watchdogzeiten müssen hinreichend groß gewählt werden.

- ▶ Aus den in Schritt 2 gemessenen maximalen Reaktionszeiten der Sicherheitsfunktion die Watchdog-Zeiten bestimmen. ifm empfiehlt folgende Formeln:
 - sichere Querkommunikation: $T(\text{WD_SafetyNetVars}) = 2 * T(\text{tMaxRespTimeMS})$
 - sichere Feldbuskommunikation: $T(\text{WD_FSoE}) = 2 * T(\text{tMaxRespTimeMS})$
- ▶ Watchdog-Zeiten einstellen (**Variable Komponenten der Reaktionszeit einstellen** (→ S. [100](#))).

Schritt 4: Worst-Case-Reaktionszeit der Sicherheitsfunktion berechnen

10922

Die Worst-Case-Prozesssicherheitszeit

- ▶ Aus den berechneten und gemessenen Werten den Worst-Case-Wert berechnen.
 - Ohne sichere Querkommunikation:

	AS-i Ausgang	Lokaler Ausgang	FSoE Slave Ausgang
AS-i Eingang	-	-	$T(WD_FSoE)$
Lokaler Eingang	-	-	$T(WD_FSoE)$
FSoE Slave Eingang	$T(WD_FSoE) + 2 * T(SafetyTask) + 15 \text{ ms}$	$T(WD_FSoE) + 2 * T(SafetyTask) + 8 \text{ ms}$	$T(WD_FSoE_in) + 2 * T(SafetyTask) + 2 * tMaxRespTimeMS(FSoE_out)$

- Mit sicherer Querkommunikation:

	nach Gerät 2		
von Gerät 1	AS-i Ausgang	Lokaler Ausgang	FSoE Slave Ausgang
AS-i Eingang	$\text{MAX}(30 \text{ ms} + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 15 \text{ ms}; T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 15 \text{ ms})$	$\text{MAX}(30 \text{ ms} + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 8 \text{ ms}; T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 8 \text{ ms})$	$\text{MAX}(30 \text{ ms} + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_FSoE_out))$
Lokaler Eingang	$\text{MAX}(16 \text{ ms} + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 15 \text{ ms}; T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 15 \text{ ms})$	$\text{MAX}(16 \text{ ms} + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 8 \text{ ms}; T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 8 \text{ ms})$	$\text{MAX}(16 \text{ ms} + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_FSoE_out))$
FSoE Slave Eingang	$\text{MAX}(T(WD_FSoE_in) + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 15 \text{ ms}; T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 15 \text{ ms})$	$\text{MAX}(T(WD_FSoE_in) + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 8 \text{ ms}; T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + 8 \text{ ms})$	$\text{MAX}(T(WD_FSoE_in) + 2 * T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2 * T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_SafetyNetVars) + 2 * T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_FSoE_out))$



Zu den errechneten Worst-Case-Zeiten müssen jeweils die Reaktionszeiten der Ausgangsseite addiert werden ($T(\text{output}) + T(\text{actuator})$).

8.8.2 Variable Komponenten der Reaktionszeit einstellen

8898

Die variablen Komponenten, die die Prozesssicherheitszeit des Gesamtsystems beeinflussen, kann der Anwender an folgenden Punkten innerhalb des CODESY-Projekts einstellen:

- T(MainTask):



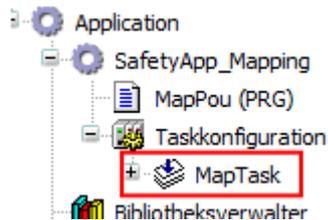
- Ort: Standardbereich des Gerätebaums
- Knoten: [MainTask]
- Parameter: [Intervall]

- T(SafetyTask):



- Ort: Safety-Extension des Gerätebaums
- Knoten: [Safety Task]
- Parameter: [Zykluszeit]

- T(MapTask):



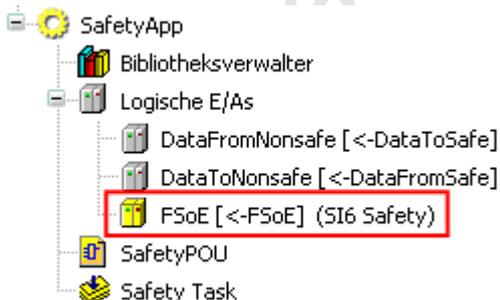
- Ort: Standardbereich des Gerätebaums
- Knoten: [MapTask]
- Parameter: [Intervall]

- T(WD_SafetyNetVars):



- Ort: Safety-Extension des AS-i Gerätebaums
- Knoten: Safety Netzwerkvariable (Empfänger)
- Parameter: [Watchdog-Zeit]

- T(WD_FS0E):



- Ort: Safety-Extension des AS-i Gerätebaums
- Knoten: [Logische E/As] > Sicheres EtherCAT-Modul
- Parameter: [Watchdog Time]

8.9 Safety-Task konfigurieren

2631



Die Konfiguration der Safety-Tasks der sicheren Anwendung erfolgt über den "Safety Task Editor" des Programmiersystems CODESYS Safety.

- ▶ Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!
 - Safety Task Editor
→ Online-Hilfe > Add-Ons > CODESYS Safety > Editoren > Safety Task Editor

8.9.1 Zykluszeit der fehlersicheren SPS einstellen

18442

Der Programmierer kann die Zykluszeit der fehlersicheren SPS frei einstellen. Gültig sind ganzzahlige Wert im Bereich 10...100 ms.

Um die Zykluszeit der fehlersicheren SPS einzustellen:

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
Doppelklick auf [Safety Logik] > [SafetyApp] > [Safety Task]
- > Editor-Fenster zeigt Konfigurationsoptionen des Safety-Tasks.
- ▶ Im Feld [Zykluszeit] die gewünschte Zykluszeit eingeben.
- ▶ In der Liste die Safety-POUs wählen, für die die eingestellte Zykluszeit gelten soll.
- > Die eingestellte Zykluszeit gilt für alle gewählten Safety-POUs.

9 Betrieb

Inhalt	
CODESYS-Projekt auf Gerät übertragen	102
Betriebszustände AC4S	107
Reset	112
Web-Visualisierung anzeigen	115

18492

9.1 CODESYS-Projekt auf Gerät übertragen

20257

Um das CODESYS-Projekt auf dem AC4S zu speichern, müssen folgende Komponenten separat auf das Gerät übertragen werden:

- Standard-Anwendung "Application" (→ **Standard-Anwendung auf AC4S laden** (→ S. [103](#)))
- sichere Anwendung "SafetyApp" (→ **Sichere Anwendung auf AC4S laden** (→ S. [104](#)))



- ▶ Hinweise zu den Betriebsarten der Standard-SPS und der fehlersicheren SPS des AC4S beachten!
→ **Betriebszustände AC4S** (→ S. [107](#))

ifm-Systemlösungen und vom Anwender erstellte CODESYS-Anwendungen dürfen nicht gleichzeitig auf dem AC4S gespeichert und ausgeführt werden!

- ▶ Vor dem Laden einer Standard-Anwendung auf das AC4S alle auf dem Gerät gespeicherten ifm-Systemlösungen löschen (→ Gerätehandbuch, ifm-Apps deinstallieren)!

Um eine ifm-Systemlösung in einem Anwender-Projekt nutzen zu können, müssen die Funktionen über separat erhältliche Bibliotheken in das Projekt eingebunden werden.

- ▶ AS-i Fachberater der ifm electronic kontaktieren!

9.1.1 Standard-Anwendung auf AC4S laden

15020



Bei gleichzeitiger Verwendung von Netzwerkvariablen und Safety Netzwerkvariablen: →
Hinweis: Projekte mit Netzwerkvariablen (→ S. [105](#))

Um die Standard-Anwendung des CODESYS-Projekts auf das AC4S zu übertragen:

Voraussetzungen:

- > Kommunikationspfad der Standard-SPS ist korrekt eingestellt (→ **Programmierschnittstelle konfigurieren** (→ S. [21](#))).
- > Alle ifm-Systemlösungen auf dem Gerät wurden gelöscht (→ Gerätehandbuch: ifm-Apps deinstallieren).

1 Standard-Anwendung auf Gerät laden

- ▶ Im Standardbereich des Gerätebaums:
Standard-Anwendung markieren und mit [Projekt] > [Aktive Applikation setzen] aktivieren.
- ▶ [Online] > [Einloggen] wählen.
- > CODESYS geht in den Online-Modus.
- > CODESYS kompiliert die Standard-Anwendung und überträgt sie auf das AC4S.

2 Standard-Anwendung starten

- ▶ [Application] starten.
- > [Application] ist im RUN-Zustand.

3 Optional: Bootapplikation erzeugen

- ▶ [Online] > [Bootapplikation erzeugen] wählen.
- > CODESYS speichert die Standard-Anwendung spannungsausfallsicher auf dem AC4S.



Ist der Parameter [Erzeuge implizit Bootapplikation beim Download] in den Eigenschaften der Applikation aktiviert, kann Schritt 3 entfallen.

Um zu prüfen, ob der Parameter aktiviert ist:

- ▶ [Application] markieren.
- ▶ [Ansicht] > [Eigenschaften] wählen.
- > Registerkarte [Bootapplikation] zeigt aktuellen Parameterwert.

9.1.2 Sichere Anwendung auf AC4S laden

20385



Bei gleichzeitiger Verwendung von Netzwerkvariablen und Safety Netzwerkvariablen: →
Hinweis: Projekte mit Netzwerkvariablen (→ S. [105](#))

Beim ersten Einloggen auf die fehlersichere Steuerung wird der Programmierer aufgefordert, eine Instanzidentifikation einzugeben. Die Instanzidentifikation ist die Seriennummer des Geräts ohne die führenden Nullen.

Beispiel:

- Seriennummer: 00000569158
- Instanzidentifikation: 569158

Um die Seriennummer des Geräts zu ermitteln:

- Aufkleber unterhalb der Feldbus-Schnittstellen (X6/X7) oder
- GUI/Web-Interface:



Um die sichere Anwendung des CODESYS-Projekts auf das AC4S zu übertragen:

Voraussetzungen:

- > Kommunikationspfad der fehlersicheren SPS ist korrekt eingestellt (→ **Programmierschnittstelle konfigurieren** (→ S. [21](#))).

1 Sichere Anwendung auf AC4S laden

- ▶ Im Safety-Extension-Bereich des Gerätebaums:
[SafetyApp] markieren und mit [Projekt] > [Aktive Applikation setzen] aktivieren.
- ▶ [Online] > [Einloggen] wählen.
- > CODESYS geht in den Online-Modus.
- > CODESYS kompiliert die sichere Anwendung und überträgt sie auf das AC4S.
- ▶ Warnmeldungen und Hinweise beachten!

2 Sichere Anwendung starten

- ▶ [SafetyApp] starten.
- > [SafetyApp] ist im RUN-Zustand.

3 Optional: Codefolgen einlernen

- ▶ Codefolgen der sicheren AS-i Eingangs-Slaves einlernen (→ **Prozesssicherheitszeit beim AC4S** (→ S. [94](#))).

4 Optional: Sichere Bootapplikation erzeugen

- ▶ Sichere Anwendung pinnen (→ CODESYS-Online-Hilfe).
- ▶ [Online] > [Bootapplikation erzeugen] wählen.
- > CODESYS speichert die sichere Anwendung spannungsausfallsicher auf dem AC4S.



Nachdem die sichere Bootapplikation auf dem AC4S erzeugt wurde, bleibt die fehlersichere SPS weiterhin im Betriebszustand "Debug-Betrieb"!

- ▶ Hinweise zum Wechsel der Betriebszustände beachten (→ **Zwischen den Zuständen wechseln** (→ S. [111](#)))

9.1.3 Hinweis: Projekte mit Netzwerkvariablen

15210

Wenn Netzwerkvariablen und Safety Netzwerkvariablen gleichzeitig genutzt werden, dann muss der Anwender folgende Reihenfolge bei der Übertragung der Anwendungen auf das Gerät beachten:

- 1 **Safety Netzwerkvariablen erstellen**
 - ▶ Gerät 1: Safety Netzwerkvariablenliste (Sender) erstellen und konfigurieren
 - ▶ Gerät 2: Safety Netzwerkvariablenliste (Empfänger) erstellen und konfigurieren
 - ▶ Optional: Weitere Sender-Empfänger-Paare erstellen und konfigurieren
- 2 **Anwendungen auf Geräte übertragen**
 - ▶ Auf allen Geräten: In Standard-SPS und die fehlersichere SPS einloggen.
 - ▶ Boot-Applikationen auf beide SPSen übertragen
 - ▶ Prüfen, ob Safety NetVars korrekt funktionieren.
- 3 **Ausloggen**
 - ▶ Aus der Standard-SPS ausloggen.
- 4 **Standard-Netzwerkvariablen erstellen**
 - ▶ Gerät1: Netzwerkvariablenliste (Sender) erstellen und konfigurieren.
 - ▶ Gerät 2: Netzwerkvariablenliste (Empfänger) erstellen und konfigurieren.
 - ▶ Optional: Weitere Sender-Empfänger-Paare erstellen und konfigurieren.
- 5 **Anwendung auf Geräte übertragen**
 - ▶ Auf allen Geräten: In Standard-SPS einloggen.
 - ▶ Boot-Applikation auf Standard-SPS übertragen.
 - ▶ Prüfen, ob Netzwerkvariablen und Safety Netzwerkvariablen korrekt funktionieren.

9.1.4 Codefolgen der sicheren AS-i Slaves einlernen (teachen)

18455

Um die Codefolge eines sicheren AS-i Slaves einzulernen (teachen):

Voraussetzungen:

- > Sicherer AS-i Slave ist mit AC4S verbunden und adressiert.
- > Alle sicheren AS-i Eingangs-Slaves sind freigeschaltet (z.B Not-Halt-Schalter: entriegelt).
- > Sichere Anwendung ist auf Gerät gespeichert und gestartet (→ **Sichere Anwendung auf AC4S laden** (→ S. [104](#))).
- > Code-Tabelle ist leer.

1 Codefolgen eines sicheren AS-i Slaves einlernen

- > AC4S fordert Testung für sicheren AS-i Eingangs-Slave an (OSC-Meldung).
 - ▶ Sicheren AS-i Eingangs-Slave betätigen und wieder freischalten (Schaltfolge: verriegeln ⇒ entriegeln)
- > Sicherer AS-i Slave sendet Codefolge an AC4S.
- > AC4S füllt Code-Tabelle mit Codefolge des sicheren AS-i Eingangs-Slaves.
- > Codefolge des sicheren AS-i Eingangs-Slaves ist eingelernt.

2 Optional: Weitere Codefolgen einlernen

- ▶ Schritt 1 für zusätzliche sichere AS-i Eingangs-Slaves wiederholen.

9.1.5 Boot-Applikation per SD-Karte löschen

15970

Für den Fall, dass nach dem Start einer komplexen Boot-Applikation das Gerät überlastet ist und nicht mehr auf auf Bedieneingaben oder Login-Versuche reagiert, muss die Boot-Applikation auf dem Gerät zwangsweise gelöscht werden.

Um die Boot-Applikation auf dem Gerät zu löschen:

- ▶ Schreibschutz der SD-Karte deaktivieren
- ▶ Im Wurzelverzeichnis der SD-Karte eine Datei mit dem Namen `KillBootApplication.txt` erstellen.
- ▶ SD-Karte in den Steckplatz für SD-Karten des Geräts stecken
- ▶ Gerät neu starten.
- > Boot-Applikation auf der geräteinternen Standard-SPS wird gelöscht.
- > Datei `KillBootApplication.txt` auf SD-Karte wird in `KillBootApplication.rdy` umbenannt.



Bei dieser Methode werden folgende Daten auf der geräteinternen Standard-SPS entfernt:

- alle Dateien der Boot-Applikation
- alle CRC-Dateien
- Verzeichnis mit Web- und/oder Target-Visualisierungen
- Daten im Speicherbereich F-RAM

9.2 Betriebszustände AC4S

Inhalt	
Standard-SPS.....	107
Fehlersichere SPS.....	109
	20258

Dieser Abschnitt gibt Hinweise zu den Betriebsmodi und den Zuständen der Standard-SPS und der fehlersicheren SPS des Geräts sowie zu den Zuständen der Anwendungen.

9.2.1 Standard-SPS

7973

Betriebsmodus der Standard-SPS

13769

Die Standard-SPS des Geräts kann in folgenden Modi betrieben werden:

- **Offline-Modus**
Im Offline-Modus ist der Nutzer auf der Standard-SPS ausgeloggt oder es besteht keine Verbindung zwischen CODESYS und der Standard-SPS (z.B. Verbindungsabbruch).
- **Online-Modus**
Im Online-Modus ist der Nutzer auf der Standard-SPS eingeloggt.

Zustände der Standard-SPS-Anwendung

4136

Die auf dem Gerät gespeicherten Standard-Anwendungen werden in separaten Tasks unabhängig voneinander ausgeführt. Ein Standard-Anwendung kann folgende Zustände einnehmen:

- **Entladen**
Auf der Standard-SPS ist keine Anwendung gespeichert.
- **LÄUFT (RUN)**
Die Standard-Anwendung wird ausgeführt (zyklisch abgearbeitet).
- **STOP**
Die Standard-Anwendung wird nicht ausgeführt.

Zustand der Standard-Anwendung anzeigen

10272

Um den aktuellen Betriebszustand der Standard-SPS anzuzeigen, eine der folgenden Aktionen ausführen:

- **CODESYS:**
 - > Im Gerätebaum: Knoten der Standard-Anwendung zeigt Status:
ODER:
 - > Im Online-Modus zeigt die CODESYS-Statusleiste den aktuellen Zustand der Standard-Anwendung.
- **GUI / Web-Interface des Geräts:**
 - ▶  >
 - ▶ Registerkarte **Applikationen** wählen.
 - > Seite zeigt Zustände der auf dem Gerät gespeicherten Standard-SPS-Applikationen.

Zwischen den Zuständen wechseln

10264

Um zwischen den Zuständen der Standard-Anwendung umzuschalten, eine der folgenden Aktionen wählen:

Standard-SPS-Applikation starten

18027

Um eine auf dem Gerät gespeicherte Standard-SPS-Applikation zu starten:

- **CODESYS:**
 - ▶ Im Standardbereich des Gerätebaums: Gewünschte Standard-Anwendung als aktive Applikation markieren.
 - ▶ Mit [Online] > [Einloggen] die Verbindung zur CODESYS-Standard-SPS herstellen.
 - ▶ Mit [Debug] > [Start] die Abarbeitung der aktiven Applikation starten.
 - > Applikation geht in RUN-Zustand.
 - ▶ Optional: Vorgang für zusätzliche Applikationen wiederholen.
- **GUI / Web-Interface:**
 - ▶  > >
 - ▶ Registerkarte [Applikationen] wählen.
 - ▶ Mit [▲] / [▼] die gewünschte Applikation wählen.
 - > Seite zeigt Betriebszustand der gewählten Applikation.
 - ▶ Mit Funktionstaste [Starten] die Abarbeitung der gewählte Applikation starten.
 - > Applikation geht in RUN-Zustand.
 - ▶ Optional: Vorgang für zusätzliche Applikationen wiederholen.

Standard-SPS-Applikation stoppen

18029

Um eine auf dem Gerät gespeicherte Standard-SPS-Applikation zu stoppen:

- **CODESYS:**
 - ▶ Im Standardbereich des Gerätebaums: Gewünschte Standard-Anwendung als aktive Applikation markieren.
 - ▶ Mit [Online] > [Einloggen] die Verbindung zur CODESYS-Standard-SPS herstellen.
 - ▶ Mit [Online] > [Stop] die Abarbeitung der aktiven Applikation stoppen.
 - > Applikation geht in den STOP-Zustand.
 - ▶ Optional: Vorgang für additional Applikationen wiederholen.
- **GUI / Web-Interface:**
 - ▶  > >
 - ▶ Registerkarte [Applikationen] wählen.
 - ▶ Mit [▲] / [▼] die gewünschte Applikation wählen.
 - > Seite zeigt Betriebszustand der gewählten Applikation.
 - ▶ Mit Funktionstaste [Stoppen] die Abarbeitung der gewählte Applikation stoppen.
 - > Applikation geht in den STOP-Zustand.
 - ▶ Optional: Vorgang für weitere Applikationen wiederholen.

9.2.2 Fehlersichere SPS

8665



Machen Sie sich mit folgenden CODESYS-Funktionen vertraut!

- Einführung Safety-Online
→ Online-Hilfe > Add-Ons > CODESYS Safety > Online-Modus > Einführung: Safety Online
- Zustände der Sicherheitssteuerung
→ Online-Hilfe > Add-Ons > CODESYS Safety > Online-Modus > Zustände der Sicherheitssteuerung
- Debug-Modus
→ Online-Hilfe > Add-Ons > CODESYS Safety > Online-Modus > Debug-Modus

Für den ordnungsgemäßen Betrieb der fehlersicheren SPS des AC4S sind folgende Eigenschaften von Bedeutung:

- **Betriebsmodus der fehlersicheren SPS** (→ S. [109](#))
- **Zustände der fehlersicheren SPS** (→ S. [110](#))
- **Zustände der sicheren Anwendung** (→ S. [110](#))

Betriebsmodus der fehlersicheren SPS

3083

Die fehlersichere SPS des AC4S kann in folgenden Modi betrieben werden:

- **Offline-Modus**
Im Offline-Modus ist der Nutzer auf der fehlersicheren SPS ausgeloggt oder es besteht keine Verbindung zwischen CODESYS und der fehlersicheren SPS (z.B. Verbindungsabbruch).
- **Online-Modus**
Im Online-Modus ist der Nutzer auf der fehlersicheren SPS eingeloggt.

Zustände der fehlersicheren SPS

7365

Die fehlersichere SPS des AC4S kann folgende Zustände einnehmen:

- **Entladen**
Im Zustand "Entladen" ist keine sichere Anwendung auf dem AC4S gespeichert.
- **Unsicherer Betrieb (Debug-Modus)**
Im Zustand "Unsicherer Betrieb" ist eine Download-Applikation oder eine Bootapplikation auf dem AC4S gespeichert, die sich im STOP-Zustand befindet (→ **Zustände der sicheren Anwendung** (→ S. [110](#))).
Der Debug-Modus hat folgende Unterzustände:
 - Unsicher (DL)
Im Unterzustand "Unsicher (DL)" ist eine sichere Anwendung auf dem AC4S gespeichert.
 - Unsicher (BA)
Im Unterzustand "Unsicher (BA)" ist eine sichere Bootapplikation auf dem AC4S gespeichert.
- **Sicherer Betrieb**
Im Zustand "Sicherer Betrieb" ist eine sichere Bootapplikation auf dem AC4S gespeichert, die sich im RUN-Zustand befindet (→ **Zustände der sicheren Anwendung** (→ S. [110](#))).
Der Zustand "Sicherer Betrieb" hat folgende Unterzustände:
 - SICHER + eingeloggt
Im Unterzustand "SICHER + eingeloggt" befindet sich die fehlersichere SPS im Online-Modus (→ **Betriebsmodus der fehlersicheren SPS** (→ S. [109](#))).
 - Ausgeloggt
Im Unterzustand "Ausgeloggt" befindet sich die fehlersichere SPS im Offline-Modus (→ **Betriebsmodus der fehlersicheren SPS** (→ S. [109](#))).



Im Online-Modus wird der aktuelle Betriebszustand der fehlersicheren SPS in der CODESYS-Statusleiste angezeigt.

Zustände der sicheren Anwendung

18383

Die sichere Anwendung auf der fehlersicheren SPS des AC4S kann folgende Zustände einnehmen:

- **LÄUFT (RUN)**
Die sichere Anwendung wird ausgeführt (zyklische Abarbeitung).
- **STOP**
Die sichere Anwendung wird nicht ausgeführt.
- **BEENDET**
Die sichere Anwendung wurde aufgrund eines Laufzeitfehlers beendet.

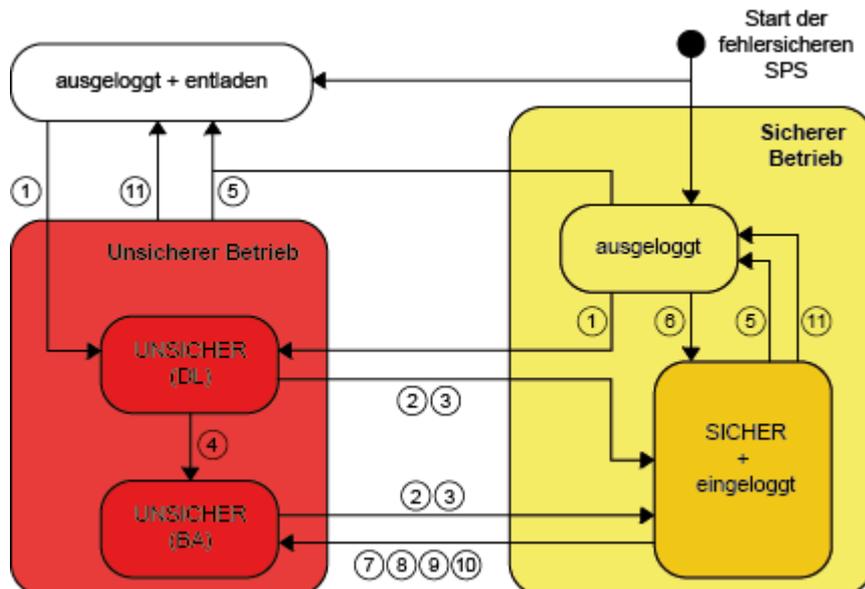


Im Online-Modus wird der aktuelle Zustand der sicheren Anwendung in der CODESYS-Statusleiste angezeigt.

Zwischen den Zuständen wechseln

7300

Folgendes Zustandsdiagramm zeigt die Zusammenhänge und möglichen Übergänge zwischen den einzelnen Betriebsmodi und Zuständen der fehlersicheren SPS:



- ① Einloggen mit Download ([Online] > [Einloggen])
- ② Bootapplikation starten ([Debug] > [Start])
- ③ Ausloggen mit Bootapplikation neu starten
- ④ Bootapplikation erzeugen ([Online] > [Bootapplikation erzeugen])
- ⑤ Ausloggen ([Online] > [Ausloggen])
- ⑥ Einloggen auf Bootapplikation ([Online] > [Einloggen])
- ⑦ Stop ([Debug] > [Stop])
- ⑧ Reset kalt ([Online] > [Reset])
- ⑨ Schreiben ([Debug] > [Werte schreiben])
- ⑩ Forcen ([Debug] > [Werte forcen])
- ⑪ Verbindungsabbruch zwischen CODESYS und AC4S (automatischer Übergang)

9.3 Reset

Inhalt	
Standard-SPS.....	112
Fehlersichere SPS.....	114
	18025

9.3.1 Standard-SPS

415



Ein Reset der Standard-SPS löst gleichzeitig einen Reset der fehlersicheren SPS des AC4S aus.

- Hinweise zum Anlaufverhalten der Steuerung beachten (→ **Anlaufverhalten der Steuerung**)!

Unterstützte Reset-Varianten

18613

Die folgende Tabelle zeigt die von der geräteinternen Standard-SPS unterstützten Reset-Varianten und das resultierende System-Verhalten:

Reset-Variante	System-Verhalten	Auslösende Aktionen
Reset (Warm)	<ul style="list-style-type: none"> Die Standard-Anwendung geht in den STOP-Zustand. Standard Variablen (VAR) der Standard-Anwendung werden neu initialisiert. Remanente Variablen (VAR RETAIN) der Standard-Anwendung behalten ihre aktuellen Werte. 	→ Standard-Anwendung rücksetzen (warm) (→ S. 113)
Reset (Kalt)	<ul style="list-style-type: none"> Die Standard-Anwendung geht in den Zustand STOP. Alle Variablen (VAR, VAR RETAIN) der Standard-Anwendung werden neu initialisiert. 	→ Standard-Anwendung rücksetzen (kalt) (→ S. 113)
Reset (Ursprung)	<ul style="list-style-type: none"> Die Standard-Anwendung geht in den STOP-Zustand. Die Standard-Anwendung auf der Standard-SPS wird gelöscht. Alle Variablen (VAR, VAR RETAIN) der Standard-Anwendung werden neu initialisiert. Standard-SPS wird in Urzustand rückgesetzt. 	→ Standard-Anwendung rücksetzen (Ursprung) (→ S. 113)



Eine Variable, die ohne einen Initialisierungswert deklariert wurde, wird mit dem variablenspezifischen Standardwert initialisiert (z.B. INT = 0).

Standard-Anwendung rücksetzen (warm)

9069

Um die Standard-Anwendung auf der Standard-SPS zurückzusetzen, eine der folgenden Aktionen ausführen:

- **CODESYS: Befehl [Reset warm]**
 - ▶ Im Standardbereich des Gerätebaums:
Gewünschte Standard-Anwendung als aktive Applikation markieren.
 - ▶ Mit [Online] > [Einloggen] die Verbindung zur CODESYS-Standard-SPS herstellen.
 - > CODESYS ist im Online-Modus.
 - ▶ Mit [Online] > [Reset warm] die Standard-Anwendung zurücksetzen.
- **GUI: Befehl [Zurücksetzen]**
 - ▶  >  wählen.
 - ▶ Registerkarte **Alle Applikationen** wählen.
 - ▶ Mit **Zurücksetzen** alle Standard-Anwendungen zurücksetzen.

Standard-Anwendung rücksetzen (kalt)

15687

Um die Standard-Anwendung auf der Standard-SPS zurückzusetzen, eine der folgenden Aktionen ausführen:

- **Standard-Anwendung auf das Gerät herunterladen**
 - ▶ → **Standard-Anwendung auf AC4S laden** (→ S. [103](#))
- **CODESYS: Befehl [Reset kalt]**
 - ▶ Im Standardbereich des Gerätebaums:
Gewünschte Standard-Anwendung als aktive Applikation markieren.
 - ▶ Mit [Online] > [Einloggen] die Verbindung zur CODESYS-Standard-SPS herstellen.
 - > CODESYS ist im Online-Modus.
 - ▶ Mit [Online] > [Reset kalt] die Standard-Anwendung zurücksetzen.

Standard-Anwendung rücksetzen (Ursprung)

18962

Um die Standard-Anwendung auf der Standard-SPS rückzusetzen:

- **CODESYS: Befehl [Reset Ursprung]**
 - ▶ Im Standardbereich des Gerätebaums:
Gewünschte Standard-Anwendung als aktive Applikation markieren.
 - ▶ Mit [Online] > [Einloggen] die Verbindung zur CODESYS-Standard-SPS herstellen.
 - > CODESYS ist im Online-Modus.
 - ▶ Mit [Online] > [Reset Ursprung] die Standard-Anwendung rücksetzen.

9.3.2 Fehlersichere SPS

17445

Unterstützte Reset-Varianten

12304

Die folgende Tabelle zeigt die von der geräteinternen fehlersicheren SPS unterstützten Reset-Varianten und das resultierende System-Verhalten:

Reset-Variante	System-Verhalten	Auslösende Aktionen
Reset (Kalt)	<ul style="list-style-type: none">▪ Sichere Anwendung geht in den Zustand STOP▪ Alle Variablen (VAR, VAR RETAIN) der sicheren Anwendung werden neu initialisiert.	→ Sichere Anwendung rücksetzen (kalt) (→ S. 114)

Sichere Anwendung rücksetzen (kalt)

12303

Um die sichere Anwendung auf der fehlersicheren SPS zurückzusetzen, eine der folgenden Aktionen ausführen:

- **Neue sichere Anwendung auf Gerät laden**
 - ▶ → **CODESYS-Projekt auf Gerät übertragen** (→ S. [102](#))
- **CODESYS: Befehl "Reset (kalt)" ausführen**
 - ▶ [SafetyApp] als aktive Anwendung setzen.
 - ▶ [Online] > [Einloggen] wählen.
 - > Fehlersichere SPS geht in den Online-Modus.
 - ▶ [Online] > [Reset kalt] wählen, um die sichere Anwendung auf dem Gerät rückzusetzen.

9.4 Web-Visualisierung anzeigen

17063

Um die Web-Visualisierung einer Standard-SPS-Anwendung oder einer ifm-Systemlösung anzuzeigen:

Voraussetzungen:

- > PC/Laptop ist mit Konfigurationsschnittstelle (X3) des Geräts verbunden (→ Gerätehandbuch: **Konfigurationsschnittstelle: Verbindungskonzepte**)

CODESYS-Standard-SPS-Anwendung

- ▶ Standard-SPS-Anwendung mit Web-Visualisierung auf das Gerät übertragen und starten (→ **Standard-Anwendung auf AC4S laden** (→ S. [103](#))).
- ▶ Auf PC/Laptop: Webbrowser starten.
- ▶ In Adresszeile folgendes eingeben und mit [ENTER] bestätigen:
<IP-Adresse-des-Geräts>:<8080>/myvisu.htm



myvisu ist der vom Programmierer gewählte Visualisierungsname (→ **Eigenschaften der Web-Visualisierung ändern** (→ S. [67](#))).

- > Webbrowser zeigt die Web-Visualisierung der Standard-SPS-Anwendung.

ifm-Systemlösung

- ▶ ifm-Systemlösung auf dem Gerät installieren und starten (→ Gerätehandbuch: **Single/Basis-App installieren** bzw. **Multi-App installieren**).
- ▶ Informationen über installierte ifm-Apps aufrufen (→ Gerätehandbuch: **Informationen über installierte ifm-Apps anzeigen**).
- ▶ Auf Hyperlink der ifm-App klicken.
- > Webbrowser zeigt die Web-Visualisierung der ifm-Systemlösung.

10 Anhang

Inhalt	
Bibliothek ACnnnn_Utils.library	117
Bibliothek ACnnnn_SYS_CMD.library	182
Bibliothek SF_IO.library	190
Bibliothek SF_LogicalInterfaces.library	206
Fehler-Codes: Sichere AS-i Slaves	275
Fehler-Codes: Lokale E/A-Schnittstelle	277

7156

10.1 Bibliothek ACnnnn_Utills.library

Inhalt	
Übersicht: AS-i Funktionen (FB_ASi)	118
Übersicht: System-Funktionen (FB_System)	166
Aufzählungstypen und komplexe Variablen	173

17722

10.1.1 Übersicht: AS-i Funktionen (FB_ASi)

Inhalt	
Get_ASi_PHY_Dat	119
Get_LAS	122
Get_LDS	124
Get_LPS	126
Get_LCE	128
Get_LCEAS	130
Get_LCEMS	132
Get_LDAE	134
Get_LPF	136
Get_CDI	138
Get_PCD	140
Get_InputParameter	142
Get_OutputParameter	144
Set_AddressMode	146
Set_ASi_Config	148
Set_LPS	150
Set_Mode	152
Set_PCD	154
Set_ProjectAll	156
Set_SlaveAddress	158
Set_SlaveExtendedID1	160
Set_SlaveParameter	162
Get_ASi_Data	164

17459

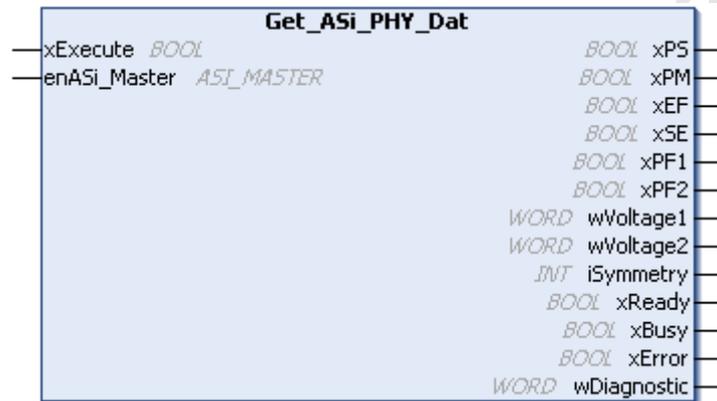
Get_ASi_PHY_Dat

16005

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16040

Der FB liest die physikalische Daten des gewählten AS-i Masters aus und gibt die Werte aus.

Eingangsparameter

16041

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16042

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xPS	BOOL	Spannungsquelle (Power Source)	FALSE	Gerät wird durch Aux versorgt.
			TRUE	Gerät wird durch AS-i versorgt.
xPM	BOOL	Power24-Modul (PM)	FALSE	Power24-Modul fehlt.
			TRUE	Power24-Modul ist gesteckt.
xEF	BOOL	Erdschlussfehler (Earth Fault)	FALSE	kein Erdschluss
			TRUE	Versorgungsspannung ist unsymmetrisch, Verdacht auf Erdschluss.
xSE	BOOL	Status der Erdschlussfehler-Erkennung (Status Earth Fault Detection)	FALSE	Erdschlussfehler-Erkennung liefert keine gültigen Daten (z.B. wenn AS-i Spannung fehlt).
			TRUE	Erdschlussfehler-Erkennung liefert gültige Daten.
xPF1	BOOL	Spannung <22,5 V (Power Fail 22,5 V)	FALSE	kein AS-i Power Fail (Classic APF)
			TRUE	AS-i Power Fail (Classic APF), d.h. AS-i Spannung liegt unterhalb 22,5 V
xPF2	BOOL	Spannung <19,0V (Power Fail 19V)	FALSE	kein AS-i Power Fail (24V-APF)
			TRUE	AS-i Power Fail (24V-APF), d.h. AS-i Spannung liegt unterhalb 19,0 V
wVoltage1	WORD	Spannung AS-i+ nach AS-i- in mV		
wVoltage2	WORD	Spannung FE nach AS-i- in mV		
iSymmetry	INT	Symmetrie in % (-100% ... +100%)	0xFF9C ... 0x0000 ... 0x0064	-100% ... 0% ... +100%
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt



Get_LAS

16008

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16068

Der FB liest die Liste der aktivierten Slaves (List of Active Slaves = LAS) des gewählten AS-i Masters aus und gibt die Werte aus.

Eingangsparameter

16069

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16070

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLAS_SA_Slaves	DWORD	Liste der aktiven S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Single/A-Slave vorhanden
			1	Single/A-Slave vorhanden
dwLAS_B_Slaves	DWORD	Liste der aktiven B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein B-Slave vorhanden
			1	B-Slave vorhanden
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LDS

16013

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16118

Der FB liest die Liste der detektierten Slaves (List of detected Slaves = LDS) des gewählten AS-i Masters aus und gibt die Werte aus.

Eingangsparameter

16119

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16120

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLDS_SA_Slaves	DWORD	Liste der detektierten S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Slave erkannt
			1	Slave erkannt
dwLDS_B_Slaves	DWORD	Liste der detektierten B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein Slave erkannt
			1	Slave erkannt
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LPS

16015

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16130

Der FB liest die Liste der projizierten Slaves (List of Projected Slaves = LPS) am gewählten AS-i Master aus und gibt die Werte aus.

Eingangsparameter

16131

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16132

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLPS_SA_Slaves	DWORD	Liste der projektierten S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	Slave nicht projektiert
			1	Slave projektiert
dwLPS_B_Slaves	DWORD	Liste der projektierten B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	Slave nicht projektiert
			1	Slave projektiert
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LCE

16009

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16075

Der FB liest die Liste der Konfigurationsfehler (List of Configuration Errors = LCE) des gewählten AS-i Masters aus und gibt die Werte aus.

Eingangsparameter

16076

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16077

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLCE_SA_Slaves	DWORD	Liste der Konfigurationsfehler der S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Konfigurationsfehler
			1	Konfigurationsfehler
dwLCE_B_Slaves	DWORD	Liste der Konfigurationsfehler der B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein Konfigurationsfehler
			1	Konfigurationsfehler
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LCEAS

16010

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16098

Der FB liest die Liste der vorhandenen, aber nicht projektierten Slaves (List of Configuration Error – Additional Slave = LCEAS) des gewählten AS-i Masters aus und gibt die Werte aus.

Eingangsparameter

16099

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16100

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLCEAS_SA_Slaves	DWORD	Liste der Konfigurationsfehler der S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Konfigurationsfehler - Zusätzlicher Slave
dwLCEAS_B_Slaves	DWORD	Liste der Konfigurationsfehler der B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein Konfigurationsfehler - Zusätzlicher Slave
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LCEMS

16011

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16106

Der FB liest die Liste der projektierten, aber nicht vorhandenen Slaves (List of Configuration Error – Missing Slave = LCEMS) am gewählten AS-i Master aus und gibt die Werte aus.

Eingangsparameter

16107

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16108

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLCEMS_SA_Slaves	DWORD	Liste der konfigurierten, aber nicht vorhandenen S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Konfigurationsfehler - Fehlender Slave
			1	Slave ist projektiert, aber nicht vorhanden
sdwLCEMS_B_Slaves	DWORD	Liste der konfigurierten, aber nicht vorhandenen B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein Konfigurationsfehler - Fehlender Slave
			1	Slave ist projektiert, aber nicht vorhanden
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LDAE

16012

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16112

Der FB liest die Doppeladressfehler (List of Duplicate Address Error = LDAE) des gewählten AS-i Masters aus und gibt die Werte in einer Liste aus.

Eingangsparameter

16113

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16114

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLDAE_SA_Slaves	DWORD	Liste der Doppeladressfehler. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Doppeladressfehler
			1	Doppeladressfehler
dwLDAE_B_Slaves	DWORD	Liste der Doppeladressfehler. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein Doppeladressfehler
			1	Doppeladressfehler
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_LPF

16014

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16124

Der FB liest die Liste der Peripheriefehler (List of Peripheral Faults = LPF) des gewählten AS-i Masters aus und gibt die Werte aus.

Eingangsparameter

16125

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16126

Parameter	Datentyp	Bedeutung	Mögliche Werte	
dwLPF_SA_Slaves	DWORD	Liste der Peripheriefehler an S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	kein Peripheriefehler
			1	Peripheriefehler erkannt
dwLPF_B_Slaves	DWORD	Liste der Peripheriefehler an B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	kein Peripheriefehler
			1	Peripheriefehler erkannt
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Get_CDI

16006

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utills.library

Symbol in CODESYS:



Beschreibung

16045

Der FB liest die Konfigurationsdaten (Configuration Data Image = CDI) der Slaves am gewählten AS-i Master aus und gibt die Werte in einem Array aus. Die Konfigurationsdaten eines Slaves besteht aus den Registern IO, ID, ID1 und ID2.

Eingangsparameter

16046

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16047

Parameter	Datentyp	Bedeutung	Mögliche Werte	
awCDI	ARRAY [0...63] OF WORD	Konfigurationsdaten der Slaves am gewählten AS-i Master	pro Word: Bits 0...3: I/O-Code Bits 4...7: ID-Code Bits 8-11: ID1-Code Bits 12-15: ID2-Code	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

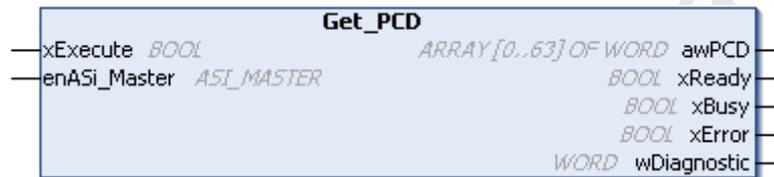
Get_PCD

16017

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16141

Der FB liest die projektorientierten Konfigurationsdaten (Projected Configuration Data = PCD) der Slaves am gewählten AS-i Master und gibt die Werte in einem Array aus.

Eingangsparameter

16142

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16143

Parameter	Datentyp	Bedeutung	Mögliche Werte	
awPCD	ARRAY [0...63] OF WORD	Permanente Konfigurationsdaten der Slaves am gewählten AS-i Master	pro Word: Bits 0...3: I/O-Code Bits 4...7: ID-Code Bits 8...11: ID1-Code Bits 12...15: ID2-Code ⓘ Daten in Word 0 sind ungültig!	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

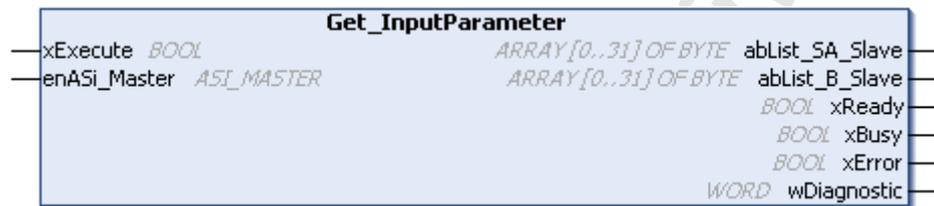
Get_InputParameter

16007

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16056

Der FB liest die Input-Parameter der Slaves am gewählten AS-i Masters aus und gibt die Werte in 2 Arrays für Single-A-Slaves und B-Slaves aus.

Eingangsparameter

16057

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16058

Parameter	Datentyp	Bedeutung	Mögliche Werte	
abList_SA_Slave	ARRAY[0..31] OF BYTE	Liste der Ausgangsparameter der S/A-Slaves am gewählten AS-i Master. Jedes Byte enthält die Ausgangsparameter eines AS-i Slaves. – Byte 0 (LSB) = res. – Byte 1 = Slave mit Adresse 1(A) ... – Byte 31 = Slave mit Adresse 31(A)	Pro Byte:	
			Bits 0...3:	P0-P3
abList_B_Slave	ARRAY[0..31] OF BYTE	Liste der Ausgangsparameter der B-Slaves am gewählten AS-i Master. Jedes Byte enthält die Ausgangsparameter eines AS-i Slaves. – Byte 0 (LSB) = res. – Byte 1 = Slave mit Adresse 1B ... – Byte 31 = Slave mit Adresse 31B	Pro Byte:	
			Bits 0...3:	P0-P3
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

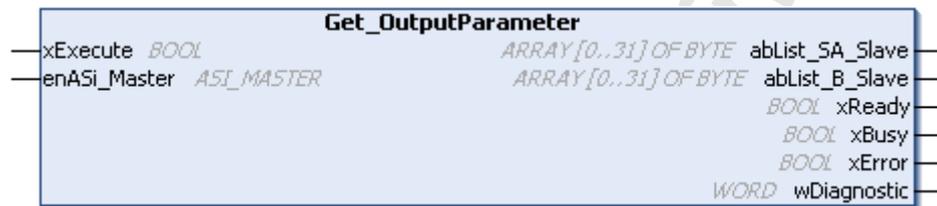
Get_OutputParameter

16016

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16136

Der FB liest die Output-Parameter der Slaves am gewählten AS-i Master und gibt die Werte in für S/A-Slaves und B-Slaves getrennt in 2 Arrays aus.

Eingangsparameter

16137

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

16138

Parameter	Datentyp	Bedeutung	Mögliche Werte	
abList_SA_Slave	ARRAY[0..31] OF BYTE	Liste der Ausgangsparameter der S/A-Slaves am gewählten AS-i Master. Jedes Byte enthält die Ausgangsparameter eines AS-i Slaves. – Byte 0 (LSB) = res. – Byte 1 = Slave mit Adresse 1(A) ... – Byte 31 = Slave mit Adresse 31(A)	Pro Byte:	
			Bits 0...3:	P0-P3
abList_B_Slave	ARRAY[0..31] OF BYTE	Liste der Ausgangsparameter der B-Slaves am gewählten AS-i Master. Jedes Byte enthält die Ausgangsparameter eines AS-i Slaves. – Byte 0 (LSB) = res. – Byte 1 = Slave mit Adresse 1B ... – Byte 31 = Slave mit Adresse 31B	Pro Byte:	
			Bits 0...3:	P0-P3
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Set_AddressMode

16018

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16146

Der FB aktiviert/deaktiviert den Parameter "Automatische Adressierung" für den gewählten AS-i Master.

Eingangsparameter

16147

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
enAuto_Address_Mode	ASI_ADDRESS_MODE	Parameter aktiviert/deaktiviert den automatischen Adressierungsmodus.	Auto_address_disable	Automatische Adressierung inaktiv
			Auto_address_enable	Automatische Adressierung aktiv

Ausgangsparameter

17017

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt

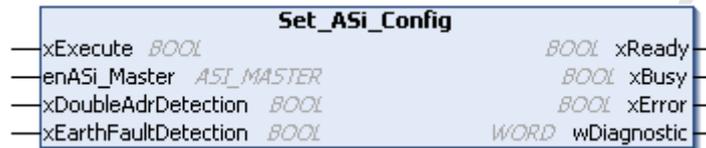
Set_ASi_Config

16019

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16149

Der FB aktiviert/deaktiviert die Parameter "Doppeladresserkennung" und "Erdschlusserkennung" für den gewählten AS-i Master.

Eingangsparameter

16150

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
xDoubleAdrDetection	BOOL	Parameter "Doppeladresserkennung" aktivieren/deaktivieren	FALSE	Doppeladresserkennung inaktiv
			TRUE	Doppeladresserkennung aktiv
xEarthFaultDetection	BOOL	Parameter "Erdschlusserkennung" aktivieren/deaktivieren	FALSE	Erdschlusserkennung inaktiv
			TRUE	Erdschlusserkennung aktiv

Ausgangsparameter

17015

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0001 Falscher Parameter übergeben, Einstellung wurde nicht übernommen.

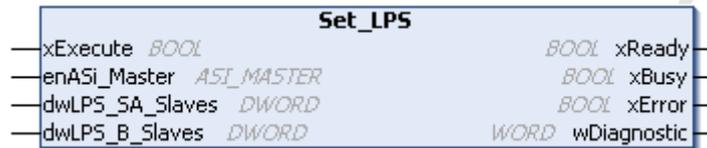
Set_LPS

16020

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16153

Der FB ändert die Liste der projizierten Slave (List of Projected Slaves = LPS) im gewählten AS-i Master.

Eingangsparameter

16154

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enAsi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
dwLPS_SA_Slaves	DWORD	Liste der projizierten S/A-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = Adresse 0 ... – Bit 31 (MSB) = Adresse 31/31A	Pro Bit:	
			0	Slave nicht projiziert
			1	Slave projiziert
dwLPS_B_Slaves	DWORD	Liste der projizierten B-Slaves. Jedes Bit repräsentiert eine AS-i Adresse: – Bit 0 (LSB) = ungenutzt – Bit 1 = Adresse 1B ... – Bit 31 (MSB) = Adresse 31B	Pro Bit:	
			0	Slave nicht projiziert
			1	Slave projiziert

Ausgangsparameter

17016

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0019 Master nicht im Projektierungsmodus

Set_Mode

16021

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16155

Der FB ändert die Betriebsart (Geschützter Betrieb, Projektierungsmodus) des gewählten AS-i Masters.

Eingangsparameter

16156

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
enMode_ASi_Master	ASI_MASTER_MODE	Betriebsart des AS-i Masters	Closed_mode	Geschützter Modus aktiv
			Project_mode	Projektierungsmodus aktiv

Ausgangsparameter

17018

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0003 Slave mit Adresse 0 gefunden (Slave not detected)

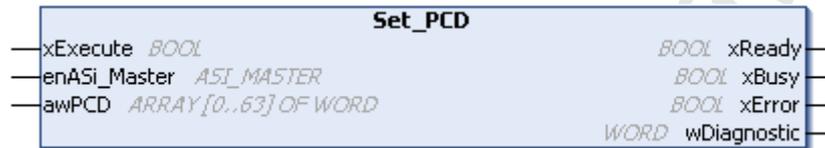
Set_PCD

16022

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16151

Der FB ändert die Konfigurationsdaten (Permanent Configuration Data = PCD) der Slaves am gewählten AS-i Master.

Eingangsparameter

16160

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
awPCD	ARRAY [0...63] OF WORD	Permanente Konfigurationsdaten der Slaves am gewählten AS-i Master	pro Word: Bits 0...3: I/O-Code Bits 4...7: ID-Code Bits 8...11: ID1-Code Bits 12...15: ID2-Code ! Daten in Word 0 sind ungültig!	

Ausgangsparameter

15574

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0019 Master nicht im Projektierungsmodus

Set_ProjectAll

16023

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16161

Der FB startet den Projektierungsabgleich auf dem gewählten AS-i Master.

Eingangsparameter

16125

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2

Ausgangsparameter

17020

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0003 Slave mit Adresse 0 gefunden (Slave not detected)
- 0x0019 Master nicht im Projektierungsmodus

Set_SlaveAddress

16024

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16162

Der FB ändert die Adresse des gewählten AS-i Slaves.

Eingangsparameter

16165

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enAsi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
enAsi_Slave	ASI_SLAVE	Adresse des AS-i Slaves	Slave_n	AS-i Slave an Adresse n (n = 1 ... 31)
enAsi_SlaveTyp	ASI_SLAVE_TYP	Typ des AS-i Slaves	SA_Slave	Single oder A-Slave
			B_Slave	B-Slave
enAsi_Slave_new	ASI_SLAVE	Neue Adresse des AS-i Slaves	Slave_n	AS-i Slave an Adresse n (n = 1 ... 31)
enAsi_SlaveTyp_new	ASI_SLAVE_TYP	Neuer Typ des AS-i Slaves	SA_Slave	Single Slave oder A-Slave
			B_Slave	B-Slave

Ausgangsparameter

17021

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0001 Slave antwortet nicht oder Wechsel in Offline-Modus während FB-Ausführung
- 0x0002 Slave mit alter Adresse nicht gefunden (Slave not detected)
- 0x0003 Slave mit Adresse 0 gefunden (Slave not detected)
- 0x0004 Slave mit neuer Adresse gefunden
- 0x0005 Fehler beim Löschen der alten Adresse (Delete Error)
- 0x0006 ExtendedID1 konnten nach dem Schreiben nicht ausgelesen werden (Read Error)
- 0x0007 Fehler beim Schreiben des ExtendedID1 (Set Error)
- 0x0008 Neue Adresse temporär gespeichert
- 0x0009 ExtendedID1 temporär gespeichert
- 0x0018 Master nicht im Normalbetrieb.

Set_SlaveExtendedID1

16025

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16169

Der FB ändert die Extended ID1 des gewählten AS-i Slaves.

Eingangsparameter

16170

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
enASi_Slave	ASI_SLAVE	Adresse des AS-i Slaves	Slave_n	AS-i Slave an Adresse n (n = 1 ... 31)
enASi_SlaveTyp	ASI_SLAVE_TYP	Typ des AS-i Slaves	SA_Slave	Single oder A-Slave
			B_Slave	B-Slave
bExtendedID1	BYTE	Erweiterter ID1-Code des gewählten AS-i Slaves	Erweiterter ID1-Code (hexadezimaler Darstellung)	

Ausgangsparameter

16148

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0003 Slave mit Adresse 0 gefunden (Slave not detected)
- 0x0005 Fehler beim Löschen der alten Adresse (Delete Error)
- 0x0006 ExtendedID1 konnten nach dem Schreiben nicht ausgelesen werden (Read Error)
- 0x0007 Fehler beim Schreiben des ExtendedID1 (Set Error)
- 0x0009 ExtendedID1 temporär gespeichert
- 0x000E Ungültige Slave-Adresse (z.B. 0 oder 0B angegeben)
- 0x0018 Master nicht im Normalbetrieb.
- 0x0021 Ungültiger ExtendedID1-Code

Set_SlaveParameter

16026

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

16172

Der FB ändert die Parameter des gewählten AS-i Slaves.

Eingangsparameter

16173

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
enASi_Slave	ASI_SLAVE	Adresse des AS-i Slaves	Slave_n	AS-i Slave an Adresse n (n = 1 ... 31)
enASi_SlaveTyp	ASI_SLAVE_TYP	Typ des AS-i Slaves	SA_Slave	Single oder A-Slave
			B_Slave	B-Slave
enASi_SlaveParam	BYTE	Parameter des gewählten AS-i Slaves	Slave-Parameter (hexadezimale Darstellung)	

Ausgangsparameter

17025

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0001 Slave antwortet nicht oder Wechsel in Offline-Modus während FB-Ausführung
- 0x000A Slave nicht in LAS
- 0x000B Dateninhalt ungültig (z.B. Parameterwert >7 bei A/B-Slaves)
- 0x000E Ungültige Slave-Adresse (z.B. 0 oder 0B angegeben)
- 0x0018 Master nicht im Normalbetrieb.

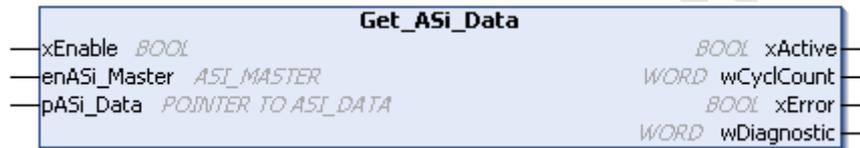
Get_ASi_Data

16000

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnnnn_Utils.library

Symbol in CODESYS:



Beschreibung

16033

Der FB liest folgende Daten aus dem gewählten AS-i Masters aus und speichert die Werte in einer Strukturvariable vom Datentyp ASI_DATA (→ **ASI_DATA (STRUCT)** (→ S. 175)):

- Liste der aktivierten Slaves (List of activated Slaves – LAS)
- Liste der erkannten Slaves (List of detected Slaves – LDS)
- Liste der projctierten Slaves (List of projected Slaves – LPS)
- Liste der Konfigurationsfehler (List of Configuration Errors – LCE)
- Liste der Konfigurationsfehler - zusätzlicher Slave (List of Configuration Errors Additional Slave – LCEAS)
- Liste der Konfigurationsfehler - fehlender Slave (List of Configuration Error Missing Slave – LCEMS)
- Liste der Peripheriefehler (List of Peripheral Faults – LPF)
- Liste der Doppeladressfehler (List of Double Adress Errors – LDAE)
- Konfigurationsdatenabbild (Configuration Data Image – CDI)
- Projektierte Konfigurationsdaten (Projected Configuration Data – PCD)
- Eingangsparameter der AS-i Slaves
- Ausgangsparameter der AS-i Slaves

Eingangsparameter

16034

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xEnable	BOOL	Aktivität des FB steuern	TRUE	FB ist aktiviert
			FALSE	FB ist deaktiviert
enASi_Master	ASI_MASTER	AS-i Master wählen	Master_1	AS-i Master 1
			Master_2	AS-i Master 2
pASi_Data	POINTER TO ASI_DATA	Strukturvariable, in der die ausgelesenen Daten gespeichert werden sollen.	! Variable muss deklariert sein!	

Ausgangsparameter

16035

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xActive	BOOL	Signal bestätigt die Ausführung des FB	FALSE	FB ist inaktiv
			TRUE	FB ist aktiv (=wird ausgeführt)
wCyclCount	WORD	Zähler für vollständig durchlaufene FB-Zyklen	Zahl in Hexadezimaldarstellung	
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0xnF01 Unbekannter Fehler *
- 0xnF02 Unbekanntes/Ungültiges Ziel *
- 0xnF03 Unbekannte Kommando-ID *
- 0xnF04 Ungültige Parameter *
- 0xnF05 Zeitüberschreitung bei Abarbeitung (Timeout) *

Legende:

* ... Get_ASi_Data führt den FB ACnxxx_SysCmd sequentiell mit verschiedenen Kommando-IDs aus, um die einzelnen Elemente der komplexen Variable zu ermitteln. Im zurückgegebenen Fehlercode kennzeichnet das Nibble n den Kommando-Aufruf, bei dem der Fehler aufgetreten ist. n kann folgende Werte besitzen:

- 1 = Fehler bei "Get LAS, LDA, LPF, LCE"
- 2 = Fehler bei "Get LPS"
- 3 = Fehler bei "Get CDI"
- 4 = Fehler bei "Get PCD"
- 5 = Fehler bei "Get Input Parameter"
- 6 = Fehler bei "Get Output Parameter"
- 7 = Fehler bei "Get LCEMS, LCEAS, LDAE"

10.1.2 Übersicht: System-Funktionen (FB_System)

Inhalt	
Get_FieldbusInfo	167
QuickSetupASi_Master	169
Set_TimeDate.....	171

17460

Get_FieldbusInfo

17453

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnrrnn_Utills.library

Symbol in CODESYS:



Beschreibung

17454

Der FB liest Informationen über den Feldbus und gibt die Werte in einer Liste aus. Folgende Informationen werden ausgelesen:

- Zustand des Feldbusses
- Feldbus-Typ
- Zustand der Ethernet-Verbindung an Port X6 und X7
- MAC-Adressen
- IP-Adresse des Feldbus-Hosts

Eingangsparameter

17455

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten

Ausgangsparameter

17456

Parameter	Datentyp	Bedeutung	Mögliche Werte	
aw_InfoList	ARRAY[0...18] OF WORD	Feldbus-Informationen	Lage der Daten innerhalb der einzelnen Worte → DS18: Feldbus-Informationen	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

QuickSetupAsi_Master

15999

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

15902

Der FB führt die Quick-Setup-Routine auf den gewählten AS-i Mastern aus.

Eingangsparameter

15903

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
xMaster1	BOOL	AS-i Master 1 für Quick-Setup wählen	FALSE	Kein Quick-Setup ausführen; AS-i Projektierung bleibt unverändert.
			TRUE	Quick-Setup auf AS-i Master ausführen
xMaster2	BOOL	AS-i Master 2 für Quick-Setup wählen	FALSE	Kein Quick-Setup ausführen; AS-i Projektierung bleibt unverändert.
			TRUE	Quick-Setup auf AS-i Master ausführen

Ausgangsparameter

15920

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0003 Slave mit Adresse 0 gefunden (Slave not detected)

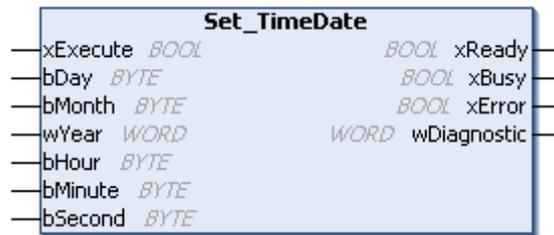
Set_TimeDate

15998

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: ACnxxx_Utils.library

Symbol in CODESYS:



Beschreibung

15949

Der FB stellt mit den übergebenen Eingangswerten die Systemzeit (Uhrzeit, Datum) des Geräts ein.

Eingangsparameter

15988

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
bDay	BYTE	Tag	0x01 ... 0x1F	1 ... 31
bMonth	BYTE	Monat	0x01 ... 0x0C	Januar ... Dezember
wYear	WORD	Jahr	0x07B3 ... 0x07F5	1971 ... 2037
bHour	BYTE	Stunde	0x00 ... 0x17	0 ... 23
bMinute	BYTE	Minute	0x00 ... 0x3B	0 ... 59
bSecond	BYTE	Sekunde	0x00 ... 0x3B	0 ... 59

Ausgangsparameter

15990

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xBusy	BOOL	Signal zeigt an, ob der FB ausgeführt wird.	FALSE	FB ist deaktiviert oder FB-Ausführung abgeschlossen.
			TRUE	Ausführung des FB gestartet, aber noch nicht beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
wDiagnostic	WORD	Diagnoseinformationen	→ Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 kein spezifischer Fehler gesetzt
- 0x0001 Übergebene Werte für Datum/Zeit sind ungültig und konnten nicht gesetzt werden.
- 0x0002 NTP aktiv, Uhrzeit konnte nicht übernommen werden.

10.1.3 Aufzählungstypen und komplexe Variablen

Inhalt	
Aufzählungstypen (ENUM)	173
Komplexe Variablen (STRUCT)	175

15986

Zusätzlich zu den Standard-Datentypen bietet das CODESYS-Package der ifm electronic folgende Aufzählungstypen (ENUM) und komplexe Variablen (STRUCT):

Aufzählungstypen (ENUM)

16176

Die Bibliothek ACnnnn_Utils stellt folgende Aufzählungstypen (ENUM) bereit:

ASI_ADDRESS_MODE (ENUM)

16177

Bezeichnung	Bedeutung	Variable	Datentyp	Wert
ASI_ADDRESS_MODE	AS-i Autoadressierungsmodus	▪ Auto_address_enable	INT	0
		▪ Auto_address_disable	INT	1

ASI_MASTER (ENUM)

16178

Bezeichnung	Bedeutung	Variable	Datentyp	Wert
ASI_MASTER	Identifizier für AS-i Master	▪ Master_1	INT	1
		▪ Master_2	INT	2

ASI_MASTER_MODE (ENUM)

16179

Bezeichnung	Bedeutung	Variable	Datentyp	Wert
ASI_MASTER_MODE	Betriebsmodus des AS-i Masters	▪ Closed_mode	INT	0
		▪ Project_mode	INT	1

ASI_SLAVE (ENUM)

16180

Bezeichnung	Bedeutung	Variable	Datentyp	Wert
ASI_SLAVE	Identifizier für AS-i Slaves	▪ Slave_1	INT	1
		▪ Slave_2	INT	2
		▪ Slave_3	INT	3
		▪ Slave_4	INT	4
		▪ Slave_5	INT	5
		▪ Slave_6	INT	6
		▪ Slave_7	INT	7
		▪ Slave_8	INT	8
		▪ Slave_9	INT	9
		▪ Slave_10	INT	10
		▪ Slave_11	INT	11
		▪ Slave_12	INT	12
		▪ Slave_13	INT	13
		▪ Slave_14	INT	14
		▪ Slave_15	INT	15
		▪ Slave_16	INT	16
		▪ Slave_17	INT	17
		▪ Slave_18	INT	18
		▪ Slave_19	INT	19
		▪ Slave_20	INT	20
		▪ Slave_21	INT	21
		▪ Slave_22	INT	22
		▪ Slave_23	INT	23
		▪ Slave_24	INT	24
		▪ Slave_25	INT	25
		▪ Slave_26	INT	26
		▪ Slave_27	INT	27
		▪ Slave_28	INT	28
		▪ Slave_29	INT	29
		▪ Slave_30	INT	30
		▪ Slave_31	INT	31

ASI_SLAVE_TYP (ENUM)

16181

Bezeichnung	Bedeutung	Variable	Datentyp	Wert
ASI_SLAVE_TYP	Typ des AS-i Slaves	▪ SA_Slave	INT	0
		▪ B_Slave	INT	1

Komplexe Variablen (STRUCT)

15991

Die Bibliothek ACnnnn_Utils.library stellt komplexe Variablen vom Datentyp STRUCT bereit. Sie werden von den FBs genutzt, können aber auch vom Programmierer in CODESYS-Projekten für die geräteinterne Standard-SPS eingesetzt werden.

ASI_DATA (STRUCT)

15992



Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ LDS_SA_Slave	DWORD	Liste der aktiven S/A-Slaves	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein Slave aktiv 1 = Slave aktiv → DS9 – Slave-Listen LAS, LDS, LPF, LCE , Worte 4...7
▪ LDS_B_Slave	DWORD	Liste der aktiven B-Slaves	
▪ LAS_SA_Slave	DWORD	Liste aktiver S/A-Slaves	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein aktiver Slave 1 = aktiver Slave → DS9 – Slave-Listen LAS, LDS, LPF, LCE , Worte 0...3
▪ LAS_B_Slave	DWORD	Liste aktiver B-Slaves	
▪ LPF_SA_Slave	DWORD	Liste der Peripheriefehler (S/A-Slaves)	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein Peripheriefehler 1 = Peripheriefehler → DS9 – Slave-Listen LAS, LDS, LPF, LCE , Worte 8...11
▪ LPF_B_Slave	DWORD	Liste der Peripheriefehler (B-Slaves)	
▪ LCE_SA_Slave	DWORD	Liste mit Konfigurationsfehler (S/A-Slaves)	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein Konfigurationsfehler 1 = Konfigurationsfehler → DS9 – Slave-Listen LAS, LDS, LPF, LCE , Worte 12...15
▪ LCE_B_Slave	DWORD	Liste mit Konfigurationsfehler (B-Slaves)	
▪ LPS_SA_Slave	DWORD	Liste der projektierten S/A-Slaves	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein projektiertes Slave 1 = projektiertes Slave → DS10 – Slave-Liste LPS
▪ LPS_B_Slave	DWORD	Liste der B-Slaves	
▪ LCEMS_SA_Slave	DWORD	Liste der Konfigurationsfehler: projektiertes, aber fehlendes S/A-Slaves	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein Konfigurationsfehler 1 = Konfigurationsfehler → DS17 – AS-i Master: Fehlerlisten LCEMS, LCEAS, LDAE , Worte 0...3
▪ LCEMS_B_Slave	DWORD	Liste der Konfigurationsfehler: projektiertes, aber fehlendes B-Slaves	
▪ LCEAS_SA_Slave	DWORD	Liste der Konfigurationsfehler: zusätzlicher S/A-Slaves	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein Fehler 1 = Fehler → DS17 – AS-i Master: Fehlerlisten LCEMS, LCEAS, LDAE , Worte 4...7
▪ LCEAS_B_Slave	DWORD	Liste der Konfigurationsfehler: zusätzlicher B-Slaves	
▪ LDAE_SA_Slave	DWORD	Liste der Doppeladressfehler (S/A-Slaves)	Jedes Bit repräsentiert eine AS-i Adresse: 0 = kein Doppeladressfehler 1 = Doppeladressfehler → DS17 – AS-i Master: Fehlerlisten LCEMS, LCEAS, LDAE , Worte 8...11
▪ LDAE_B_Slave	DWORD	Liste der Doppeladressfehler (B-Slaves)	
▪ CDI	ARRAY[0..63] OF WORD	Abbild der aktuellen Konfigurationsdaten (CDI = Configuration Data Image)	Pro Slave steht ein Wort zur Verfügung: Bits 0...3 = IO-Code Bits 4...7 = ID-Code Bits 8...11 = Extended ID1-Code Bits 12...15 = Extended ID2-Code → DS11 – Aktuelle Konfigurationsdaten (CDI)

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ PCD	ARRAY[0..63] OF WORD	Abbild der projektierten Konfigurationsdaten (PCD = Projected Configuration Data)	Jedes Wort enthält Daten eines Slave: Bits 0...3 = IO Bits 4...7 = ID Bits 8...11 = ID1 Bits 12...15 = ID2 → DS12 – Projektierte Konfigurationsdaten (PCD)
▪ InputParam_SA_Slave	ARRAY[0..31] OF BYTE	Eingangsparameter der S/A-Slaves	Jedes Byte enthält Parameter eines S/A-Slaves: Bits 0...3 = P0-P3 Bits 4...7 = reserviert → DS13 – Abbild der Eingangsparameter der Slaves (PI) , Worte 0...15
▪ InputParam_B_Slave	ARRAY[0..31] OF BYTE	Eingangsparameter der B-Slaves	Jedes Byte enthält Parameter eines B-Slaves: Bits 0...3 = P0-P3 Bits 4...7 = reserviert → DS13 – Abbild der Eingangsparameter der Slaves (PI) , Worte 16...31
▪ OutputParam_SA_Slave	ARRAY[0..31] OF BYTE	Ausgangsparameter der S/A-Slaves	Jedes Byte enthält Parameter eines S/A-Slaves: Bits 0...3 = P0-P3 Bits 4...7 = reserviert → DS14 – Abbild der Ausgangsparameter der Slaves (PP) , Worte 0...15
▪ OutputParam_B_Slave	ARRAY[0..31] OF BYTE	Ausgangsparameter der B-Slaves	Jedes Byte enthält Parameter eines B-Slaves: Bits 0...3 = P0-P3 Bits 4...7 = reserviert → DS14 – Abbild der Ausgangsparameter der Slaves (PP) , Worte 16...31

ASI_NET (STRUCT)

15993

Die Struktur enthält das komplette Prozessabbild (Ein- und Ausgänge) eines AS-i Netzwerks.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ binIO	ASI_BIN_IO	Binäre Ein- und Ausgangsdaten	→ ASI_BIN_IO (STRUCT) (→ S. 178)
▪ analO	ASI_ANA_IO	Analoge Ein- und Ausgangsdaten	→ ASI_ANA_IO (STRUCT) (→ S. 179)

ASI_BIN_IO (STRUCT)

16002

Die Struktur enthält die Prozessdaten der digitalen Ein- und Ausgangs-Slaves:

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ bin_IN_Slaves	ASI_BIN_IN	Eingangsdaten der digitalen AS-i Slaves	→ ASI_BIN_IN (STRUCT) (→ S. 178)
▪ bin_OUT_Slaves	ASI_BIN_OUT	Ausgangsdaten der digitalen AS-i Slaves	→ ASI_BIN_OUT (STRUCT) (→ S. 178)

ASI_BIN_IN (STRUCT)

16004

Die Struktur enthält die Eingangsdaten der digitalen AS-i Slaves:

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ SA_Slave	ARRAY[1...31] OF BYTE	Eingangsdaten der digitalen S/A-Slaves, 1 Byte pro S/A-Slave	entspricht 1. bis 15. Wort des azyklischen Datensatzes 2 (DS2) → DS2 – Digitale Slave-Eingänge und Masterflags
▪ B_Slave	ARRAY[1...31] OF BYTE	Eingangsdaten der digitalen B-Slaves, 1 Byte pro B-Slave	entspricht 16. bis 31. Wort des azyklischen Datensatzes 2 (DS2) → DS2 – Digitale Slave-Eingänge und Masterflags

ASI_BIN_OUT (STRUCT)

16003

Die Struktur enthält die Ausgangsdaten der digitalen AS-i Slaves.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ SA_Slave_bin_OUT	ARRAY[1...31] OF BYTE	Ausgangsdaten der digitalen S/A-Slaves (1 Byte pro S/A-Slave)	entspricht 1. bis 15. Wort des azyklischen Datensatzes 5 (DS5) → DS5 – Digitale Slave-Ausgänge
▪ B_Slave_bin_OUT	ARRAY[1...31] OF BYTE	Ausgangsdaten der digitalen B-Slaves (1 Byte pro B-Slave)	entspricht 16. bis 31. Wort des azyklischen Datensatzes 5 (DS5) → DS5 – Digitale Slave-Ausgänge

ASI_ANA_IO (STRUCT)

16001

Die Struktur enthält die Prozessdaten der analogen Ein- und Ausgangs-Slaves.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ ana_IN_Slave	ARRAY[1..31] OF ASI_ANALOG_IN	Eingangsdaten und Status-Flags der analogen AS-i Slaves	→ ASI_ANALOG_IN (STRUCT) (→ S. 179)
▪ ana_OUT_Slave	ARRAY[1..31] OF ASI_ANALOG_OUT	Ausgangsdaten und Status-Flags der analogen AS-i Slaves	→ ASI_ANALOG_OUT (STRUCT) (→ S. 180)

ASI_ANALOG_IN (STRUCT)

15994

Die Struktur enthält die Prozessdaten der analogen Eingangs-Slaves sowie die übertragenen Status-Flags.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ chan_1	INT	Analogwert Kanal 1: S oder A-Eingangs-Slave	entspricht 1. bis 4. Wort eines 5 Worte umfassenden Bereichs in den azyklischen datensätzen 3 und 4 (DS3+4) → DS3 – Analoge Eingänge der Slaves 1(A)...15(B) → DS4 – Analoge Eingänge der Slaves 16(A)...31(B)
▪ chan_2	INT	Analogwert Kanal 2: S- oder A-Eingangs-Slave	
▪ chan_3	INT	Analogwert Kanal 3: S- oder B-Eingangs-Slave	
▪ chan_4	INT	Analogwert Kanal 4: S- oder B-Eingangs-Slave	
▪ flags	ASI_ANALOG_OUT_FLAGS	Status-Flags	→ ASI_ANALOG_IN_FLAGS (STRUCT) (→ S. 180)

ASI_ANALOG_IN_FLAGS (STRUCT)

15996

Die Struktur enthält die Status-Flags des analogen Eingangs-Slaves.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ V0	BOOL	Valid-Bit, Kanal 1	entspricht dem 5. Wort eines 5 Worte umfassenden Bereichs in den azyklischen Datensätzen 3 und 4 (DS3+4) → DS3 – Analoge Eingänge der Slaves 1(A)...15(B) → DS4 – Analoge Eingänge der Slaves 16(A)...31(B)
▪ O0	BOOL	Overflow-Bit, Kanal 1	
▪ V1	BOOL	Valid-Bit, Kanal 2	
▪ O1	BOOL	Overflow-Bit, Kanal 2	
▪ V2	BOOL	Valid Bit Kanal 3	
▪ O2	BOOL	Overflow-Bit, Kanal 3	
▪ V3	BOOL	Valid-Bit, Kanal 4	
▪ O3	BOOL	Overflow-Bit, Kanal 5	
▪ na1	BOOL	--	
▪ TOA	BOOL	Transfer-Output, S/A-Slave	
▪ na2	BOOL	--	
▪ TOB	BOOL	Transfer-Output, B-Slave	
▪ na3	BOOL	--	
▪ TIA	BOOL	Transfer-Input, S/A-Slave	
▪ na4	BOOL	--	
▪ TIB	BOOL	Transfer-Input, B-Slave	

ASI_ANALOG_OUT (STRUCT)

15995

Die Struktur bildet die Prozessdaten der analogen Ausgangs-Slaves sowie die übertragenen Flags ab.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ chan_1	INT	Analogwert Kanal 1: S oder A-Ausgangs-Slave	entspricht azyklischen Datensätzen 6 und 7 (DS6, DS7) → DS6 – Analoge Ausgänge der Slaves 1(A)...15(B) → DS7 – Analoge Ausgänge der Slaves 16(A)...31(B)
▪ chan_2	INT	Analogwert Kanal 2: S- oder A-Ausgangs-Slave	
▪ chan_3	INT	Analogwert Kanal 3: S- oder B-Ausgangs-Slave	
▪ chan_4	INT	Analogwert Kanal 4: S- oder B-Ausgangs-Slave	
▪ flags	ASI_ANALOG_OUT_FLAGS	Status-Flags	→ ASI_ANALOG_OUT_FLAGS (STRUCT) (→ S. 181)

ASI_ANALOG_OUT_FLAGS (STRUCT)

15997

Die Struktur enthält die Status-Flags des analogen Ausgangs-Slaves.

Bezeichnung	Datentyp	Bedeutung	Mögliche Werte
▪ na1	BOOL	--	entspricht dem azyklischen Datensatz 8 (DS8) → DS8 – Statusflags der analogen Ausgänge der Slaves 1...31
▪ na2	BOOL	--	
▪ na3	BOOL	--	
▪ na4	BOOL	--	
▪ na5	BOOL	--	
▪ na6	BOOL	--	
▪ na7	BOOL	--	
▪ na8	BOOL	--	
▪ OVA	BOOL	Output valid, S/A-Slave	
▪ na9	BOOL	--	
▪ OVB	BOOL	Output valid, B-Slave	
▪ n10	BOOL	--	
▪ TOA	BOOL	Transfer Output, S/A-Slave	
▪ na11	BOOL	--	
▪ TOB	BOOL	Transfer Output, B-Slave	
▪ na12	BOOL	--	

© ifm electronic gmbh

10.2 Bibliothek ACnnnn_SYS_CMD.library

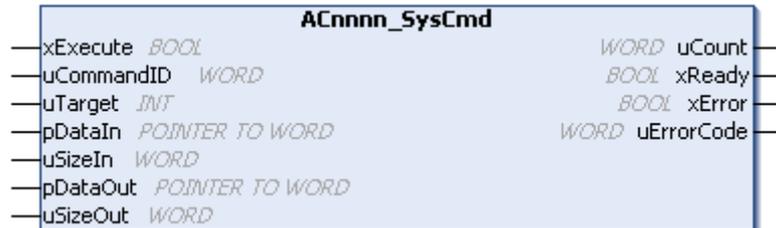
Inhalt	
ACnnnn_SysCmd	183
	17723

10.2.1 ACnxxx_SysCmd

15890

Baustein-Typ: Funktionsbaustein (FB)
Bibliothek: ACnxxx_SYS_CMD.library

Symbol in CODESYS:



Beschreibung

15893

Mit dem FB können einzelne Kommandos an das System oder einen AS-i Master gesendet werden. Jedes Kommando bezieht sich auf die Datenstrukturen eines der folgenden Elemente:

- Kommando-Anforderungs-Kanal: → **Beispiel: Spracheinstellung des Geräts ändern** (→ S. 187)
- Kommando-Antwort-Kanal: → **Beispiel: Datum / Uhrzeit / NTP-Einstellungen lesen** (→ S. 188)
- Azyklischen Datensatz (DSx): → **Beispiel: LCEMS, LCEAS und LDAE von AS-i Master 2 lesen** (→ S. 189)

Eingangsparameter

15895

Parameter	Datentyp	Bedeutung	Mögliche Werte	
xExecute	BOOL	Ausführung des FB steuern	FALSE	FB-Ausführung stoppen
			TRUE	Einmalige FB-Ausführung starten
uCommandID	WORD	ID des auszuführenden Kommandos	→ Tabelle: Systemkommandos (→ S. 184) → Tabelle: AS-i Master-Kommandos (→ S. 184)	
uTarget	INT	Geräte-Komponente, an die das Kommando geschickt werden soll	0	System
			1	AS-i Master 1
			2	AS-i MAster 2
pDataIn	POINTER TO WORD	Puffer für Daten, die dem Kommando als Eingangsparameter übergeben werden.	Im Kommando-Anforderungs-Kanal beschriebene Kommandoparameter (Worte 3 bis 120)	
uSizeIn	WORD	Größe des Puffers für die Eingangsparameter (Anzahl der Bytes)		
pDataOut	POINTER TO WORD	Puffer für Daten, die als Ausgangsparameter zurückgegeben werden (Ergebnisse)	Im Kommando-Antwort-Kanal beschriebene Rückgabewerte (Daten ab dem Wort 5)	
uSizeOut	WORD	Größe des Puffers für die Ausgangsparameter (Anzahl der Bytes)		

Tabelle: Systemkommandos

17053

uCommandID	Beschreibung
0x0001	DS1 – Systeminformationen
0x1101	Kommando 0x0101 – Quick Setup AS-i Master 1 + 2
0x1103	Kommando 0x0103 – Bediensprache umschalten
0x1104	Kommando 0x0104 – Display-Einstellungen ändern
0x1105	Kommando 0x0105 – Ausgangskontrolle setzen
0x1109	Kommando 0x0109 – Datum / Uhrzeit setzen
0x110A	Kommando 0x010A – Einstellungen des NTP-Servers setzen
0x110B	Kommando 0x010B – Datum / Uhrzeit / NTP-Einstellungen lesen
0x110C	Kommando 0x010C – System neu starten (Reboot)
0x110D	Kommando 0x010D – Feldbus-Info lesen
0x1110	Kommando 0x0110 - Target-Visualisierung anzeigen



Systemkommandos nur mit dem Eingangsparameter uTarget = 0 ausführen!



Detaillierte Informationen zu den azyklischen Datensätzen und der Kommandoschnittstelle enthält die Ergänzung zum Gerätehandbuch des SmartSPS SafeLine AC4S (→ **Übersicht: Anwender-Dokumentation für AC4S** (→ S. 7)).

Tabelle: AS-i Master-Kommandos

17054

uCommandID	Beschreibung / korrespondierendes Kommando
0x0002	DS2 – Digitale Slave-Eingänge und Masterflags
0x0003	DS3 – Analoge Eingänge der Slaves 1(A)...15(B)
0x0004	DS4 – Analoge Eingänge der Slaves 16(A)...31(B)
0x0005	DS5 – Digitale Slave-Ausgänge
0x0006	DS6 – Analoge Ausgänge der Slaves 1(A)...15(B)
0x0007	DS7 – Analoge Ausgänge der Slaves 16(A)...31(B)
0x0008	DS8 – Statusflags der analogen Ausgänge der Slaves 1...31
0x0009	DS9 – Slave-Listen LAS, LDS, LPF, LCE
0x000A	DS10 – Slave-Liste LPS
0x000B	DS11 – Aktuelle Konfigurationsdaten (CDI)
0x000C	DS12 – Projektierte Konfigurationsdaten (PCD)
0x000D	DS13 – Abbild der Eingangsparameter der Slaves (PI)
0x000E	DS14 – Abbild der Ausgangsparameter der Slaves (PP)
0x000F	DS15 – Slave-Fehlerzähler, Konfigurations-Fehlerzähler, AS-i Zykluszähler
0x0011	DS17 – AS-i Master: Fehlerlisten LCEMS, LCEAS, LDAE
0x1001	Kommando 0x0001 – AS-i Slave-Parameter ändern

uCommandID	Beschreibung / korrespondierendes Kommando
0x1003	Kommando 0x0003 – Aktuelles AS-i Netz projektieren
0x1004	Kommando 0x0004 – LPS ändern
0x1005	Kommando 0x0005 – Betriebsmodus des AS-i Masters ändern
0x1006	Kommando 0x0006 – AS-i Slave-Adresse ändern
0x1007	Kommando 0x0007 – Autoadress-Modus des AS-i Masters einstellen
0x1009	Kommando 0x0009 – Extended ID1 im AS-i Slave ändern
0x100A	Kommando 0x000A – PCD ändern
0x100D	Kommando 0x000D – AS-i Master Versorgungsspannung, Symmetrie, Erdschluss
0x1015	Kommando 0x0015 – ID-String eines AS-i Profils (S-7.4) lesen
0x101A	Kommando 0x001A – AS-i Master-Info lesen
0x101C	Kommando 0x001C – Slave-Reset beim Übergang in den geschützten Betrieb deaktivieren
0x1021	Kommando 0x0021 – Diagnose-String eines AS-i Slave (S-7.4) lesen
0x1022	Kommando 0x0022 – Parameter-String eines AS-i Slave (S-7.4) lesen
0x1023	Kommando 0x0023 – Parameter-String eines AS-i Slave (S-7.4) schreiben
0x1024	Kommando 0x0024 – CTT2 Standard Read
0x1025	Kommando 0x0025 – CTT2 Standard Write
0x1026	Kommando 0x0026 – CTT2 Vendor Specific Read
0x1027	Kommando 0x0027 – CTT2 Vendor Specific Write
0x1040	Kommando 0x0040 – CTT2 Device Group Read
0x1041	Kommando 0x0041 – CTT2 Device Group Write
0x1042	Kommando 0x0042 – CTT2 Vendor Specific Selective Read from Buffer
0x1043	Kommando 0x0043 – CTT2 Vendor Specific Selective Write from Buffer
0x1044	Kommando 0x0044 – CTT2 Vendor Specific Selective Read
0x1045	Kommando 0x0045 – CTT2 Vendor Specific Selective Write
0x1046	Kommando 0x0046 – CTT2 Device Group Selective Read
0x1047	Kommando 0x0047 – CTT2 Device Group Selective Write
0x1049	Kommando 0x0049 – CTT2 Vendor Specific Exchange
0x104A	Kommando 0x004A – CTT2 Device Group Exchange
0x104B	Kommando 0x004B – CTT2 Device Group Selective Read from Buffer
0x104C	Kommando 0x004C – CTT2 Device Group Selective Write from Buffer
0x1050	Kommando 0x0050 – AS-i Master-Einstellungen setzen
0x1051	Kommando 0x0051 – Fehlerzähler zurücksetzen



AS-i Master-Kommandos nur mit dem Eingangsparameter uTarget = 1 oder 2 ausführen!



Detaillierte Informationen zu den azyklischen Datensätzen und der Kommandoschnittstelle enthält die Ergänzung zum Gerätehandbuch des SmartSPS SafeLine AC4S (→ **Übersicht: Anwender-Dokumentation für AC4S** (→ S. 7)).

Ausgangsparameter

15897

Parameter	Datentyp	Bedeutung	Mögliche Werte	
uCount	WORD	Anzahl der gültigen Bytes in pDataOut ($uCount \leq uSizeOut$)	ganzahliger Wert in Hexadezimal-Darstellung	
xReady	BOOL	Signal zeigt an, ob die Ausführung des FB abgeschlossen ist.	FALSE	FB ist inaktiv oder wird ausgeführt.
			TRUE	FB-Ausführung ist beendet.
xError	BOOL	Signal zeigt an, ob während der FB-Ausführung Fehler aufgetreten sind.	FALSE	FB ist deaktiviert oder wird gerade ausgeführt oder FB-Ausführung verlief fehlerlos.
			TRUE	Bei der Ausführung des FB ist ein Fehler aufgetreten.
uErrorCode	WORD	Fehler-Code des ausgeführten Kommandos	Kommando-Fehlercode des korrespondierenden Kommando-Kanals ODER: → Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0F01 Unbekannter Fehler
- 0x0F02 Unbekanntes/Ungültiges Ziel
- 0x0F03 Unbekannte Kommando-ID
- 0x0F04 Ungültige Parameter
- 0x0F05 Zeitüberschreitung bei Abarbeitung (Timeout)

Beispiel: Spracheinstellung des Geräts ändern

17051

Aufgabe: Mit dem FB ACnxxx_SysCmd die Sprache der grafischen Benutzeroberfläche des Geräts auf "Spanisch" setzen.

Befehlstyp: Kommando-Anforderungs-Kanal

Eingangsparameter des FB:

Parameter	Wert	Erklärung
uCommandID	0x1103	Korrespondierendes Kommando: → Kommando 0x0103 – Bediensprache umschalten (→ Ergänzung Gerätehandbuch)
uTarget	0	System-Kommando
pDataIn	arDataIn	<ul style="list-style-type: none"> ▪ Variable vom Datentyp Array of Words ▪ arDataIn enthält die Kommandoparameter
uSizeIn	0x0001	arDataIn besteht nur aus 1 Zeile, da Wort 1 und 2 sowie 4 bis 120 des Kommando-Anforderungskanals unberücksichtigt bleiben.
pDataOut	--	irrelevant, da Kommando-Anforderungs-Kanal
uSizeOut	--	irrelevant, da Kommando-Anforderungs-Kanal

Inhalt von arDataIn:

Wort-Nr.	Inhalt	Erklärung
1	0x4553	Spanisch

Beispiel: Datum / Uhrzeit / NTP-Einstellungen lesen

17423

Aufgabe: Mit dem FB ACnxxx_SysCmd die aktuelle Systemzeit und die NTP-Einstellungen lesen.**Befehlstyp:** Kommando-Antwort-Kanal**Eingangsparameter des FB:**

Parameter	Wert	Erklärung
uCommandID	0x110B	Korrespondierendes Kommando. → Kommando 0x010B – Datum / Uhrzeit / NTP-Einstellungen lesen (→ Ergänzung Gerätehandbuch)
uTarget	0	System-Kommando
pDataIn	--	irrelevant, da Kommando-Antwort-Kanal
uSizeIn	--	irrelevant, da Kommando-Antwort-Kanal
pDataOut	arDataOut	<ul style="list-style-type: none"> ▪ Variable vom Datentyp Array of Words ▪ enthält die Rückgabewerte des Antwort-Kanals
uSizeOut	0x0007	Array besteht aus 7 Zeilen, da der Antwort-Kanal 7 Worte zurückgibt (Worte 5 ... 11).

Inhalt von arDataOut:

Wort-Nr.	Inhalt
1	Monat Tag
2	Jahr Jahr
3	Minuten Stunden
4	reserviert Sekunden
5	NTP-Offset NTP-Status
6 ... 7	IP-Adresse NTP-Server

Beispiel: LCEMS, LCEAS und LDAE von AS-i Master 2 lesen

17424

Aufgabe: Mit dem FB ACnxxx_SysCmd die Fehlerlisten LCEMS, LCEAS und LDAE des AS-i Master 2 lesen.

Befehlstyp: Azyklischer Datensatz

Eingangsparameter des FB:

Parameter	Wert	Erklärung
uCommandID	0x0011	Korrespondierender azyklischer Datensatz: → DS17 – AS-i Master: Fehlerlisten LCEMS, LCEAS, LDAE (→ Ergänzung Gerätehandbuch)
uTarget	1	Master-Kommando (1 = AS-i Master 2))
pDataIn	--	irrelevant
uSizeIn	--	irrelevant
pDataOut	arDataOut	<ul style="list-style-type: none"> ▪ Variable vom Datentyp Array of Words ▪ enthält die Rückgabewerte
uSizeOut	0x000C	Array besteht aus 12 Zeilen, da der Datensatz DS17 genau 12 Worte umfasst

Inhalt von arDataOut:

Wort-Nr.	Inhalt
1 ... 4	LCEMS (Liste der Konfigurationsfehler - Fehlende Slaves)
5 ... 8	LCEAS (Liste der Konfigurationsfehler - Zusätzliche Slaves)
9 ... 12	LDAE (Liste der Doppeladressfehler)

10.3 Bibliothek SF_IO.library

Inhalt	
CtrlASi_InSlave	191
CtrlASi_OutSlave.....	193
CtrlASi_ResetAllSlaves	196
CtrlLocalInputs.....	198
Ctrl_SetDiagInfo	200
GetLocalInput	203
SetLocalOutput.....	204

18930

Die Bibliothek SF_IO.library der ifm electronic stellt Funktionsbausteine (FUB) für die Programmierung sicherheitsgerichteter SPS-Applikationen bereit.

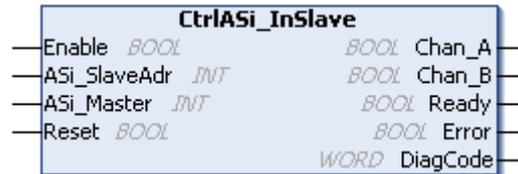
10.3.1 CtrlASi_InSlave

18240

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18239

Funktionsbaustein (FB) zur Steuerung des logischen Geräts eines sicheren AS-i Eingangs-Slaves (ASi_SlaveAdr, ASi_Master).

Mit CtrlASi_InSlave kann der Programmierer folgende Aktionen ausführen:

- das logische Gerät entsperren und aus dem verriegelten Fehlerzustand S_ERROR in den Initialisierungszustand S_INIT versetzen (RESET)

Der FB gibt folgende Zustands- und Diagnoseinformationen als nicht-sichere Daten aus:

- logische Werte der beiden sicheren Code-Halbfolgen des AS-i Eingangs-Slaves (Chan_A, Chan_B)
- Zustand der FB-Abarbeitung (Ready)
- Fehlerzustand des Funktionsbausteins (Error)
- Diagnoseinformationen des logischen Geräts (DiagCode)



Der Funktionsblock kann innerhalb der Grenzen von CODESYS beliebig oft instanziiert werden.

Eingangsparameter

18237

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
ASi_SlaveAdr	INT	Adresse des sicheren AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Masters, an den der sichere AS-i Slave angeschlossen ist.	1	AS-i Master 1
			2	AS-i Master 2
Reset	BOOL	Steuersignal zum Zurücksetzen des logischen Geräts aus dem verriegelten Fehlerzustand (S_ERROR) in den Initialisierungszustand (S_INIT).	FALSE ⇔ TRUE	Steuersignal wird gesendet
			sonst.	keine Änderungen



Der sichere AS-i Slave, der mithilfe des FB gesteuert werden soll, muss im Gerätebaum eingebunden sein!

Ausgangsparameter

18238

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Chan_A	BOOL	Logischer Zustand des sicheren Signalkanals A (= 1. Code-Halbfolge)	FALSE	OFF
			TRUE	ON
Chan_B	BOOL	Logischer Zustand des sicheren Signalkanals B (= 2. Code-Halbfolge)	FALSE	OFF
			TRUE	ON
Ready	BOOL	Status der FB-Abarbeitung (Gültigkeit der Ausgangssignale)	FALSE	FB nicht fertig abgearbeitet oder deaktiviert; Signale sind ungültig
			TRUE	FB fertig abgearbeitet; Signale sind gültig
Error	BOOL	Fehleranzeige	FALSE	kein Fehler
			TRUE	Logisches Gerät befindet sich im verriegelten Fehlerzustand S_ERROR
DiagCode	WORD	Diagnose-Code der internen Zustände des gesteuerten logischen Geräts	Umfang der Diagnose-Codes ist abhängig vom gesteuerten logischen Gerät: Beschreibung des gewählten logischen Geräts (→ Bibliothek SF_LogicalInterfaces.library (→ S. 206))	

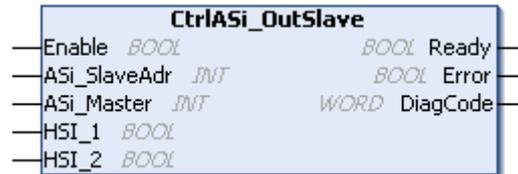
10.3.2 CtrlASi_OutSlave

18244

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18243

Funktionsbaustein (FB) zur Steuerung des logischen Geräts eines sicheren AS-i Control-Slaves (ASi_SlaveAdr, ASi_Master)

Mit CtrlASi_OutSlave kann der Programmierer folgende Aktionen ausführen:

- Signal für das Entsperren der sicheren AS-i Slaves senden (Hilfssignal HSI_1)
- Signal für den automatischen Wiederanlauf senden (Hilfssignal HSI_2)

Der FB gibt folgende Zustands- und Diagnoseinformationen aus:

- Zustand der FB-Abarbeitung (Ready)
- Fehlerzustand des Funktionsbausteins (Error)
- Diagnoseinformationen des logischen Geräts (DiagCode)



Der Funktionsblock kann innerhalb der Grenzen von CODESYS beliebig oft instanziiert werden.

Eingangsparameter

18242

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
ASi_SlaveAdr	INT	Adresse des sicheren AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Masters, an den der sichere AS-i Slave angeschlossen ist.	1	AS-i Master 1
			2	AS-i Master 2
HSI_1	BOOL	Hilfssignal 1 (HSI_1) Reset-Signal zum Verlassen des verriegelten Fehlerzustands (S_Error)	FALSE ⇒ TRUE	Hilfssignal wird einmalig gesendet
			sonst.	keine Aktion
HSI_2	BOOL	Hilfssignal 2 (HSI_2) Steuerung des Anlaufverhaltens des gesteuerten AS-i Slaves nach Kommunikationsfehler	FALSE ⇒ TRUE	Hilfssignal wird einmalig gesendet
			sonst.	keine Aktion



Der sichere AS-i Slave, der mithilfe des FB gesteuert werden soll, muss im Gerätebaum eingebunden sein!

Eine einmal ausgelöste Anforderung der beiden Hilfssignal darf nicht zurückgenommen werden, bis der Erfolg oder der Abbruch der Übertragung angezeigt wurde.

- Die Eingänge HSI_1 und/oder HSI_2 erst wieder auf FALSE setzen, wenn am Ausgang DiagCode ein Wert mit HS_ACK = OK oder HS_ACK = nOK anliegt (→ **Ausgangsparameter** (→ S. [195](#))).



Detaillierte Informationen zu den Hilfssignalen HSI_1 und HSI_2: → Betriebsanleitung des sicheren AS-i Control-Slaves

Ausgangsparameter

18241

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Ready	BOOL	Status der FB-Abarbeitung (Gültigkeit der Ausgangssignale)	FALSE	FB nicht fertig abgearbeitet oder deaktiviert; Signale sind ungültig
			TRUE	FB fertig abgearbeitet; Signale sind gültig
Error	BOOL	Fehleranzeige	FALSE	kein Fehler
			TRUE	Logisches Gerät befindet sich im verriegelten Fehlerzustand S_ERROR
DiagCode	WORD	Diagnose-Code der internen Zustände des gesteuerten logischen Geräts	Umfang der Diagnose-Codes ist abhängig vom gesteuerten logischen Gerät: Beschreibung des gewählten logischen Geräts (→ Bibliothek SF_LogicalInterfaces.library (→ S. 206))	

Diagnose-Codes:

- 0x8000 Ausgang = EIN; HS_ACK = idle
- 0x8002 Ausgang = AUS; HS_ACK = idle
- 0x8830 Ausgang = EIN; HS_ACK = OK
- 0x8832 Ausgang = AUS; HS_ACK = OK
- 0x8840 Ausgang = EIN; HS_ACK = nOK
- 0x8842 Ausgang = AUS; HS_ACK = nOK
- 0x8850 Ausgang = EIN; HS_ACK = RUN
- 0x8852 Ausgang = AUS; HS_ACK = RUN

10.3.3 CtrlASi_ResetAllSlaves

18254

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18253

Funktionsbaustein (FB) zur Steuerung des logischen Gerats mehrerer sicherer AS-i Slaves eines AS-i Netzwerks (ASi_Master).

Mit CtrlASi_ResetAllSlaves kann der Programmierer folgende Aktionen ausfuhren:

- das logische Gerat aller sicherer AS-i Slaves im angegebenen AS-i Netzwerk entsperren und aus dem verriegelten Fehlerzustand S_ERROR in den Initialisierungszustand S_INIT versetzen (RESET)

Der FB gibt folgende Zustands- und Diagnoseinformationen aus:

- Zustand der FB-Abarbeitung (Ready)
- Fehlerzustand des Funktionsbausteins (Error)

Eingangsparameter

18252

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
ASi_Master	INT	AS-i Masters, an den der sichere AS-i Slave angeschlossen ist.	1	AS-i Master 1
			2	AS-i Master 2
Reset	BOOL	Steuersignal zum Zurücksetzen des logischen Gerats aus dem verriegelten Fehlerzustand (S_ERROR) in den Initialisierungszustand (S_INIT).	FALSE ⇔ TRUE	Steuersignal wird gesendet
			sonst.	keine nderungen

Ausgangsparameter

18255

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Ready	BOOL	Status der FB-Abarbeitung (Gültigkeit der Ausgangssignale)	FALSE	FB nicht fertig abgearbeitet oder deaktiviert; Signale sind ungültig
			TRUE	FB fertig abgearbeitet; Signale sind gültig
Error	BOOL	Fehleranzeige	FALSE	keines der logischen Geräte befindet sich im verriegelten Fehlerzustand S_ERROR
			TRUE	Mindestens eines der logischen Geräte befindet sich im verriegelten Fehlerzustand S_ERROR

10.3.4 CtrlLocalInputs

18260

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18268

Funktionsbaustein (FB) zur Steuerung des logischen Geräts für zwei Eingangskanäle der lokalen E/A-Schnittstelle (IN_Channel_A, IN_Channel_B)

Mit CtrlLocalInputs kann der Programmierer folgende Aktionen ausführen:

- das logische Gerät entsperren und aus dem verriegelten Fehlerzustand S_ERROR in den Initialisierungszustand S_INIT versetzen (Reset)

Der FB gibt folgende Zustands- und Diagnoseinformationen aus:

- logische Werte der beiden Eingangskanäle (Chan_A, Chan_B)
- Zustand der FB-Abarbeitung (Ready)
- Fehlerzustand des Funktionsbausteins (Error)
- Diagnoseinformationen des logischen Geräts (DiagCode)



Der FB kann in den Grenzen von CODESYS beliebig oft instanziiert werden.

Eingangsparameter

18267

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
Reset	BOOL	Steuersignal zum Zurücksetzen des logischen Geräts aus dem verriegelten Fehlerzustand (S_ERROR) in den Initialisierungszustand (S_INIT).	FALSE ⇔ TRUE	Steuersignal wird gesendet
			sonst.	keine Änderungen

Ausgangsparameter

18269

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Chan_A	BOOL	Logischer Zustand des Signalkanals A	FALSE	OFF
			TRUE	ON
Chan_B	BOOL	Logischer Zustand des sicheren Signalkanals B	FALSE	OFF
			TRUE	ON
Ready	BOOL	Status der FB-Abarbeitung (Gültigkeit der Ausgangssignale)	FALSE	FB nicht fertig abgearbeitet oder deaktiviert; Signale sind ungültig
			TRUE	FB fertig abgearbeitet; Signale sind gültig
Error	BOOL	Fehleranzeige	FALSE	kein Fehler
			TRUE	Logisches Gerät befindet sich im verriegelten Fehlerzustand S_ERROR
DiagCode	WORD	Diagnose-Code der internen Zustände des gesteuerten logischen Geräts	→ Fehler-Codes: Lokale E/A-Schnittstelle (→ S. 277) ODER: → Liste unten (Diagnose-Codes)	

Diagnose-Codes:

- 0x0000 Initialisierung

10.3.5 Ctrl_SetDiagInfo

18265

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18264

Funktionsbaustein (FB) zur Verarbeitung der Diagnoseinformationen eines PLCopen-Funktionsbausteins.

Mit Ctrl_SetDiagInfo kann der Programmierer folgende Aktionen ausführen:

- Diagnose-Code (Code) eines Funktionsbausteins der Safety PLCopen-Bibliothek auslesen und an das Online Support Center (OSC) der AC4S-Bedienoberfläche weiterleiten.
- Die erzeugte Fehlermeldung hat folgende Struktur:
Fehler PLCopen-FB (FB_Type), Instanz-Nr. (FB_Number): DC=%s1, DC-1=%s2
 - FB_Type = ID des Funktionsbausteins
 - FB_Number = Instanznummer des Funktionsbausteins
 - DC = aktueller Diagnosecode (als hexadezimaler Wert)
 - DC-1 = letzter Diagnosecode (als hexadezimaler Wert)

Der FB gibt folgende Zustands- und Diagnoseinformationen aus:

- Zustand der FB-Abarbeitung (Ready)
- Anzeige eines Fehlers bei FB-Abarbeitung (Error)



Der FB generiert für jeden Zustandswechsel eine separate Meldung im Online Support Center (OSC). Der Ringspeichers des OSC stößt deshalb in kurzer Zeit an seine Kapazitätsgrenzen.

- ▶ Den FB Ctrl_SetDiagInfo nur für die Fehlersuche nutzen (Debugging)!

Eingangsparameter

18263

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
FB_Type	INT	ID des Funktionsbausteins (wird vom Programmierer vergeben)	0x3D6B	SF_Antivalent
			0x5096	SF_EDM
			0x50A9	SF_EmergencyStop
			0x5151	SF_EnableSwitch
			0x516E	SF_Equivalent
			0x56DD	SF_ESPE
			0x56E2	SF_GuardLocking
			0x571A	SF_GuardMonitoring
			0x5725	SF_ModeSelector
			0x5A03	SF_MutingPar
			0x5A3C	SF_MutingPar_2Sensor
			0x5BC4	SF_MutingSeq
			0x5BFB	SF_OutControl
			0x5C48	SF_SafetyRequest
0x5C77	SF_TestableSafetySensor			
0x5D8F	SF_TwoHandControlTypeII			
0x5DB0	SF_TwoHandControlTypeIII			
FB_Number	INT	Instanznummer des Funktionsbausteins (wird vom Programmierer vergeben)		
DiagCode	INT	Diagnose-Code des Safety-PLCopen-Funktionsbausteins	→ Bedienungsanleitung Safety-PLCopen-Bibliothek in CODESYS	



Die Diagnose-Codes der Safety-PLCopen-FBs sind vom Datentyp WORD.

Ausgangsparameter

266

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Ready	BOOL	Status der Übertragung des Diagnose-Codes	FALSE	derzeit keine Übertragungsanforderung
			TRUE	Übertragung des Diagnose-Codes erfolgreich abgeschlossen
Error	BOOL	Fehleranzeige	FALSE	kein Fehler bei FB-Abarbeitung aufgetreten
			TRUE	Fehler bei FB-Abarbeitung aufgetreten; Übertragung des Diagnose-Codes fehlgeschlagen

10.3.6 GetLocalInput

18335

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18336

Funktionsbaustein (FB) für den Zugriff auf einen nicht-sicheren Eingangskanal der lokalen E/A-Schnittstelle (IN_Channel)

Mit GetLocalInput kann der Programmierer folgende Aktionen ausführen:

- den nicht-sicheren, logischen Zustand des gewählten Eingangskanals erfassen und den Wert ausgeben (IN).

Der FB gibt folgende Zustands- und Diagnoseinformationen aus:

- Zustand der FB-Abarbeitung (Ready)
- Fehlerzustand des Funktionsbausteins (Error)

Eingangsparameter

18337

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
IN_Channel	INT	Lokaler Eingangskanal, an den der Sensor angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8

Ausgangsparameter

18338

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN	BOOL	Logischer Zustand des gewählten Eingangskanals	FALSE	AUS
			TRUE	EIN
Ready	BOOL	Status der FB-Abarbeitung (Gültigkeit der Ausgangssignale)	FALSE	FB nicht fertig abgearbeitet oder deaktiviert; Signale sind ungültig
			TRUE	FB fertig abgearbeitet; Signale sind gültig
Error	BOOL	Fehleranzeige	FALSE	kein Fehler
			TRUE	bei FB-Abarbeitung ist Fehler aufgetreten

10.3.7 SetLocalOutput

18341

Baustein-Typ: Funktionsbaustein (FB)

Bibliothek: sf_io.library

Symbol in CODESYS:



Beschreibung

18342

Funktionsbaustein (FB) für den Zugriff auf einen nicht-sicheren Ausgangskanal der lokalen E/A-Schnittstelle (OUT_Channel)

Mit SetLocalOutput kann der Programmierer folgende Aktionen ausführen:

- am gewählten Ausgangskanal einen nicht-sicheren, booleschen Wert ausgeben (OUT)

Der FB gibt folgende Zustands- und Diagnoseinformationen aus:

- Zustand der FB-Abarbeitung (Ready)
- Fehlerzustand des Funktionsbausteins (Error)

Eingangsparameter

18343

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Enable	BOOL	Aktivierung des FB	FALSE	FB ist deaktiviert.
			TRUE	FB ist aktiviert.
OUT_Channel	INT	Lokaler Ausgang, an dem der Aktuator angeschlossen ist.	1 ... 4	lokaler Ausgang OUT1 ... lokaler Ausgang OUT4
OUT	BOOL	Gewünschter logischer Zustand des lokalen Ausgangskanals	FALSE	AUS
			TRUE	EIN



Die Ausgänge der lokalen E/A-Schnittstelle können eine maximale Schaltfrequenz von 25 Hz realisieren.

- Bei der Erzeugung digitaler Signale an den lokalen Ausgängen die minimale Ein- bzw. Ausschaltzeit von 20 ms beachten!

Ausgangsparameter

18344

Parameter	Datentyp	Beschreibung	Mögliche Werte	
Ready	BOOL	Status der FB-Abarbeitung (Gültigkeit der Ausgangssignale)	FALSE	FB nicht fertig abgearbeitet oder deaktiviert; Signale sind ungültig
			TRUE	FB fertig abgearbeitet; Signale sind gültig
Error	BOOL	Fehleranzeige	FALSE	kein Fehler
			TRUE	bei FB-Abarbeitung ist Fehler aufgetreten

10.4 Bibliothek SF_LogicalInterfaces.library

Inhalt	
Hinweise zu logischen Geräten	207
Logische Geräte für sichere AS-i Slaves.....	211
Logische Geräte für die lokale E/A-Schnittstelle	242

7269

Die Bibliothek SF_LogicalInterfaces.library stellt logische Geräte bereit.

10.4.1 Hinweise zu logischen Geräten

Inhalt	
Zustandsmaschinen für logische Vorverarbeitung	207
Zustände	207
Zustandsübergänge	207
Diagnose-Meldungen	210

18235

Zustandsmaschinen für logische Vorverarbeitung

7036

Die logische Vorverarbeitung der sicheren Eingangssignale erfolgt mithilfe von Zustandsmaschinen. Jedes logische Gerät verfügt über eine eigene Zustandsmaschine, die aus den Eingangssignalen und den eingestellten Parametern ein sicheres Prozesssignal erzeugt.

Eine Zustandsmaschine besteht aus folgenden Komponenten:

- Zuständen (→ **Zustände** (→ S. [207](#)))
- Zustandsübergängen (→ **Zustandsübergänge** (→ S. [207](#)))

Zustände

14281

Anzahl und Art der Zustände ergeben sich aus dem Umfang der logischen Auswertefunktion. Die genaue Beschreibung der Zustände liefert die Dokumentation der Diagnose-Meldungen der einzelnen logischen Geräte.

Zustandsübergänge

18233

In jedem SPS-Zyklus überprüft die Zustandsmaschine einmal, welche Bedingung für einen Zustandsübergang erfüllt ist.

Zustandsübergänge können durch folgende Bedingungen ausgelöst werden:

- Signalzustände der Eingangssignale (→ **Zustände der Eingangssignale** (→ S. [208](#)))
- System-Events (→ **System-Events** (→ S. [209](#)))
- Timer-Events (→ **Timer-Events** (→ S. [209](#)))

Zustände der Eingangssignale

18945

In Abhängigkeit von der Art des logischen Gerätes gelten folgende Definitionen:

- **AS-i Slaves**
Sichere AS-i Eingangs-Slaves übertragen ihre Eingangsdaten als eindeutige Code-Folgen. Eine Code-Folge besteht aus 2 Halbfolgen, wobei jede Halbfolge einen Signalkanal repräsentiert.
- **Lokale E/A-Schnittstelle**
An der lokalen E/A-Schnittstelle repräsentiert jeder Eingang einen Signalkanal.

Aus der Kombination der beiden Signalkanäle ergeben sich folgende Zustände und die entsprechenden Zustands-Events:

Zustand	Zustand / Beschreibung	Übergangs-Event
NoChan	Die Signalkanäle A und B sind im logischen Zustand FALSE.	E_NoChan
ChanA_only	Signalkanal A ist im logischen Zustand TRUE. Signalkanal B ist im logischen Zustand FALSE.	E_ChanA_Only
ChanB_only	Signalkanal A ist im logischen Zustand FALSE. Signalkanal B ist im logischen Zustand TRUE.	E_ChanB_only
ChanAB	Die Signalkanäle A und B sind im logischen Zustand TRUE.	E_ChanAB
ChanInvalid*	Der logische Wert der Signalkanäle ist ungültig (z.B. Code-Folge ist unvollständig, Code-Folge ungültig)	E_ChanInvalid

* ... Nur verfügbar bei logischen Geräten für AS-i Eingangs-Slaves

Reihenfolge der Signalwechsel

10244

Aus der Kombination der Eingangssignale "ChanA_only" und "ChanB_only" sowie des Zustand des Parameters "P_ChanB_first" lassen sich darüber Zustandsübergänge definieren, welche die Reihenfolge der Signalwechsel an den Eingangskanälen berücksichtigt:

Zustand / Beschreibung	Übergangs-Event
ChanA = TRUE UND ChanB = FALSE UND Chan_B_first = TRUE	E_ChanA; P_ChanB_first = TRUE
ChanA = TRUE UND ChanB = FALSE UND Chan_B_first = FALSE	E_ChanA; P_ChanB_first = FALSE
ChanB = TRUE UND ChanA = FALSE UND Chan_B_first = TRUE	E_ChanB; P_ChanB_first = TRUE
ChanB = TRUE UND ChanA = FALSE UND Chan_B_first = FALSE	E_ChanB; P_ChanB_first = FALSE

Startup-Test

12226

Sind bei aktiviertem Startup-Test im Startzustand beide Eingangssignale "ChanA" = TRUE und "ChanB" = TRUE, so durchläuft die Zustandsmaschine zunächst einen Zustandszyklus, bis beide Eingangssignale gleichzeitig den Wert FALSE einnehmen.

Zustand / Beschreibung	Übergangs-Event
ChanA = TRUE UND ChanB = TRUE UND StartupTest = FALSE	E_ChanAB_without Startup Test
ChanA = TRUE UND ChanB = TRUE UND StartupTest = TRUE	E_ChanAB_with Startup Test

System-Events

18946

Folgende Systemereignisse starten die Abarbeitung der Zustandsmaschine des logischen Geräts:

- Start-Ereignis nach erfolgreicher Initialisierung der Zustandsmaschine (E_Start)

Folgende Systemereignisse versetzen die Zustandsmaschine des logischen Geräts in den Initialisierungszustand (S_INIT):

- Ein in der sicheren Anwendung erzeugtes Reset-Signal (E_ErrorReset)
- AS-i Eingangs-Slaves: Kommunikations- und Konfigurationsfehler (→ **Fehler-Codes: Sichere AS-i Slaves** (→ S. 275))
- Lokale Ein-/Ausgänge: Konfigurations- und Systemfehler (→ **Fehler-Codes: Lokale E/A-Schnittstelle** (→ S. 277))

Timer-Events

18236

Timer werden beim Verlassen eines Zustands gestartet. Ist der Timer abgelaufen, wird ein entsprechender Event erzeugt. Dieser Timer-Event dient der Zustandsmaschine als Bedingung für den Übergang in den nächsten Zustand. Treten zwei oder mehrere Timer-Events gleichzeitig auf, wird das Event mit der höchsten Priorität verarbeitet.

Die logischen Geräte können bei der Generierung des sicheren Prozesssignals folgende Zeiten auswerten:

Bezeichnung	Bedeutung	Timer-Event	Priorität
InvalidTime	<ul style="list-style-type: none"> ▪ Zeitspanne, innerhalb der die Signalkanäle einen ungültigen Zustand besitzen dürfen, ohne dass das Gerät in den verriegelten Fehlerzustand wechselt. ▪ fest eingestellt 	E_InvalidTimeout	1
SynchTime	<ul style="list-style-type: none"> ▪ Zeitspanne, innerhalb der die Signalkanäle den gleichen logischen Zustand einnehmen müssen. ▪ durch Programmierer einstellbar 	E_SynchTimeout	2
StabilisingTime	<ul style="list-style-type: none"> ▪ Zeitspanne, innerhalb der die Zustände der Signalkanäle stabil sein müssen, um einen Statuswechsel in der Auswertelogik auszulösen. ▪ durch Programmierer einstellbar 	E_StabTimeout	3
ToleranceTime	<ul style="list-style-type: none"> ▪ Zeitspanne, innerhalb der der Zustand FALSE an einem der beiden Signalkanäle toleriert wird, ohne ausschalten oder in den Zustand Test zu wechseln. ▪ durch Programmierer einstellbar 	E_TolTimeout	4

Diagnose-Meldungen

422

Mithilfe des Parameters "DisableLogging" kann der Programmierer die Weiterleitung von Zustandsmeldungen der logischen Geräte an das Online Support Center (OSC) steuern. Ist der Parameter deaktiviert, werden alle Zustandswechsel und Fehler-Ereignisse an das OSC weitergeleitet. Ist der Parameter aktiviert, werden nur die Fehler-Ereignisse an das OSC weitergeleitet.

Meldungstyp	Zustände	Bedeutung
Fehlermeldung	S_Test_e	Zustand zum Durchlaufen des Startup-Tests
	S_Error_e	verriegelter Fehlerzustand
	S_Init_e	Initialisierungszustand
Zustandsmeldung	S_OFF_e	Sicheres AUS
	S_ON_e	Sicheres AUS

10.4.2 Logische Geräte für sichere AS-i Slaves

Inhalt	
SF_IN_ASi_forced.....	212
SF_IN_ASi_independent.....	216
SF_IN_ASi_conditionally_dependent.....	220
SF_IN_ASi_dependent.....	224
SF_IN_ASi_dependent_filter_w_testreq.....	228
SF_IN_ASi_dependent_filter_ntestreq.....	232
SF_IN_ASi_dependent_filter_nshutdown.....	236
SF_OUTcontrol_ASi.....	240

19796

SF_IN_ASi_forced

18194

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_forced

Beschreibung

18201

Logisches Gerät für die Auswertung eines sicheren AS-i Eingangs-Slave mit 2 zwangsgeführten Kontakten/Kanälen

SF_IN_ASi_forced erfasst die Schaltzustände der beiden Signalkanäle des AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird beim Ein- und Ausschalten überwacht: Wenn beide Signalkanäle nicht innerhalb der fest eingestellten Zeit den gleichen Schaltzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 100ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Um das logische Gerät zu aus dem verriegelten Fehlerzustand zurückzusetzen:
→ **CtrlASi_InSlave** (→ S. [191](#))

Parameterdaten

18195

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON

* ... Standardwert (Default)

Prozessdaten

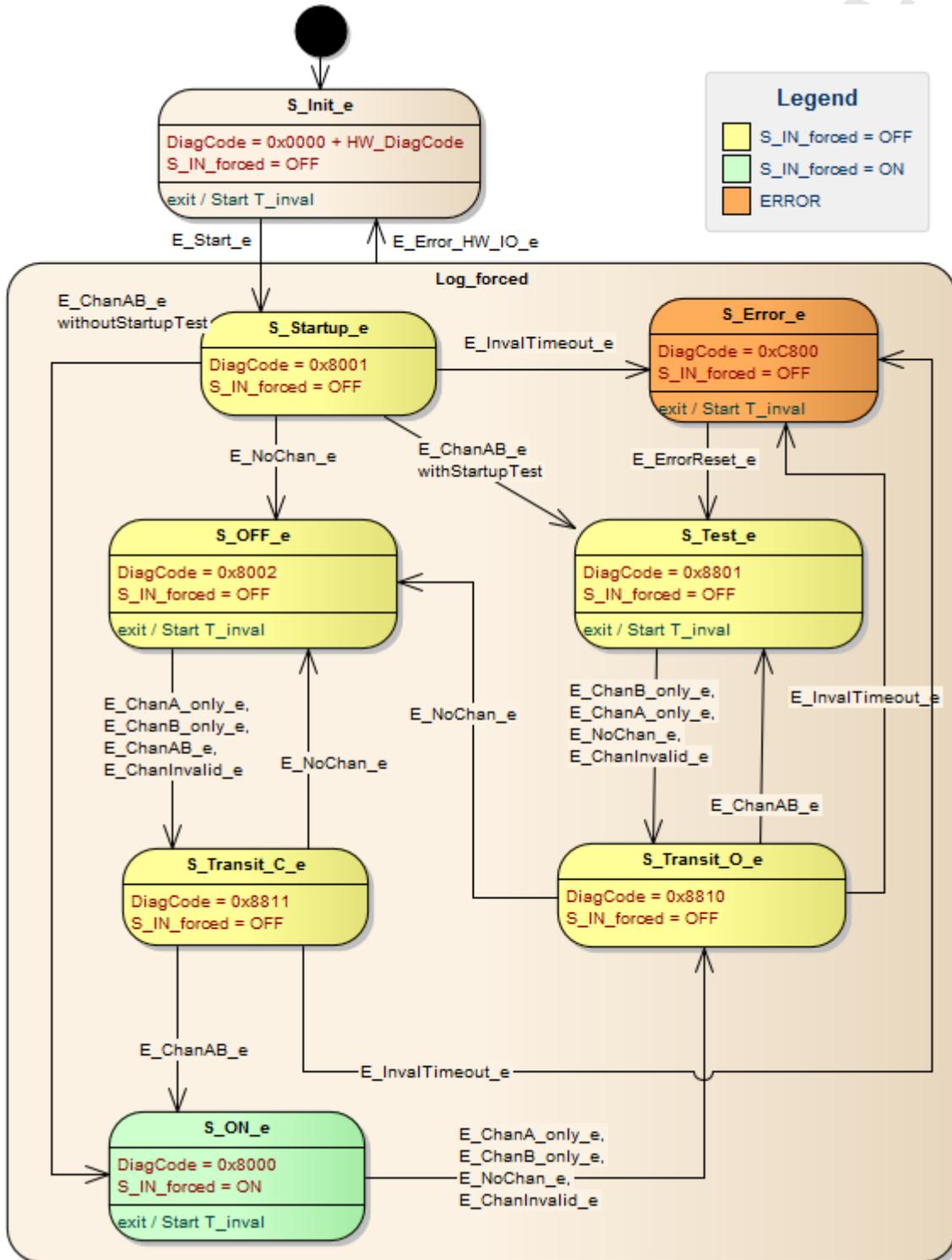
18200

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_forced	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS
			TRUE	Sicheres EIN

Zustandsdiagramm (State machine)

18933

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19760

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_ASi_independent

18212

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_independent

Beschreibung

18211

Logisches Gerät für einen sicheren Sensor mit 2 voneinander unabhängig wirkenden Kontakten/Kanälen

SF_IN_ASi_independent erfasst die Schaltzustände der beiden Signalkanäle/Kontakte eines sicheren AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Gleichzeitigkeit der Signalwechsel wird nicht überwacht.
- Wenn der Signalzustand der beiden Kanäle länger ungültig ist als die fest eingestellte Zeit, dann müssen beide Signalkanäle gleichzeitig den Zustand FALSE einnehmen, bevor wieder eingeschaltet werden kann. (InvalidTime = konst. = 2000ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.



Um das logische Gerät aus dem verriegelten Fehlerzustand zurückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlASi_InSlave** (→ S. [191](#))

Parameterdaten

18214

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON

* ... Standardwert (Default)

Prozessdaten

18213

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_indep	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicherheitsfunktion AUS
			TRUE	Sicherheitsfunktion EIN

Diagnose-Informationen

19769

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wider (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8822	S_Half	1 Kanal = TRUE	OFF

SF_IN_ASi_conditionally_dependent

18207

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_conditionally_dependent

Beschreibung

18208

Logisches Gerät für einen sicheren Sensor mit 2 voneinander bedingt abhängig wirkenden Kontakten/Kanälen

SF_IN_ASi_conditionally_dependent erfasst die Schaltzustände der beiden Signalkanäle des AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Reihenfolge der Signalwechsel der beiden Kontakte/Kanäle wird beim Ein- und Ausschalten überwacht: Wenn die Reihenfolge der Signalwechsel abweicht von der eingestellten Reihenfolge, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (Chan_B_first)
- Wenn der Signalzustand der beiden Signalkanäle länger ungültig ist als die fest eingestellte Zeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.



Um das logische Gerät aus dem verriegelten Fehlerzustand zurückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlASi_InSlave** (→ S. [191](#))

Parameterdaten

18209

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
Chan_B_first	BOOL	Gewünschte Reihenfolge der Signalzustandsänderungen	FALSE	Kanal A = TRUE vor Kanal B = TRUE
			TRUE	Kanal B = TRUE vor Kanal A = TRUE

* ... Standardwert (Default)

Prozessdaten

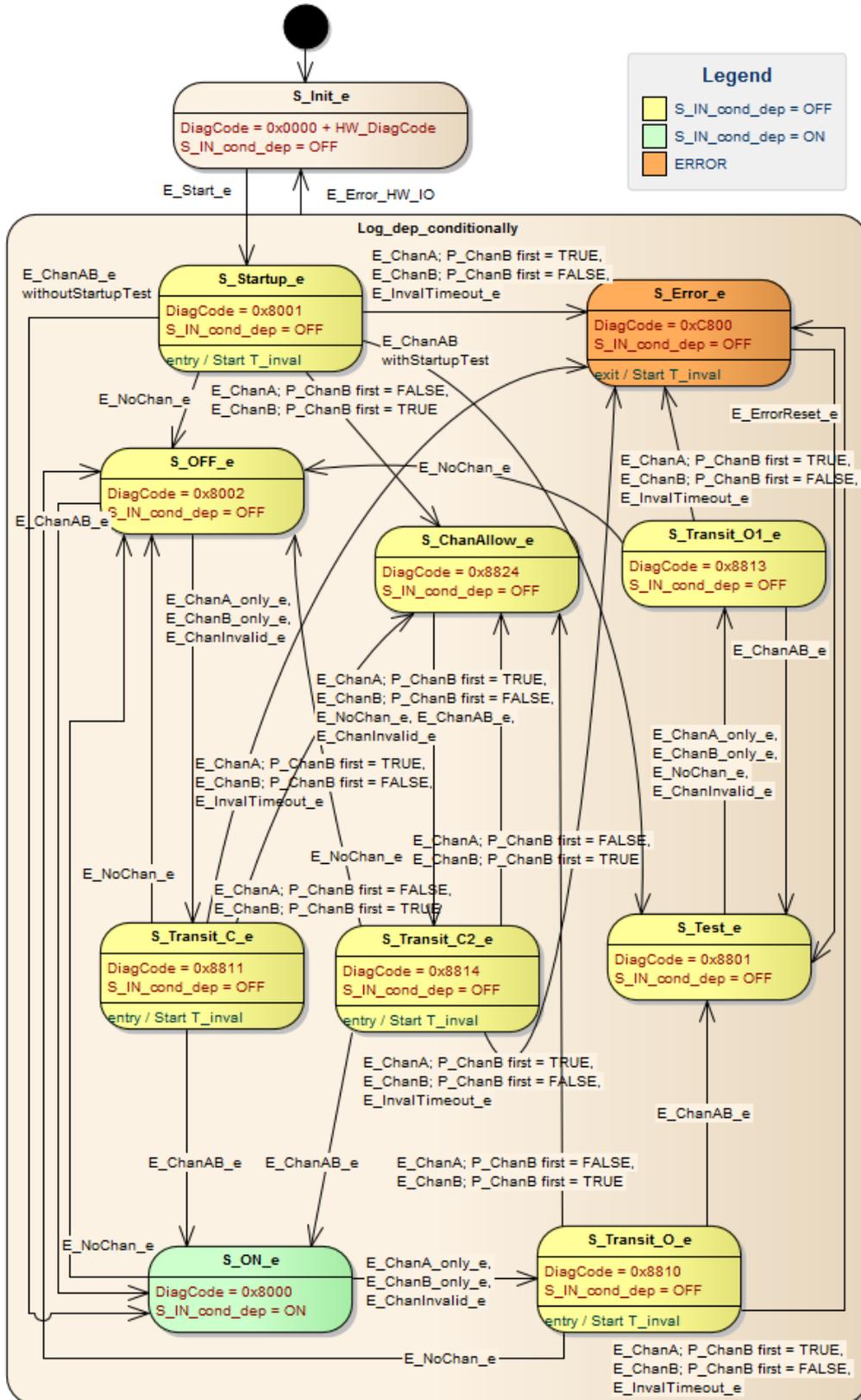
18210

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_cond_dep	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS
			TRUE	Sicheres EIN

Zustandsdiagramm (State machine)

594

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19776

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8813	S_Transit	Übergangszustand	OFF
0x8814	S_Transit	Übergangszustand	OFF
0x8824	S_ChanAllow	1 Kanal = TRUE, Signal-Reihenfolge o.K.	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_ASi_dependent

18202

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_dependent

Beschreibung

18203

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_ASi_dependent erfasst die Schaltzustände der beiden Signalkanäle/Kontakte des AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird beim Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Wenn der Signalzustand der beiden Signalkanäle länger ungültig ist als die fest eingestellte Zeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.



Um das logische Gerät aus dem verriegelten Fehlerzustand zurückzusetzen und Diagnose-Information des logischen Geräts auszulesen: → **CtrlASi_InSlave** (→ S. [191](#))

Parameterdaten

18204

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

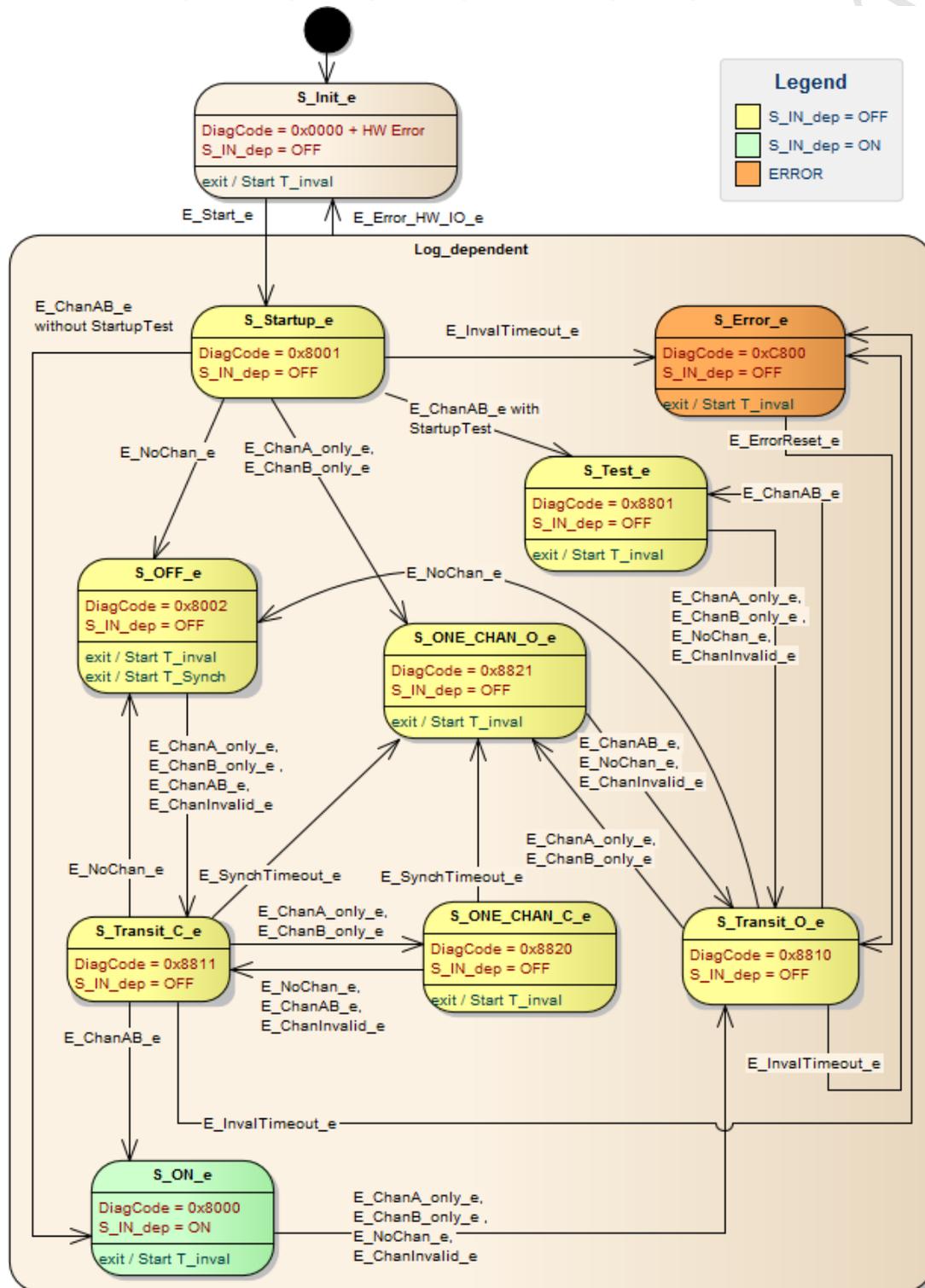
18205

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dependent	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS
			TRUE	Sicheres EIN

Zustandsdiagramm (State machine)

591

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Geräts:



Diagnose-Informationen

19773

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_ASi_dependent_filter_w_testreq

18215

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_dependent_filter_w_testreq

Beschreibung

18216

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_dependent_filter_w_testreq erfasst die Schaltzustände der beiden Kontakte/Kanäle des AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Kontakte/Kanäle werden beim Einschalten entprellt: Signalwechsel, die kürzer als die eingestellte Stabilisierungszeit anliegen, führen nicht zu einem Zustandswechsel. (StabilisingTime).
- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird bei Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Vor dem Wiedereinschalten müssen beide Kontakte/Kanäle gleichzeitig FALSE sein.
- Wenn nach dem Start die beiden Kontakte/Kanäle nicht innerhalb einer fest eingestellten Zeit einen gültigen Signalzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140 ms)
- Wenn die beiden Signale beim einem Umschaltvorgang länger ungültig als die eingestellte Stabilisierungszeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (StabilisingTime)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Um das logische Gerät aus dem verriegelten Fehlerzustand zurückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlASi_InSlave** (→ S. [191](#))

Parameterdaten

18217

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇔ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden
StabilisingTime	WORD	max. Prelldauer	250 ... 10000	250 Millisekunden * ... 10000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

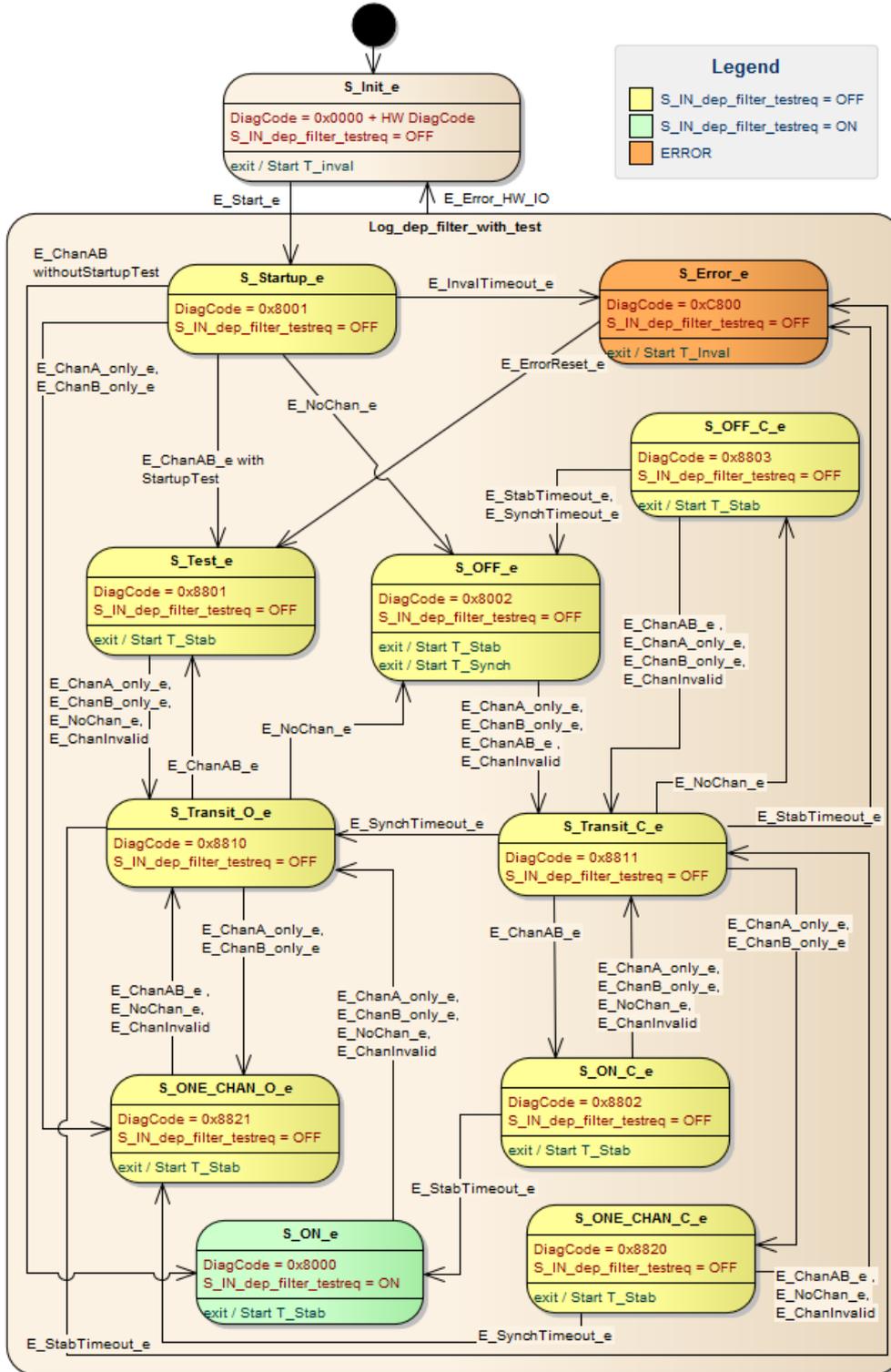
18218

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep_filter_testreq	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicherheitsfunktion AUS
			TRUE	Sicherheitsfunktion EIN

Zustandsdiagramm (State machine)

595

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19780

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8802	S_ON_C	beide Kanäle = TRUE, Prüfung der eingestellten Timer	OFF
0x8803	S_OFF_C	beide Kanäle = FALSE, Prüfung der eingestellten Timer	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_ASi_dependent_filter_ntestreq

18222

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_dependent_filter_ntestreq

Beschreibung

18221

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_dependent_filter_ntestreq erfasst die Schaltzustände der beiden Signalkanäle/Kontakte des AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Kontakte/Kanäle werden beim Einschalten entprellt: Signalwechsel, die kürzer als die eingestellte Stabilisierungszeit anliegen, führen nicht zu einem Zustandswechsel. (StabilisingTime)
- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird bei Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Wenn beim Ausschaltvorgang die beiden Kontakte/Kanäle innerhalb der einstellbaren Toleranzzeit wieder den Wert TRUE einnehmen, dann wird sofort wieder eingeschaltet. Nach Ablauf der Toleranzzeit müssen die beiden Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (ToleranceTime)
- Wenn nach dem Start die beiden Kontakte/Kanäle nicht innerhalb einer fest eingestellten Zeit einen gültigen Signalzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140 ms)
- Wenn die beiden Signale beim einem Umschaltvorgang länger ungültig als die eingestellte Stabilisierungszeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (StabilisingTime)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Um das logische Gerät aus dem verriegelten Fehlerzustand zurückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlASi_InSlave** (→ S. [191](#))

Parameterdaten

18220

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden
StabilisingTime	WORD	max. Prelldauer	250 ... 10000	250 Millisekunden * ... 10000 Millisekunden
ToleranceTime	WORD	Toleranzzeit für Abschaltung	250 ... 1000	250 Millisekunden * ... 1000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

18223

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep_filter_ntest	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicherheitsfunktion AUS
			TRUE	Sicherheitsfunktion EIN

Diagnose-Informationen

19783

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8802	S_ON_C	beide Kanäle = TRUE, Prüfung der eingestellten Timer	OFF
0x8803	S_OFF_C	beide Kanäle = FALSE, Prüfung der eingestellten Timer	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_ASi_dependent_filter_nshutdown

18227

Gerätetyp:	AS-i Slave (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_dependent_filter_nshutdown

Beschreibung

18226

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_dependent_filter_nshutdown erfasst die Schaltzustände der beiden Kontakte/Kanäle des AS-i Slaves (ASi_SlaveAdr, ASi_Master). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Kontakte/Kanäle werden beim Einschalten entprellt: Signalwechsel, die kürzer als die eingestellte Stabilisierungszeit anliegen, führen nicht zu einem Zustandswechsel. (StabilisingTime)
- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird bei Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Wenn beim Ausschaltvorgang die beiden Kontakte/Kanäle innerhalb der einstellbaren Toleranzzeit wieder den Wert TRUE einnehmen, dann wird sofort wieder eingeschaltet. Nach Ablauf der Toleranzzeit müssen die beiden Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (ToleranceTime)
- Wenn nach dem Start die beiden Kontakte/Kanäle nicht innerhalb einer fest eingestellten Zeit einen gültigen Signalzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140 ms)
- Wenn die beiden Signale beim einem Umschaltvorgang länger ungültig als die eingestellte Stabilisierungszeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (StabilisingTime)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Um das logische Gerät aus dem verriegelten Fehlerzustand zurückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlASi_InSlave** (→ S. [191](#))



Bei Verwendung dieses logischen Geräts verlängert sich die Prozesssicherheitszeit (PST) des Systems um eine Zykluszeit der fehlersicheren SPS!

- ▶ Hinweise zur Prozesssicherheitszeit beachten (→ Gerätehandbuch, **Prozesssicherheitszeit!**)
- ▶ Eingestellte Zykluszeit der fehlersicheren SPS beachten (→ **Zykluszeit der fehlersicheren SPS einstellen** (→ S. [101](#)))!

Parameterdaten

18225

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden
StabilisingTime	WORD	max. Prelldauer	250 ... 10000	250 Millisekunden * ... 10000 Millisekunden
ToleranceTime	WORD	Toleranzzeit für Abschaltung	250 ... 1000	250 Millisekunden * ... 1000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

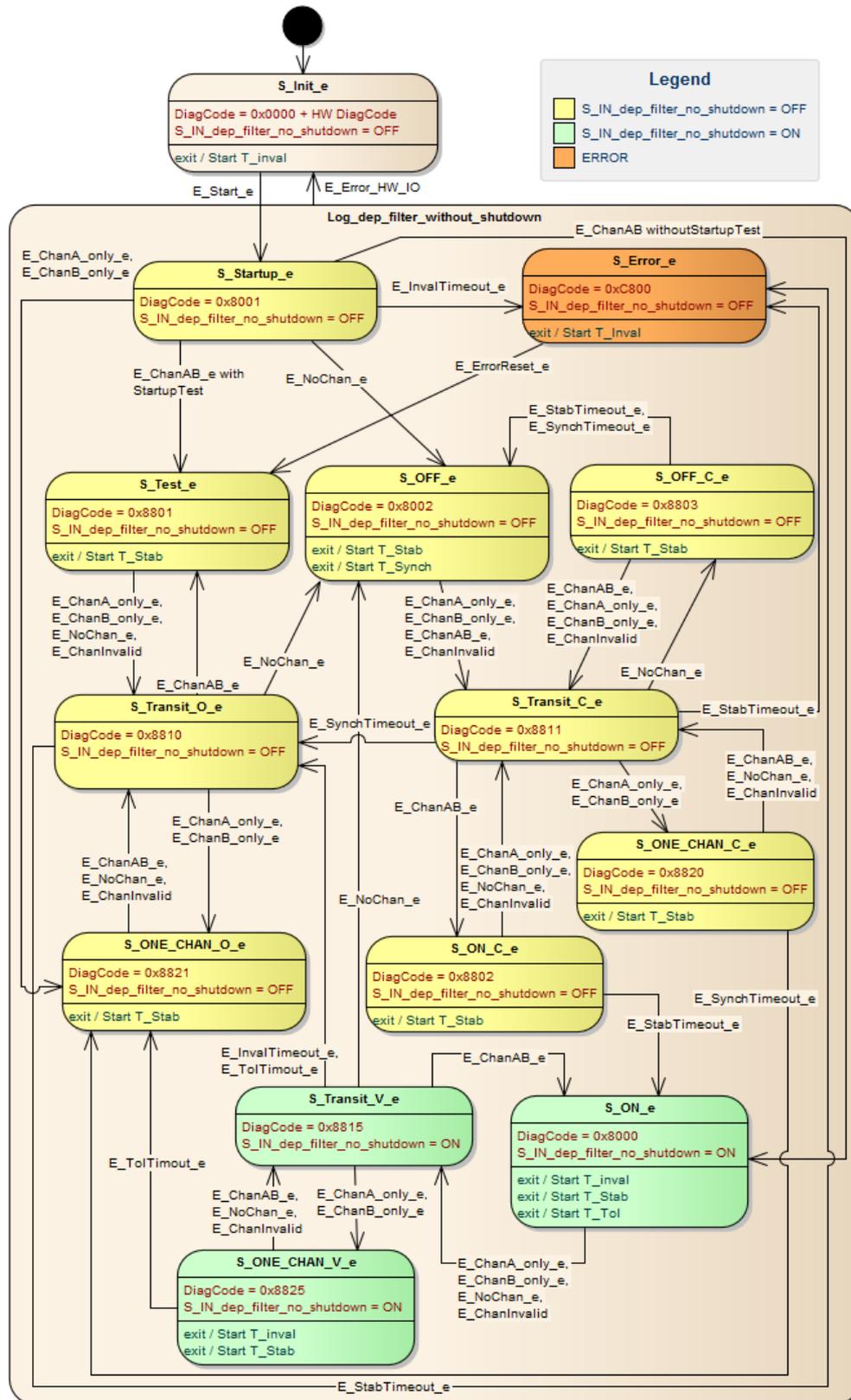
18228

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep_nshutdown	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicherheitsfunktion AUS
			TRUE	Sicherheitsfunktion EIN

Zustandsdiagramm (State machine)

18931

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19786

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8802	S_ON_C	beide Kanäle = TRUE, Prüfung der eingestellten Timer	OFF
0x8803	S_OFF_C	beide Kanäle = FALSE, Prüfung der eingestellten Timer	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8815	S_Transit	Übergangszustand	ON
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0x8825	S_ONE_CHAN_V	1 Kanal = TRUE; Toleranzzeit noch nicht abgelaufen	ON
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_OUTcontrol_ASi

18231

Gerätetyp: AS-i Slave (Ausgang)
Bibliothek: sf_logicalinterfaces.library
Symbol in CODESYS:  SF_OUTcontrol_ASi

Beschreibung

18230

Logisches Gerät für einen sicheren AS-i Control-Slaves (ASi_SlaveAdr, ASi_Master)

SF_OUTControl_ASi erzeugt aus dem konfigurierten Prozessdatenwert ein sicheres Ausgangssignal (S_OUT). Das sichere Ausgangssignal steuert den AS-i Control-Slaves, der eine sichere Code-Folge generiert, mit der einer oder mehrere sichere AS-i Ausgangs-Slaves freigegeben werden können.

Der Programmierer kann folgende Funktionen aktivieren:

- **DisableLogging:**
 Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.

Parameterdaten

18229

Parameter	Datentyp	Beschreibung	Mögliche Werte	
ASi_SlaveAdr	INT	Adresse des AS-i Slaves	1 ... 31	Slave-Adresse 1 ... Slave-Adresse 31
ASi_Master	INT	AS-i Master, an den der AS-i Slave angeschlossen ist	1	AS-i Master 1
			2	AS-i Master 2
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)

* ... Standardwert (Default)

Prozessdaten

18232

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_OUT	SAFEBOOL	Freigabe des sicheren AS-i Ausgangs-Slaves	FALSE	Sicherheitsfunktion AUS
			TRUE	Sicherheitsfunktion EIN

10.4.3 Logische Geräte für die lokale E/A-Schnittstelle

Inhalt	
SF_IN_local_forced	243
SF_IN_local_independent	247
SF_IN_local_conditionally_dependent	251
SF_IN_local_dependent	255
SF_IN_local_dependent_filter_testreq	259
SF_IN_local_dependent_filter_ntestreq	263
SF_IN_local_dependent_filter_nshutdown	267
SF_OUT_local_single	272
SF_OUT_local_dual	273
SF_OUT_local_testpulse	274

19795

SF_IN_local_forced

18279

Schnittstellen-Typ:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	➡ SF_IN_local_forced

Beschreibung

18278

Logisches Gerät für sichere Sensoren mit 2 zwangsgeführten Kontakten/Kanälen

SF_IN_local_forced erfasst die beiden Signale eines sicheren Sensors an der lokalen E/A-Schnittstelle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird beim Ein- und Ausschalten überwacht: Wenn beide Signalkanäle nicht innerhalb der fest eingestellten Zeit den gleichen Schaltzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 100 ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Um das logische Gerät aus dem verriegelten Fehlerzustand rückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs** (→ S. [198](#))

Parameterdaten

18277

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON

* ... Standardwert (Default)

Prozessdaten

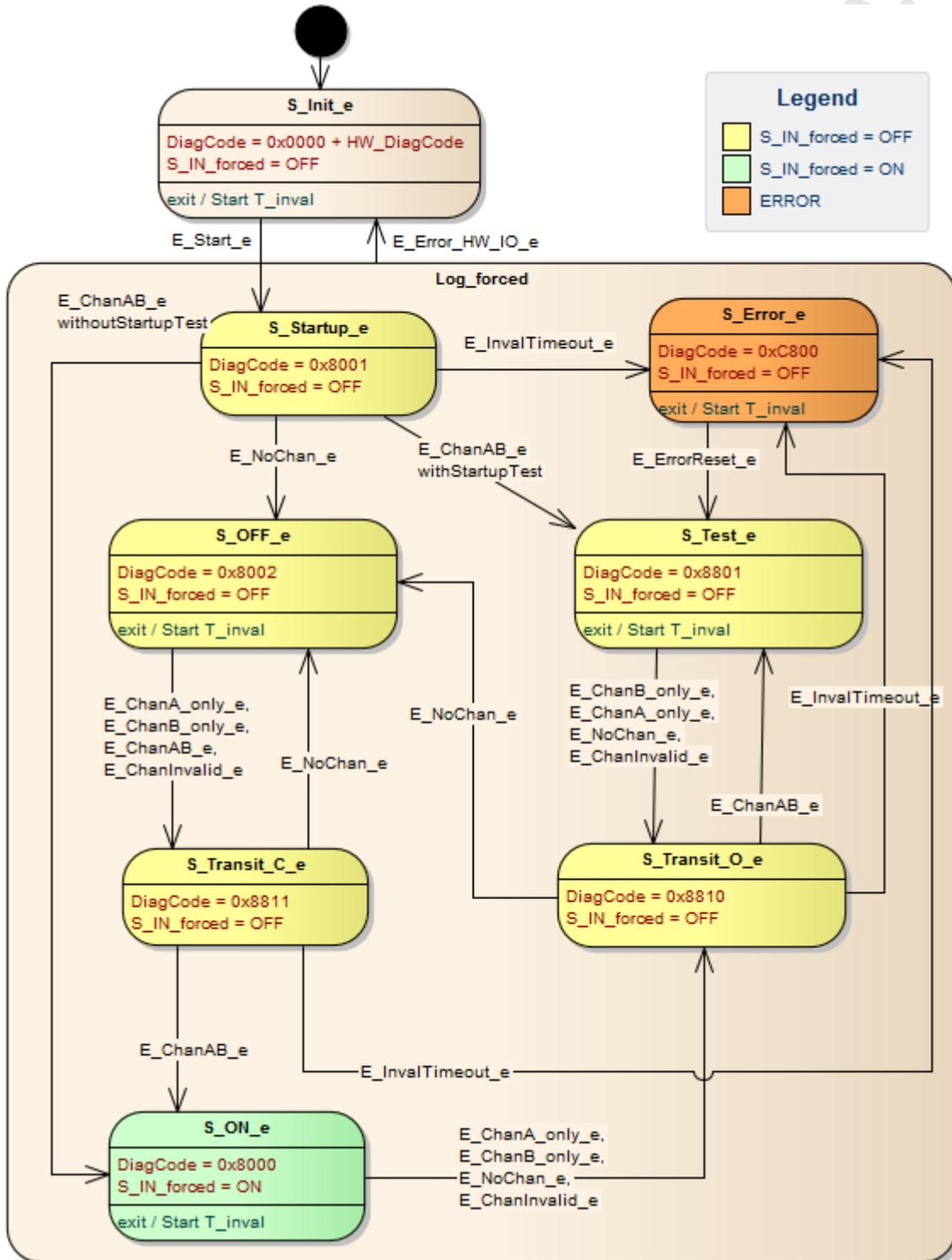
18276

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_forced	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS (OFF)
			TRUE	Sicheres EIN (ON)

Zustandsdiagramm (State machine)

20685

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19760

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_local_independent

18297

Interface-Typ:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_io.library
Symbol in CODESYS:	➡ SF_IN_local_independent

Beschreibung

18298

Logisches Gerät für einen sicheren Sensor mit 2 voneinander unabhängig wirkenden Kontakten/Kanälen

SF_IN_local_independent erfasst die Schaltzustände von 2 Eingangskanälen der lokalen E/A-Schnittstelle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Gleichzeitigkeit der Signalwechsel wird nicht überwacht.
- Wenn der Signalzustand der beiden Kanäle länger ungültig ist als die fest eingestellte Zeit, dann müssen beide Signalkanäle gleichzeitig den Zustand FALSE einnehmen, bevor wieder eingeschaltet werden kann. (InvalidTime = konst. = 140ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Um die Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs**
(→ S. [198](#))

Parameterdaten

18299

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON

* ... Standardwert (Default)

Prozessdaten

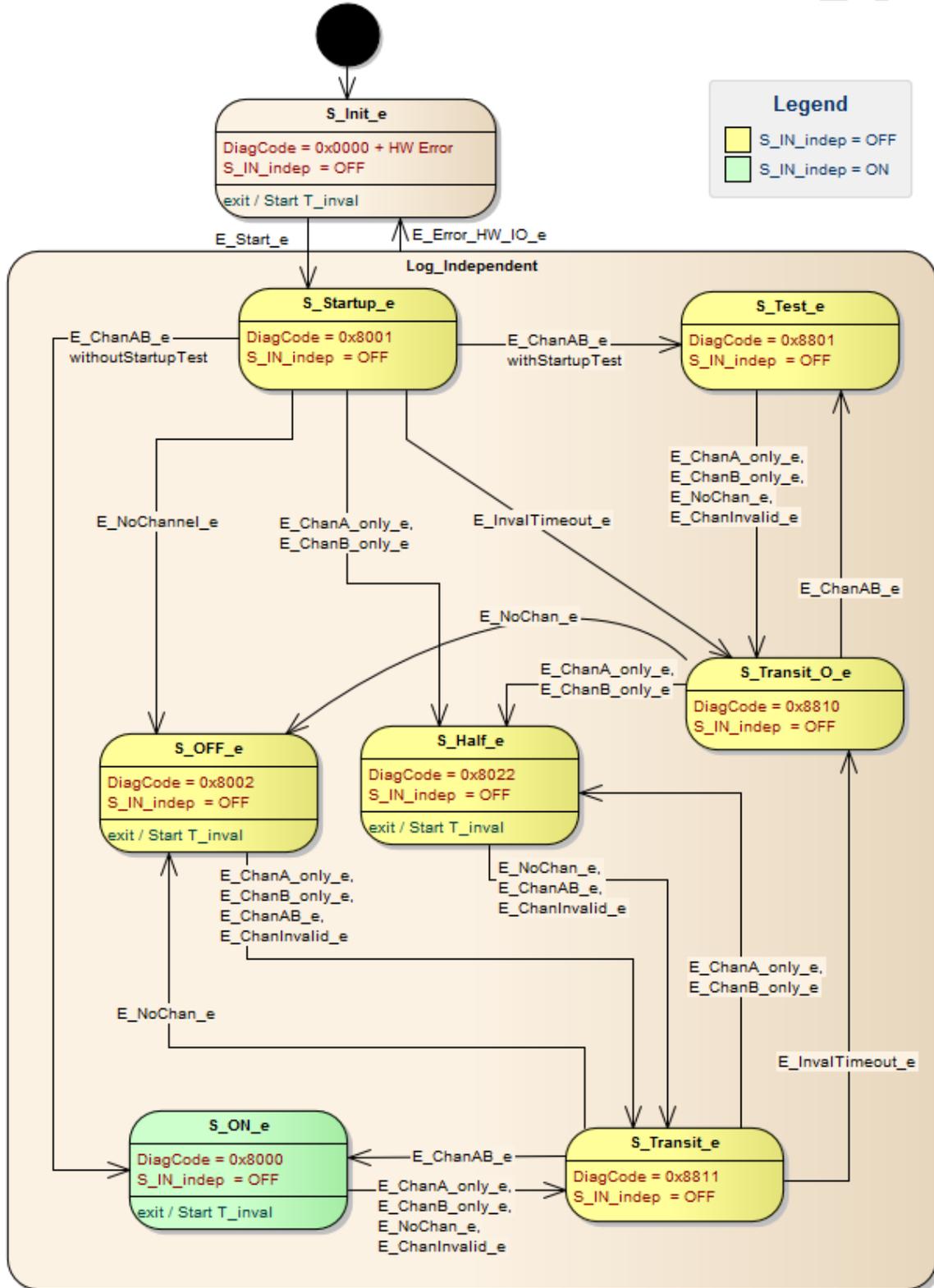
18300

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_indep	SAFEBOOL	Status der Sicherheitsfunktion	FALSE	Sicheres AUS (OFF) oder Fehler
			TRUE	Sicheres EIN (ON) und kein Fehler

Zustandsdiagramm (State machine)

20687

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19769

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wider (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8822	S_Half	1 Kanal = TRUE	OFF

SF_IN_local_conditionally_dependent

18287

Schnittstellen-Typ:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	➡ SF_IN_local_conditionally_dependent

Beschreibung

18290

Logisches Gerät für einen sicheren Sensor mit 2 voneinander bedingt abhängig wirkenden Kontakten/Kanälen

SF_IN_local_conditionally_dependent erfasst die Schaltzustände der beiden Eingangskanäle der lokalen E/A-Schnittstelle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Reihenfolge der Signalwechsel der beiden Kontakte/Kanäle wird beim Ein- und Ausschalten überwacht: Wenn die Reihenfolge der Signalwechsel abweicht von der eingestellten Reihenfolge, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (Chan_B_first)
- Wenn beide Signalkanäle nicht innerhalb der fest eingestellten Zeit den gleichen Schaltzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Um das logische Gerät aus dem verriegelten Fehlerzustand rückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs** (→ S. [198](#))

Parameterdaten

18288

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
Chan_B_first	BOOL	Gewünschte Reihenfolge der Signalzustandsänderungen	FALSE	Kanal A = TRUE vor Kanal B = TRUE
			TRUE	Kanal B = TRUE vor Kanal A = TRUE

* ... Standardwert (Default)

Prozessdaten

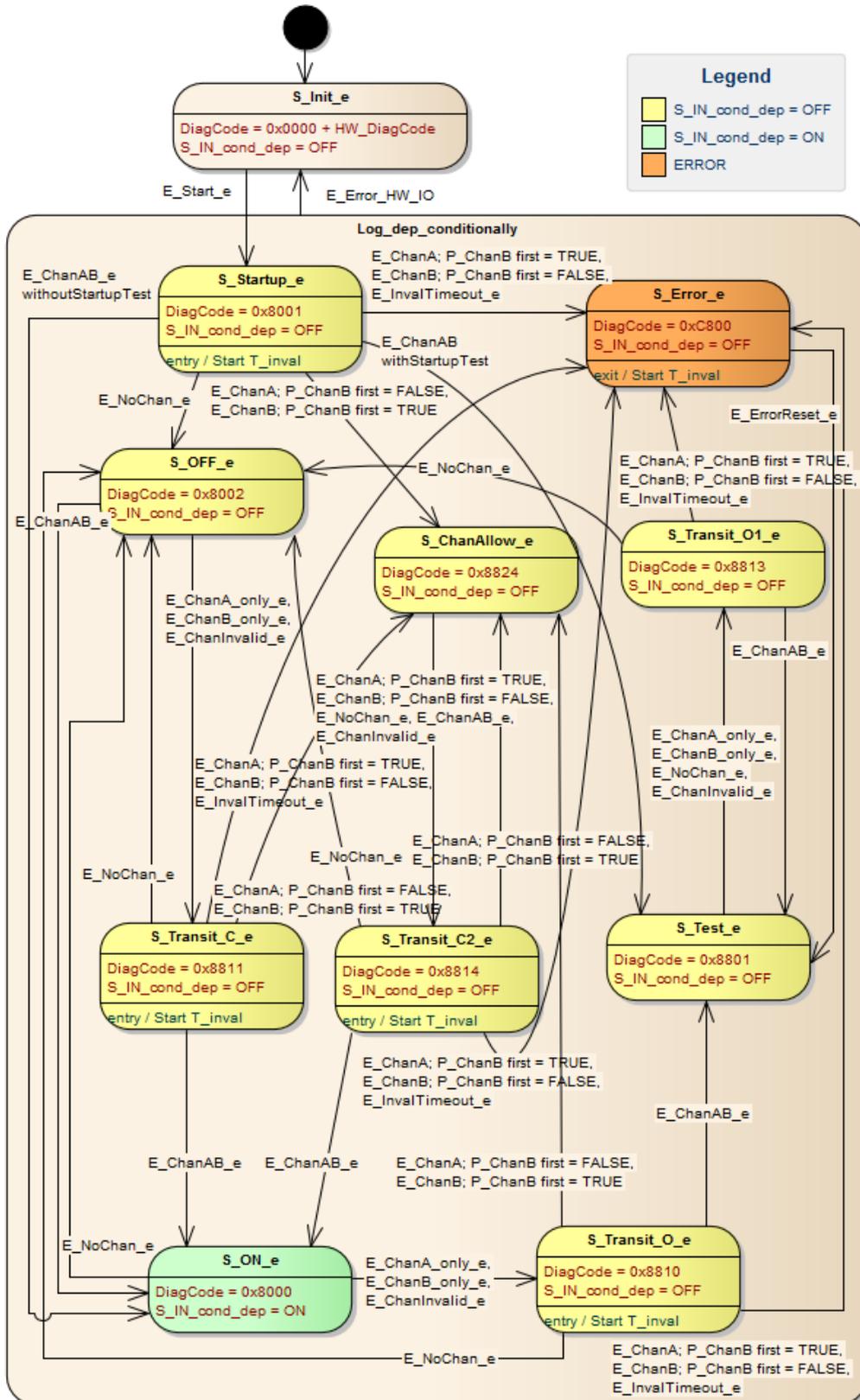
18289

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_cond_dep	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS (OFF) oder Fehlerzustand
			TRUE	Sicheres EIN (ON) und kein Fehlerzustand

Zustandsdiagramm (State machine)

20694

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19776

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8813	S_Transit	Übergangszustand	OFF
0x8814	S_Transit	Übergangszustand	OFF
0x8824	S_ChanAllow	1 Kanal = TRUE, Signal-Reihenfolge o.K.	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_local_dependent

18286

Schnittstellen-Typ:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	➡ SF_IN_local_dependent

Beschreibung

18285

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_local_dependent erfasst die Schaltzustände der beiden Eingangskanäle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird beim Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Wenn der Signalzustand der beiden Signalkanäle länger ungültig ist als die fest eingestellte Zeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Um das logische Gerät aus dem verriegelten Fehlerzustand rückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs** (→ S. [198](#))

Parameterdaten

18284

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

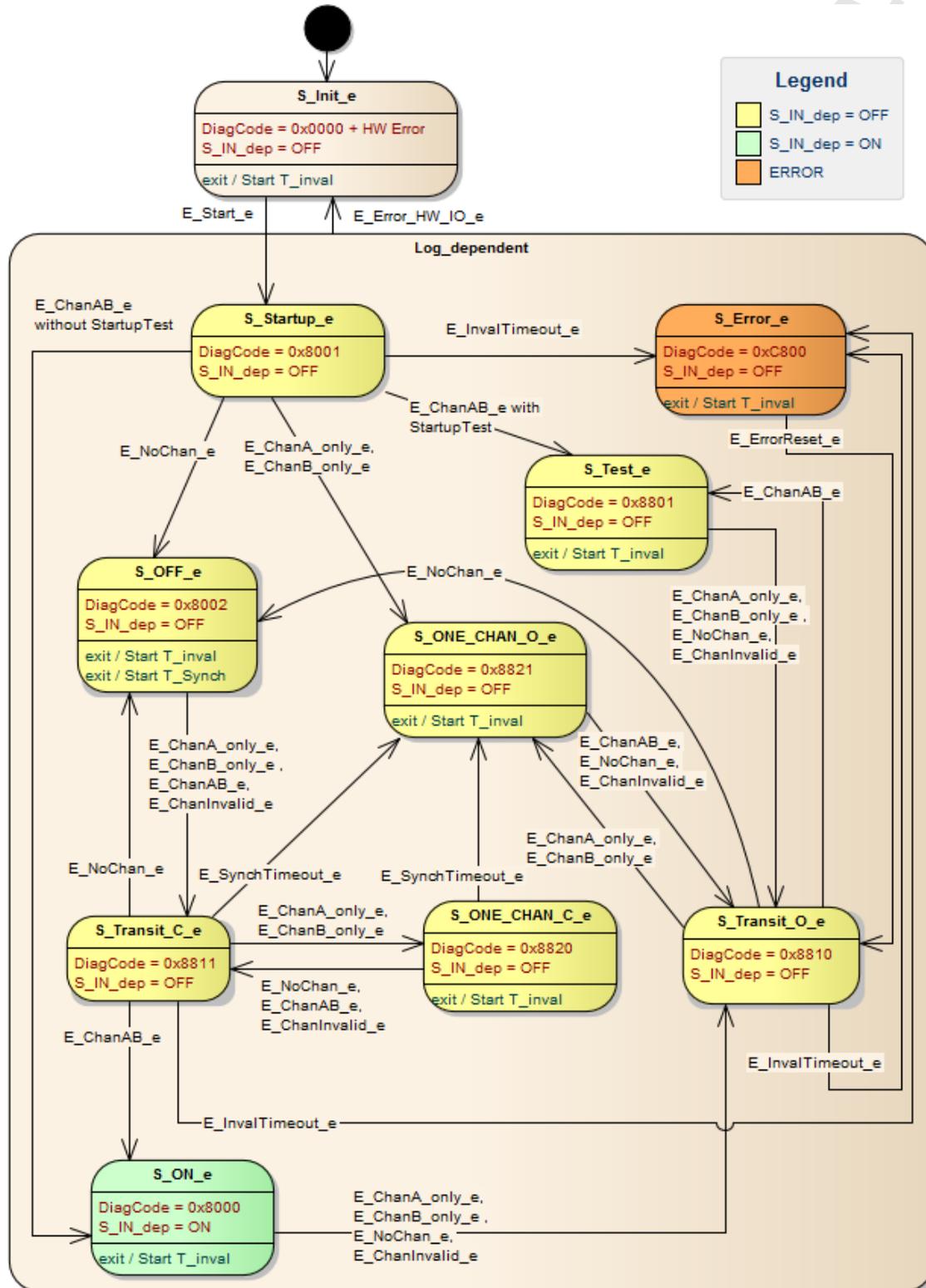
18283

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS (OFF)
			TRUE	Sicheres EIN (ON)

Zustandsdiagramm (State machine)

20689

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19773

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_local_dependent_filter_testreq

18301

Schnittstellen-Typ:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	➔ SF_IN_local_dependent_filter_testreq

Beschreibung

18302

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_local_dependent_filter_w_testreq erfasst die Schaltzustände der beiden Eingangskanäle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Kontakte/Kanäle werden beim Einschalten entprellt: Signalwechsel, die kürzer als die eingestellte Stabilisierungszeit anliegen, führen nicht zu einem Zustandswechsel. (StabilisingTime)
- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird bei Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Vor dem Wiedereinschalten müssen beide Kontakte/Kanäle gleichzeitig FALSE sein.
- Wenn nach dem Start die beiden Kontakte/Kanäle nicht innerhalb einer fest eingestellten Zeit einen gültigen Signalzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140 ms)
- Wenn die beiden Signale beim einem Umschaltvorgang länger ungültig als die eingestellte Stabilisierungszeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (StabilisingTime)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Um das logische Gerät aus dem verriegelten Fehlerzustand rückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs** (→ S. [198](#))

Parameterdaten

18303

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden
StabilisingTime	WORD	max. Prelldauer	250 ... 10000	250 Millisekunden * ... 10000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

18304

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep_filter_testreq	SAFEBOOL	Sicheres Prozesssignal	FALSE =	Sicheres AUS (OFF)
			TRUE =	Sicheres EIN (ON)

Diagnose-Informationen

19780

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8802	S_ON_C	beide Kanäle = TRUE, Prüfung der eingestellten Timer	OFF
0x8803	S_OFF_C	beide Kanäle = FALSE, Prüfung der eingestellten Timer	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_local_dependent_filter_ntestreq

18305

Schnittstellen-Typ:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_local_dependent_filter_ntestreq

Beschreibung

18306

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_local_dependent_filter_w_testreq erfasst die Schaltzustände der beiden Eingangskanäle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Kontakte/Kanäle werden beim Einschalten entprellt: Signalwechsel, die kürzer als die eingestellte Stabilisierungszeit anliegen, führen nicht zu einem Zustandswechsel. (StabilisingTime)
- Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird bei Einschalten überwacht: Wenn beim Ausschaltvorgang die beiden Kontakte/Kanäle innerhalb der einstellbaren Toleranzzeit wieder den Wert TRUE einnehmen, dann wird sofort wieder eingeschaltet. Nach Ablauf der Toleranzzeit müssen die beiden Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (ToleranceTime)
- Wenn nach dem Start die beiden Kontakte/Kanäle nicht innerhalb einer fest eingestellten Zeit einen gültigen Signalzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140 ms)
- Wenn die beiden Signale beim einem Umschaltvorgang länger ungültig als die eingestellte Stabilisierungszeit, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (StabilisingTime)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Um das logische Gerät aus dem verriegelten Fehlerzustand rückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs** (→ S. [198](#))

Parameterdaten

18307

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden
StabilisingTime	WORD	max. Prelldauer	250 ... 10000	250 Millisekunden * ... 10000 Millisekunden
ToleranceTime	WORD	Toleranzzeit für Abschaltung	250 ... 1000	250 Millisekunden * ... 1000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

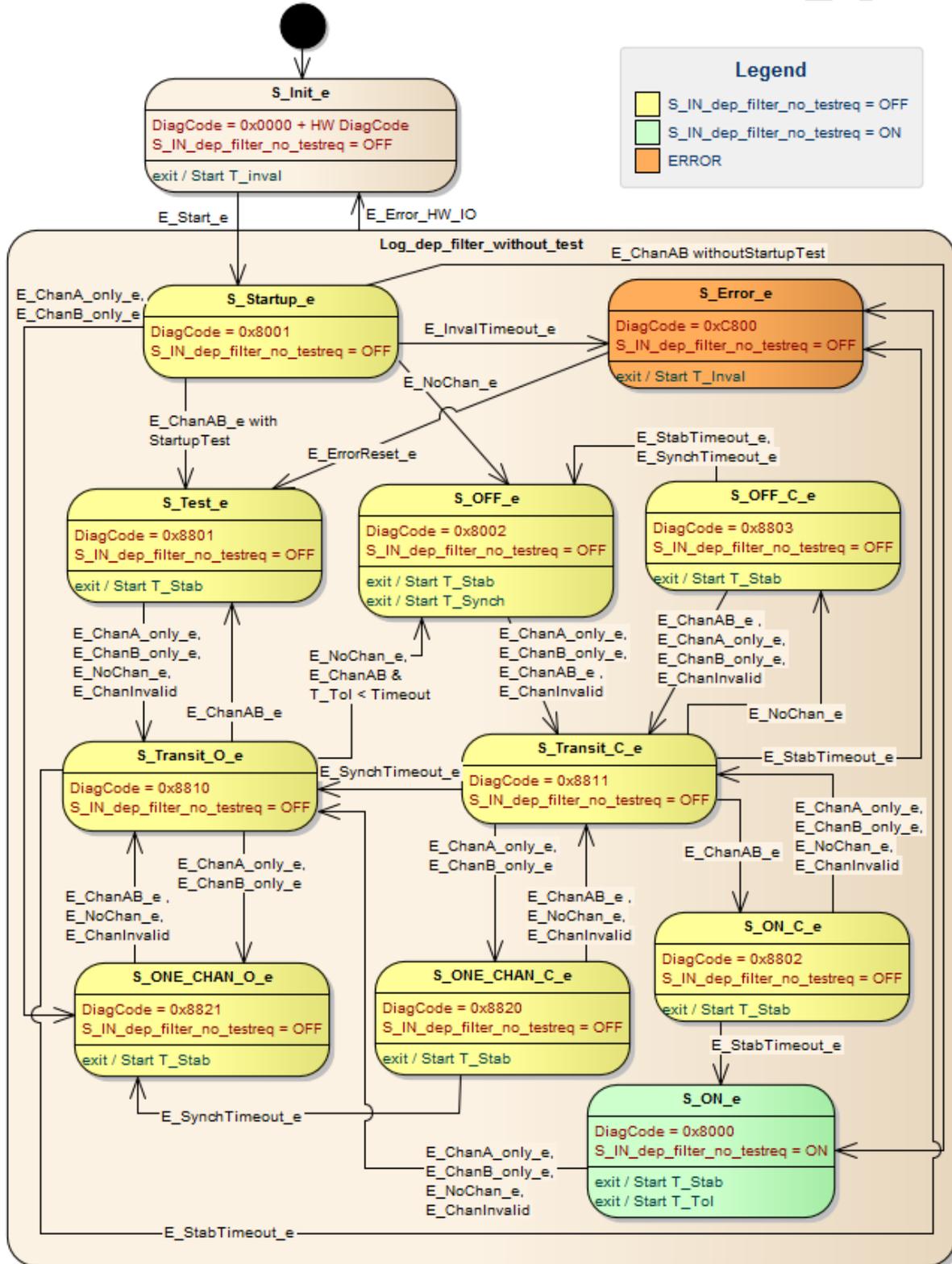
18308

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep_filter_no_testreq	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS (OFF)
			TRUE	Sicheres EIN (ON)

Zustandsdiagramm (State machine)

20692

Das Zustandsdiagramm zeigt die logische Signalauswertung des logischen Gerats:



Diagnose-Informationen

19783

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8802	S_ON_C	beide Kanäle = TRUE, Prüfung der eingestellten Timer	OFF
0x8803	S_OFF_C	beide Kanäle = FALSE, Prüfung der eingestellten Timer	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_IN_local_dependent_filter_nshutdown

18312

Gerätetyp:	Lokale E/A-Schnittstelle (Eingang)
Bibliothek:	sf_logicalinterfaces.library
Symbol in CODESYS:	➡ SF_IN_local_dependent_filter_nshutdown

Beschreibung

18311

Logisches Gerät für einen sicheren Sensor mit 2 voneinander abhängig wirkenden Kontakten/Kanälen SF_IN_local_dependent_filter_nshutdown erfasst die Schaltzustände der beiden Eingangskanäle der lokalen E/A-Schnittstelle (IN_Channel_A, IN_Channel_B). Das logische Gerät bildet aus den beiden Werten die logische AND-Verknüpfung und gibt das Resultat als sicheres Steuerungssignal aus. Bei der logischen Signalauswertung gelten folgende Nebenbedingungen:

- Die Kontakte/Kanäle werden beim Einschalten entprellt: Signalwechsel, die kürzer als die eingestellte Stabilisierungszeit anliegen, führen nicht zu einem Zustandswechsel. (StabilisingTime)
- Die Gleichzeitigkeit der Signalwechsel der beiden Kontakte/Kanäle wird bei Einschalten überwacht: Wenn die Signalzustände länger ungleich sind als die eingestellte Synchronisationszeit, dann müssen beide Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (SynchTime)
- Wenn beim Ausschaltvorgang die beiden Kontakte/Kanäle innerhalb der einstellbaren Toleranzzeit wieder den Wert TRUE einnehmen, dann wird sofort wieder eingeschaltet. Nach Ablauf der Toleranzzeit müssen die beiden Kontakte/Kanäle gleichzeitig den Wert FALSE einnehmen, bevor wieder eingeschaltet werden kann. (ToleranceTime)
- Wenn nach dem Start die beiden Kontakte/Kanäle nicht innerhalb einer fest eingestellten Zeit einen gültigen Signalzustand einnehmen, dann wechselt das logische Gerät in den verriegelten Fehlerzustand. (InvalidTime = konst. = 140ms)

Der Programmierer kann optional folgende Funktionen aktivieren:

- **EnableIN_Test:**
Ein an einem lokalen Ausgangskanal der lokalen E/A-Schnittstelle erzeugtes Testsignal zur Querschlusserkennung wird ausgewertet.
- **StartupTest:**
Beide Signalkanäle müssen gleichzeitig den Zustand FALSE einnehmen, bevor das logische Gerät mit der Signalauswertung beginnen kann.
- **DisableLogging:**
Nur Fehlerereignisse werden an das Online Support Center (OSC) der AC4S-Bedienoberfläche gesendet.



Ist die Querschlusserkennung für 2-kanalige sichere Geräte an der lokalen E/A-Schnittstelle aktiviert (EnableIN_Test = TRUE), gilt folgendes:
Signalwechsel an den Eingängen (IN_Channel_A, IN_Channel_B) müssen länger anliegen als die Zyklusdauer des Testimpulses (3 SPS-Zyklen). Andernfalls geht das logische Gerät in den verriegelten Fehlerzustand.



Bei Verwendung dieses logischen Geräts verlängert sich die Prozesssicherheitszeit (PST) des Systems um eine Zykluszeit der fehlersicheren SPS!

- ▶ Hinweise zur Prozesssicherheitszeit beachten (→ Gerätehandbuch, **Prozesssicherheitszeit!**)!
- ▶ Eingestellte Zykluszeit der fehlersicheren SPS beachten (→ **Zykluszeit der fehlersicheren SPS einstellen** (→ S. [101](#)))!



Um das logische Gerät aus dem verriegelten Fehlerzustand rückzusetzen und Diagnose-Informationen des logischen Geräts auszulesen: → **CtrlLocalInputs** (→ S. [198](#))

Parameterdaten

18310

Parameter	Datentyp	Beschreibung	Mögliche Werte	
IN_Channel_A	INT	Lokaler Eingang, an den der Sensorkanal A angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
IN_Channel_B	INT	Lokaler Eingang, an den der Sensorkanal B angeschlossen ist.	1 ... 8	lokaler Eingang IN1 ... lokaler Eingang IN8
EnableIN_Test	BOOL	Auswertung eines Testsignals, das durch einen lokalen Ausgang erzeugt wurde.	FALSE	Testsignal wird ignoriert *
			TRUE	Testsignal wird ausgewertet
DisableLogging	BOOL	Übertragung der Ereignis-Meldungen an OSC des Geräts	FALSE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: * <ul style="list-style-type: none"> ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Folgende Zustände des Zustandsdiagramms werden an das OSC übertragen: <ul style="list-style-type: none"> ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Ausführung des Startup-Test steuern	FALSE	kein Startup-Test *
			TRUE	Startup-Test wird ausgeführt; Slave wartet auf die Startup-Sequenz OFF ⇒ ON
SynchTime	WORD	Synchronisationszeit	0 250 ... 500 ... 60000	unendlich 250 Millisekunden ... 500 Millisekunden * ... 60000 Millisekunden
StabilisingTime	WORD	max. Prelldauer	250 ... 10000	250 Millisekunden * ... 10000 Millisekunden
ToleranceTime	WORD	Toleranzzeit für Abschaltung	250 ... 1000	250 Millisekunden * ... 1000 Millisekunden

* ... Standardwert (Default)

Prozessdaten

18309

Parameter	Datentyp	Beschreibung	Mögliche Werte	
S_IN_dep_ntshutdown	SAFEBOOL	Sicheres Prozesssignal	FALSE	Sicheres AUS (OFF) oder Fehler
			TRUE	Sicheres EIN (ON) und kein Fehler

Diagnose-Informationen

19786

Das logische Gerät erzeugt Diagnose-Codes. Sie spiegeln den jeweils aktiven internen Zustand der Auswertelogik wieder (State Machine). Die Diagnose-Codes können mit folgendem FB ausgelesen werden: → **CtrlASi_InSlave** (→ S. [191](#))

Folgende Diagnose-Codes existieren:

DiagCode	Name	Bedeutung	Ausgang
0x0000	S_Init	Initialisierungszustand	OFF
0x8001	S_Startup	Startzustand	OFF
0x8002	S_OFF	Sicheres AUS	OFF
0x8000	S_ON	Sicheres EIN	ON
0x8801	S_Test	Zustand zum Durchlaufen des Startup-Tests	OFF
0x8802	S_ON_C	beide Kanäle = TRUE, Prüfung der eingestellten Timer	OFF
0x8803	S_OFF_C	beide Kanäle = FALSE, Prüfung der eingestellten Timer	OFF
0x8810	S_Transit	Übergangszustand	OFF
0x8811	S_Transit	Übergangszustand	OFF
0x8815	S_Transit	Übergangszustand	ON
0x8820	S_ONE_CHAN_C	1 Kanal = TRUE, Prüfung der Synchronisationszeit	OFF
0x8821	S_ONE_CHAN_O	1 Kanal = TRUE, Synchronisationszeit abgelaufen	OFF
0x8825	S_ONE_CHAN_V	1 Kanal = TRUE; Toleranzzeit noch nicht abgelaufen	ON
0xC800	S_Error	Verriegelter Fehlerzustand	OFF

SF_OUT_local_single

18313

Schnittstellen-Typ: Lokale E/A-Schnittstelle (Ausgang)**Bibliothek:** sf_logicalinterfaces.library**Symbol in CODESYS:**  SF_OUT_local_single

Beschreibung

18314

Logisches Gerät für einen 1-kanaligen sicheren Aktuator an der lokalen E/A-Schnittstelle SF_OUT_local_single erfasst den am Prozessdaten-Signal anliegenden booleschen Wert (OUT_). Das logische Gerät generiert daraus ein sicheres boolesches Signal und gibt den Wert am konfigurierten Ausgangskanal der lokalen E/A-Schnittstelle aus (OUT_Channel).

Parameterdaten

18316

Parameter	Datentyp	Beschreibung	Mögliche Werte	
OUT_Channel	INT	Lokaler Ausgang, an dem der Aktuator angeschlossen ist.	1 ... 4	lokaler Ausgang OUT1 ... lokaler Ausgang OUT4
DisableLogging	reserviert - Einstellungen haben keinen Einfluss auf Funktion des logischen Geräts			

Prozessdaten

18315

Parameter	Datentyp	Beschreibung	Mögliche Werte	
OUT_	SAFEBOOL	Soll-Status des Ausganges / der Ausgänge	FALSE	Sicheres AUS (OFF)
			TRUE	Sicheres EIN (ON)

SF_OUT_local_dual

18321

Gerätetyp: Lokale E/A-Schnittstelle (Ausgang)

Bibliothek: sf_logicalinterfaces.library

Symbol in CODESYS:  SF_OUT_local_dual

Beschreibung

18323

Logisches Gerät für die Steuerung eines 2-kanaligen sicheren Aktuators an der lokalen E/A-Schnittstelle

SF_OUT_local_dual erfasst den am Prozessdaten-Signal anliegenden booleschen Wert (OUT_). Das logische Gerät generiert daraus ein sicheres boolesches Signal und gibt den Wert an den konfigurierten Ausgangskanälen der lokalen E/A-Schnittstelle aus (OUT_Channel_A, OUT_Channel_B).

Parameterdaten

18322

Parameter	Datentyp	Beschreibung	Mögliche Werte	
OUT_Channel_A	INT	Lokalen Ausgang, an dem der Aktuator kanal A angeschlossen ist.	1 ... 4	lokaler Ausgang OUT1 ... lokaler Ausgang OUT4
OUT_Channel_B	INT	Lokalen Ausgang, an dem der Aktuator kanal B angeschlossen ist.	1 ... 4	lokaler Ausgang OUT1 ... lokaler Ausgang OUT4
DisableLogging	reserviert - Einstellungen haben keinen Einfluss auf Funktion des logischen Geräts			

Prozessdaten

18324

Parameter	Datentyp	Beschreibung	Mögliche Werte	
OUT_	SAFEBOOL	Soll-Status des Ausgangs / der Ausgänge	FALSE	Sicheres AUS (OFF)
			TRUE	Sicheres EIN (ON)

SF_OUT_local_testpulse

18317

Geräte-Typ: Lokale E/A-Schnittstelle (Ausgang)

Bibliothek: sf_logicalinterfaces.library

Symbol in CODESYS:  SF_OUT_local_testpulse

Beschreibung

18318

Logisches Gerät für die Generierung eines Testimpulses an einem Ausgangskanal der lokalen E/A-Schnittstelle

SF_OUT_local_testpulse erzeugt einen Testimpuls und gibt diesen am konfigurierten Ausgangskanal der lokalen E/A-Schnittstelle aus (OUT_Channel). Dauer und Frequenz des Impulses sind fest eingestellt:

Eigenschaften Testimpuls	
Impulsdauer:	2 SPS-Zyklen
Impulsfrequenz:	12 SPS-Zyklen
Phasenverschiebung:	OUT1 ⇒ OUT2 ⇒ OUT3 ⇒ OUT4 ⇒ OUT1 = 3 SPS-Zyklen

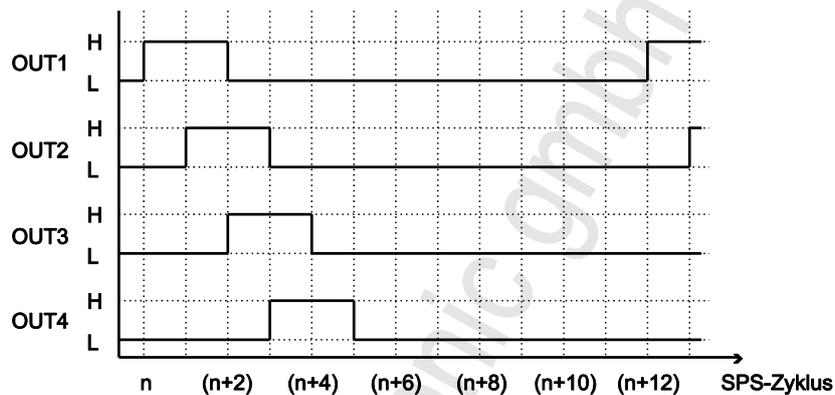


Abbildung: Signalverlauf und Phasenverschiebung des Testpulses

Parameterdaten

18319

Parameter	Datentyp	Beschreibung	Mögliche Werte	
OUT_Channel	INT	Lokaler Ausgang, an dem der Testimpuls ausgegeben werden soll	1 ... 4	lokaler Ausgang OUT1 ... lokaler Ausgang OUT4

10.5 Fehler-Codes: Sichere AS-i Slaves

18256

Kommunikations- und Konfigurationsfehler versetzen das logische Gerät in den Initialisierungszustand (S_INIT). Gleichzeitig wird an den Steuerungs-Funktionsbaustein (→ **CtrlASi_InSlave** (→ S. 191), → **CtrlASi_OutSlave** (→ S. 193)) ein Diagnose-Code übergeben, der die Fehlerursache beschreibt.

Folgende Diagnose-Codes sind verfügbar:

DiagCode	Fehlernamen (FB)	Fehlerbeschreibung
	Fehlermeldung (OSC)	
0x4001	ASi_SYS_ERROR_CFG_MASTERID_INVALID	Falscher Master im Steuerungs-FB oder logischen Gerät
	Wrong Master-Id in ASi-Control-FUB	
0x4002	ASi_SYS_ERROR_CFG_SLAVEADDR_INVALID	Falscher Slave im Steuerungs-FB oder logischen Gerät (Adresse 0 im Gerätebaum CODESYS provoziert Fehler beim Download)
	Wrong Slave Address in ASi-Control-FUB	
0x4003	ASi_SYS_ERROR_CFG_MULTIPLE_SLAVE	Fehlerhafte Projektierung durch zwei Slaves mit gleicher Adresse
	Try to config multiple Input Slaves with same Master-Id and Slave-Address	
0x4004	ASi_SYS_ERROR_CFG_MISMATCH_SLAVE	Fehlerhafte Projektierung durch Ein- und Ausgangs-Slave mit gleicher Adresse
	Try to config a Input and Output Slave at the same Master-Id and Slave-Address	
0x4005	ASi_SYS_ERROR_CFG_CANTADD_SLAVE	Neuer AS-i Eingangs-Slave konnte nicht zum Projekt hinzugefügt werden.
	Can't add a new ASi-Input-Slave in case of a error return from AS-i-Control	
0x4009	ASi_SYS_ERROR_DIAGCFG_NOTCONFIGUED	Logisches Gerät nicht konfiguriert
	Try to monitor a not configured ASi-Slave in a AS-i-Control-FUB	
0x400A	ASi_SYS_ERROR_DIAGCFG_HSI_REQUEST	Hilfssignale HSI1 und HSI2 wurden gleichzeitig aktiviert.
	At CtrlASiOut-FUB : HSI1 and HSI2 Request at the same Time. Don't activate HSI1 and HSI2 together.	
0x4010	ASi_SYS_OUTPUT_CANTADD	<ul style="list-style-type: none"> ▪ Fehlerhafte Projektierung durch zu viele Ausgangs-Control-Slaves im Projekt. ▪ Dieser Fehler kann von Fehler 0x4009 überdeckt werden.
	Can't add a new ASi-Output-Contol-Slave (ex. Max. count of ControlSlaves reached)	
0x5800	ASi_HW_SUB_STATE_SLAVE_MISSING_TESTREQUEST	Testanforderung erkannt.
	Forced Testrequest from AS-i-SlaveCheck; Waiting for manual Test of related AS-i-Input-Slave	
0x5820	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_TIMEOUT	Fehlercode von AS-i Control-Slave empfangen
	Errorcode received from AS-i-Control	
0x5821	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_A_B_CHANNEL_OPEN	Fehlercode von AS-i Control-Slave empfangen
	Errorcode received from AS-i-Control	
0x5822	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_A_CHANNEL_OPEN	Fehlercode von AS-i Control-Slave empfangen
	Errorcode received from AS-i-Control	
0x5823	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_B_CHANNEL_OPEN	Fehlercode von AS-i Control-Slave empfangen
	Errorcode received from AS-i-Control	

DiagCode	Fehlername (FB)	Fehlerbeschreibung
0x5824	ASi_HW_CONFIG_SLAVE_ERROR_CODESEQUENCE_INVALID	Fehlercode von AS-i Control-Slave empfangen
	Errorcode received from AS-i-Control	
0x5825	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_CODESEQUENCE_DOUBLY	<ul style="list-style-type: none"> ▪ 2 Slaves mit gleicher Codefolge im AS-i Netz ▪ Fehler 0x5825 und 0x5826 können gleichzeitig auftreten
	Errorcode received from AS-i-Control	
0x5826	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_CODESEQUENCE_RULES	<ul style="list-style-type: none"> ▪ Fehler beim Einlernen der Codefolge ▪ Fehler 0x5825 und 0x5826 können gleichzeitig auftreten
	Errorcode received from AS-i-Control	
0x5827	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_NO_CODESEQUENCE	<ul style="list-style-type: none"> ▪ Fehler beim Einlernen der Codefolge ▪ Am Steuerungs-FB wird dieser Fehler durch Fehler 0x5840 überdeckt.
	Errorcode received from AS-i-Control	
0x5830	ASi_HW_GLOBAL_COM_MISSING_BUS_CYCLE_EVENT	Alle Busfehler können lokale Fehler überdecken.
	Errorcode received from AS-i-Control	
0x5831	ASi_HW_GLOBAL_COM_ADDRESS_SEQUENCE_ERROR	Alle Busfehler können lokale Fehler überdecken.
	Errorcode received from AS-i-Control	
0x5832	ASi_HW_GLOBAL_COM_FAILURE_TO_LESS_SLAVES	Zu wenige AS-i Slaves am Bus
	Errorcode received from AS-i-Control	
0x5840	ASi_HW_ERROR_STATE_INIT	<ul style="list-style-type: none"> ▪ kein AS-i Slave am Bus ▪ Alle Busfehler können lokale Fehler überdecken. ▪ Dieser Fehler kann von Fehler 0x5820 überdeckt werden.
	AS-i Slave fehlt	

10.6 Fehler-Codes: Lokale E/A-Schnittstelle

18259

Kommunikations- und Konfigurationsfehler versetzen das logische Gerät in den Initialisierungszustand (S_INIT). Gleichzeitig wird an den Steuerungs-Funktionsbaustein ein Diagnose-Code übergeben, der die Fehlerursache beschreibt (→ **CtrlLocalInputs** (→ S. 198)).

Folgende Diagnose-Codes sind verfügbar:

DiagCode	Fehlername (FB)	Fehlerbeschreibung
	Fehlermeldung (OSC)	
0x4020	LIO_SYS_ERROR_CFG_CHANNEL_A_INVALID	Falsche ID für Kanal A (Channel A = 0 oder >8)
	Wrong Id for Channel A in LIO-CtrlFUB	
0x4021	LIO_SYS_ERROR_CFG_CHANNEL_B_INVALID	Falsche ID für Kanal B (Channel A = 0 oder >8)
	Wrong Id for Channel B in LIO-CtrlFUB	
0x4022	LIO_SYS_ERROR_CFG_MULTIPLE_CHANNEL	Mehrere logische Geräte mit gleicher Kanalzuweisung
	Try to config multiple Local IOs with same Channel	
0x4023	LIO_SYS_ERROR_CFG_DOUBLE_CHANNEL	Gleiche ID für beide Kanäle eines zweikanaligen logischen Geräts verwendet
	Try to configure both channels of a dual channel IO with the same channel number (Channel A equal Channel B)	
0x4024	LIO_SYS_ERROR_CFG_NOT_SUPPORTED	nicht unterstütztes Ausgangs-Gerät
	Actually not supported output device (e.g. a DEV_SF_OUT_Local_dual)	
0x4025	LIO_SYS_ERROR_CFG_EXHAUSTED	Maximale Zahl der konfigurierbaren Eingänge erreicht
	Max. count of configurable Inputs reached	
0x4026	LIO_SYS_ERROR_INPUT_CANTADD	Hinzufügen eines lokalen Eingangs fehlgeschlagen
	Can't add a new Local Input in case of a error return from Local-IO-Control	
0x4031	LIO_SYS_ERROR_DIAGCFG_NOTCONFIGURED	nicht konfigurierter lokaler Eingang Fehler kann von Fehler 0x4031 überdeckt werden
	Try to monitor a not configured Local Input in a LIO-Control-FUB	
0x4032	LIO_SYS_ERROR_DIAGCFG_CONFIGMISMATCH	Falsche Zuordnung der Eingangskanäle in einem Steuerungs-FB ODER Ein als fehlersicher konfigurierter Eingang wird mit FB GetLocalInput abgefragt
	Try to config the same Input at single and dual	
0xE100	LIO_HW_ERROR_WRONG_TEST SIGNAL_e	Querschlusserkennung
	Detect unexpected Testsignal in a local Input channel (a reason could be a short cut)	
0xE101	LIO_HW_ERROR_NO_TEST SIGNAL_e	Querschlusserkennung
	Can't detect a expected Testsignal (no connection)	
0xE102	LIO_HW_INPUT_TEST_IN_PROCESS_e	Querschlusserkennung
	Wait for Test after both input signals switched ON.	

11 Index

A

ACnnnn_SysCmd.....	183
Allgemeine Sicherheitshinweise.....	9
Analoge Ausgangsdaten.....	55
Analoge Ein- und Ausgangsdaten.....	53
Änderungshistorie.....	8
Anhang.....	116
AS-i Master konfigurieren.....	59
AS-i Netzwerk verwalten.....	60
AS-i Slaves konfigurieren.....	59
ASI_ADDRESS_MODE (ENUM).....	173
ASI_ANA_IO (STRUCT).....	179
ASI_ANALOG_IN (STRUCT).....	179
ASI_ANALOG_IN_FLAGS (STRUCT).....	180
ASI_ANALOG_OUT (STRUCT).....	180
ASI_ANALOG_OUT_FLAGS (STRUCT).....	181
ASI_BIN_IN (STRUCT).....	178
ASI_BIN_IO (STRUCT).....	178
ASI_BIN_OUT (STRUCT).....	178
ASI_DATA (STRUCT).....	175
ASI_MASTER (ENUM).....	173
ASI_MASTER_MODE (ENUM).....	173
ASI_NET (STRUCT).....	178
ASI_SLAVE (ENUM).....	174
ASI_SLAVE_TYP (ENUM).....	174
Auf Standard-Ein- und Ausgangsdaten zugreifen.....	50
Aufzählungstypen (ENUM).....	173
Aufzählungstypen und komplexe Variablen.....	173
Ausgangsdaten der AS-i Slaves.....	55
Ausgangsparameter ..120, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 161, 163, 165, 168, 170, 172, 186, 192, 195, 197, 199, 202, 203, 205	

B

Beispiel	
Datum / Uhrzeit / NTP-Einstellungen lesen.....	188
LCEMS, LCEAS und LDAE von AS-i Master 2 lesen.....	189
Spracheinstellung des Geräts ändern.....	187
Beschreibung..... 119, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 167, 169, 171, 183, 191, 193, 196, 198, 200, 203, 204, 212, 216, 220, 224, 228, 232, 236, 240, 243, 247, 251, 255, 259, 263, 267, 272, 273, 274	
Betrieb.....	102
Betriebsmodus der fehlersicheren SPS.....	109
Betriebsmodus der Standard-SPS.....	107
Betriebszustände AC4S.....	107
Bibliothek ACnnnn_SYS_CMD.library.....	182
Bibliothek ACnnnn_Utils.library.....	117
Bibliothek SF_IO.library.....	190
Bibliothek SF_LogicalInterfaces.library.....	206
Boot-Applikation per SD-Karte löschen.....	106

C

Codefolgen der sicheren AS-i Slaves einlernen (teachen).....	105
CODESYS Development System installieren.....	12
CODESYS starten.....	15
CODESYS-Dokumentation nutzen.....	20

CODESYS-Projekt auf Gerät übertragen.....	102
CODESYS-Projekt erstellen.....	16
Ctrl_SetDiagInfo.....	200
CtrlASi_InSlave.....	191
CtrlASi_OutSlave.....	193
CtrlASi_ResetAllSlaves.....	196
CtrlLocalInputs.....	198

D

Datenaustausch zwischen Standard-SPS und fehlersicherer SPS.....	85
Datentransfer Safety-Bereich >>> Standard-Bereich.....	87
Datentransfer Standard-Bereich >>> Safety-Bereich.....	86
Diagnose-Informationen ..215, 219, 223, 227, 231, 235, 239, 246, 250, 254, 258, 262, 266, 271	
Diagnose-Meldungen.....	210
Digitale Ausgangsdaten.....	55
Digitale Ein- und Ausgangsdaten.....	53

E

Eigenschaften der Web-Visualisierung ändern.....	67
Ein- und Ausgangsdaten der Feldbus-Schnittstelle.....	54
Eingangsparameter ..119, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 167, 169, 171, 183, 192, 194, 196, 198, 201, 203, 204	
EtherCAT Master hinzufügen.....	43
EtherCAT Master konfigurieren.....	44
EtherCAT Master nutzen.....	43
EtherCAT Slave hinzufügen und konfigurieren.....	45

F

FB mit einmaliger Ausführung.....	57
FB mit zyklischer Ausführung.....	58
Fehler-Codes	
Lokale E/A-Schnittstelle.....	277
Sichere AS-i Slaves.....	275
Fehlersichere SPS.....	109, 114
Fehlersichere SPS konfigurieren.....	24
Feldbus-Daten.....	54
Feldbus-Funktionalität erweitern.....	42
Feldbus-Stack hinzufügen.....	46
FSoE-Verbindung konfigurieren.....	92
Funktionen des ifm-Packages nutzen.....	57

G

Get_Asi_Data.....	164
Get_Asi_PHY_Dat.....	119
Get_CDI.....	138
Get_FieldbusInfo.....	167
Get_InputParameter.....	142
Get_LAS.....	122
Get_LCE.....	128
Get_LCEAS.....	130
Get_LCEMS.....	132
Get_LDAE.....	134
Get_LDS.....	124
Get_LPF.....	136
Get_LPS.....	126
Get_OutputParameter.....	144
Get_PCD.....	140
GetLocalInput.....	203

Index

Gültigkeit der Schnittstellen-Daten51

H

Hardware11
 Haupt-Task konfigurieren70
 Hinweis
 Projekte mit Netzwerkvariablen105
 Hinweise25, 32
 Hinweise zu FB FSoEMaster93
 Hinweise zu logischen Geräten207
 Hinweise zur Verwendung von Austauschobjekten85

I

ifm AS-i Package13
 ifm weltweit • ifm worldwide • ifm à l'échelle internationale281
 ifm-Package deinstallieren14
 ifm-Package installieren14
 Installation12
 Intervall des Datenaustauschs einstellen88
 IP-Einstellungen ändern40

K

Kommandos an System und AS-i Master senden63
 Kommunikationspfad der fehlersicheren SPS setzen22
 Kommunikationspfad der Standard-SPS setzen21
 Komplexe Variablen (STRUCT)175
 Komplexe Variablen nutzen60
 Komponenten des ifm-Packages13
 Konfigurationsdaten der Slaves lesen62
 Konzept der logischen Geräte25, 32

L

Lizenzierung11
 Logische Geräte für die lokale E/A-Schnittstelle242
 Logische Geräte für sichere AS-i Slaves211
 Logisches Gerät konfigurieren29, 37
 Logisches Gerät zum Projektbaum hinzufügen27, 34
 Lokale E/A-Schnittstelle
 Auf nicht sichere Ein-/Ausgänge zugreifen80
 Diagnose-Informationen ausgeben80
 Logisches Gerät rücksetzen79

M

Mindestanzahl an AS-i Slaves76
 Möglichkeiten des Zugriffs auf Ein- und Ausgangsdaten50

N

Netzwerkeinstellungen ändern60
 Netzwerkeinstellungen lesen61
 Netzwerkvariablen nutzen68
 Neues Projekt mit AC4S erstellen17
 Notwendige Vorkenntnisse9
 Nutzerkonto anlegen20

O

Objekte einer sicheren Anwendung72
 Objekte einer Standard-SPS-Anwendung47

P

Parameter des logischen Geräts einstellen30, 37
 Parameter-Abbilder lesen61
 Parameterdaten 213, 217, 221, 225, 229, 233, 237, 240, 244, 248, 252, 256, 260, 264, 268, 272, 273, 274
 PLCopen Safety
 Diagnose-Informationen ausgeben81
 Programmierschnittstelle konfigurieren21
 Programmiersoftware CODESYS12
 Programmierung47
 Programmierung der fehlersicheren SPS71
 Prozessdaten213, 217, 221, 225, 229, 233, 237, 241, 244, 248, 252, 256, 260, 264, 269, 272, 273
 Prozessdaten der AS-i Slaves52
 Prozessdaten der nicht-sicheren lokalen Geräte56
 Prozesssicherheitszeit beim AC4S94
 Prozesssicherheitszeit einstellen94

Q

Querkommunikation68
 QuickSetupASi_Master169

R

Rechtliche Hinweise6
 Reihenfolge der Signalwechsel208
 Remanente Variablen nutzen48
 Reset112

S

SafetyFSoEMaster-Bibliothek84
 Safety-Funktionen des ifm AS-i Packages nutzen78
 Safety-Funktionen von CODESYS nutzen82
 SafetyPLCopen-Bibliothek83
 SafetyStandard-Bibliothek82
 Safety-Task konfigurieren101
 Schnelleinstieg15
 Schritt 1
 Typische Reaktionszeiten berechnen96
 Schritt 2
 Reaktionszeit der Sicherheitsfunktion berechnen97
 Schritt 3
 Watchdog-Zeiten definieren98
 Schritt 4
 Worst-Case-Reaktionszeit der Sicherheitsfunktion berechnen99
 Set_AddressMode146
 Set_ASi_Config148
 Set_LPS150
 Set_Mode152
 Set_PCD154
 Set_ProjectAll156
 Set_SlaveAddress158
 Set_SlaveExtendedID1160
 Set_SlaveParameter162
 Set_TimeDate171
 SetLocalOutput204
 SF_IN_ASi_conditionally_dependent220
 SF_IN_ASi_dependent224
 SF_IN_ASi_dependent_filter_nshutdown236
 SF_IN_ASi_dependent_filter_ntestreq232

Index

SF_IN_ASi_dependent_filter_w_testreq	228
SF_IN_ASi_forced	212
SF_IN_ASi_independent	216
SF_IN_local_conditionally_dependent	251
SF_IN_local_dependent	255
SF_IN_local_dependent_filter_nshutdown	267
SF_IN_local_dependent_filter_ntestreq	263
SF_IN_local_dependent_filter_testreq	259
SF_IN_local_forced	243
SF_IN_local_independent	247
SF_OUT_local_dual	273
SF_OUT_local_single	272
SF_OUT_local_testpulse	274
SF_OUTcontrol_ASi	240
Sichere Anwendung auf AC4S laden	104
Sichere Anwendung erstellen	73
Sichere Anwendung rücksetzen (kalt)	114
Sichere AS-i Ausgangs-Slaves	26
Sichere AS-i Eingangs-Slaves	25
Sichere AS-i Slaves	
Diagnose-Informationen ausgeben	79
Hilfssignale HSI_1 und HSI_2 an sichere AS-i Ausgangsmodule senden ..	79
Logisches Gerät rücksetzen	78
Sichere AS-i Slaves aus Projekt entfernen	31
Sichere AS-i Slaves in Projekt einbinden	27
Sichere AS-i Slaves konfigurieren	25
Sichere Feldbuskommunikation	92
Sichere Geräte an lokalen Ausgängen	33
Sichere Geräte an lokalen Eingängen	32
Sichere Geräte an lokaler E/A-Schnittstelle konfigurieren	32
Sichere Netzwerkvariablen einrichten	89
Sichere Netzwerkvariablen in Anwendung nutzen	91
Sichere Querkommunikation	89
Sicheres lokales Gerät aus Projekt entfernen	39
Sicheres lokales Gerät in Projekt einbinden	34
Sicheres Projekt pinnen	77
Sicheres Prozesssignal auf Variable abbilden	30, 38
Sicherheitshinweise	9
Slave-Listen lesen	61
Software	11
Standard-Anwendung auf AC4S laden	103
Standard-Anwendung rücksetzen (kalt)	113
Standard-Anwendung rücksetzen (Ursprung)	113
Standard-Anwendung rücksetzen (warm)	113
Standard-Geräte an lokaler E/A-Schnittstelle konfigurieren	40
Standard-SPS	107, 112
Standard-SPS konfigurieren	23
Standard-SPS-Anwendung erstellen	48
Standard-SPS-Applikation starten	108
Standard-SPS-Applikation stoppen	108
Startup-Test	208
Status der Spannungsversorgung lesen	62
Steuerinterface der ifm-Funktionsbausteine	57
System konfigurieren	59
System-Events	209
Systemkonfiguration	23
Systemvoraussetzungen	11
Systemzeit des Geräts ändern	49

T

Tabelle

AS-i Master-Kommandos	184
Systemkommandos	184
Task-Abarbeitung konfigurieren	70
Testsignal an lokalem Ausgang erzeugen	39
Timer-Events	209

U

Übersicht

Anwender-Dokumentation für AC4S	7
AS-i Funktionen (FB_ASi)	118
CODESYS-Dokumentation von 3S	8
Projektstruktur mit AC4S	19
System-Funktionen (FB_System)	166
Unterstützte Programmierbausteine (POUs)	74
Unterstützte Programmiersprachen	48, 75
Unterstützte Reset-Varianten	112, 114
Unterstützte Visualisierungstypen	64

V

Variable Komponenten der Reaktionszeit einstellen	100
Variablenklärung bei Datentransfer zwischen Standard- und Safety-Bereich	76
Verfügbare Feldbus-Stacks	42
Verfügbare logische Geräte für sichere AS-i Slaves	26
Verfügbare logische Geräte für sichere Ein-/Ausgänge	33
Verfügbare Safety-Bibliotheken	75
Verfügbarer Speicherplatz	77
Verwendete Symbole und Formatierungen	7
Verwendete Warnhinweise	10
Visualisierung erstellen	66
Visualisierung konfigurieren	67
Visualisierung zu einem Projekt hinzufügen	65
Visualisierungen nutzen	64
Visualisierung-Task konfigurieren	70
Vorbemerkung	6

W

Web-Visualisierung anzeigen	115
Werkseitig angelegte Austauschvariablen nutzen	86

Z

Zusätzliche Objekte für Datenaustausch anlegen	88
Zusätzliches Gerät zum Projekt hinzufügen	41
Zustand der Standard-Anwendung anzeigen	107
Zustände	207
Zustände der Eingangssignale	208
Zustände der fehlersicheren SPS	110
Zustände der sicheren Anwendung	110
Zustände der Standard-SPS-Anwendung	107
Zustandsdiagramm (State machine) .214, 218, 222, 226, 230, 234, 238, 245, 249, 253, 257, 261, 265, 270	
Zustandsmaschinen für logische Vorverarbeitung	207
Zustandsübergänge	207
Zweck des Dokuments	6
Zwischen den Zuständen wechseln	108, 111
Zykluszeit der fehlersicheren SPS einstellen	101

12 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Stand: 2017-12-18

8310

ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen

www.ifm.com • E-Mail: info@ifm.com

Service-Hotline: 0800 16 16 16 4 (nur Deutschland, Mo...Fr, 07.00...18.00 Uhr)

ifm Niederlassungen • Sales offices • Agences

D	Niederlassung Nord • 31135 Hildesheim • Tel. 05121 7667-0 Niederlassung West • 45128 Essen • Tel. 0201 36475 -0 Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 02351 4301-0 Niederlassung Süd-West • 64646 Heppenheim • Tel. 06252 7905-0 Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 07021 8086-0 Niederlassung Bayern • 82178 Puchheim • Tel. 089 80091-0 Niederlassung Ost • 07639 Tautenhain • Tel. 036601 771-0
AE	ifm electronic FZC • Saif Zone, Sharjah • Tel. +971- 6-5573601
AR	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 5353-3436
AT	ifm electronic gmbh • 1120 Wien • Tel. +43 / 1 / 617 45 00
AU	ifm efector pty ltd. • Mulgrave Vic 3170 • Tel. +61 1300 365 088
BE	ifm electronic n.v./s.a. • 1731 Zellik • Tel. +32 2 481 0220
BG	ifm electronic eood • 1202 Sofia • Tel. +359 2 807 59 69
BR	ifm electronic Ltda. • 03337-000 Sao Paulo / SP • Tel. +55-11-2672-1730
CA	ifm efector Canada inc. • Mississauga, ON L5N 2X7 • Tel. +1 855-436-2262
CH	ifm electronic ag • 4624 Härkingen • Tel. +41 / 800 88 80 33
CL	ifm electronic SpA • Oficina 5041 Comuna de Conchalí • Tel. +56-2-32239282
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • Tel. +86 21 3813 4800
CZ	ifm electronic, spol. s.r.o. • 140 00 Praha 4 • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 Brøndby • Tel. +45 70 20 11 08
ES	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
FI	ifm electronic oy • 00440 Helsinki • Tel. +358 75 329 5000
FR	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
GB	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 / 20 / 8213 0000
GR	ifm electronic monoprosofi E.P.E. • 15125 Amaroussio • Tel. +30 210 61 800 90
HU	ifm electronic kft. • 9028 Győr • Tel. +36-96 / 518-397
IN	ifm electronic India Private Limited • Kolhapur, 416234 • Tel. +91 / 231 / 267 27 70
IE	ifm electronic (Ireland) Ltd. • Dublin 22 • Tel. +353 / 1 / 461 32 00
IT	ifm electronic s.r.l. • 20864 Agrate Brianza (MB) • Tel. +39 39-6899982
JP	efector co., ltd. • Chiba-shi, Chiba 261-7118 • Tel. +81 043-299-2070
KR	ifm electronic Ltd. • 04420 Seoul • Tel. +82 2-790-5610
MX	ifm efector S. de R.L. de C.V. • San Pedro Garza Garcia, N.L. 66269 • Tel. +52-81-8040-3535
MY	ifm electronic Pte. Ltd • 47100 Puchong, Selangor • Tel. +603 8066 9853
NA	ifm elctronic (pty) Ltd • 25 Dr. W. Kulz Street Windhoek • Tel. +264 61 300984
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341-438 438
NZ	ifm efector pty ltd • 930 Great South Road Penrose, Auckland • Tel. +64 / 95 79 69 91
PL	ifm electronic sp. z o.o. • 40-106 Katowice • Tel. +48 32 70 56 400
PT	ifm electronic s.a. • 4410-137 São Félix da Marinha • Tel. +351 223 71 71 08
RO	ifm electronic s.r.l • Sibiu 557260 • Tel. +40 269 224 550
RU	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
SG	ifm electronic Pte Ltd • 609 916 Singapore • Tel. +65 6562 8661
SK	ifm electronic s.r.o. • 831 06 Bratislava • Tel. +421 244 872 329
SE	ifm electronic ab • 412 50 Göteborg • Tel. +46 31-750 23 00
TR	ifm electronic Ltd. Sti. • 34381 Sisli, Istanbul • Tel. +90 212 210 50 80
TW	ifm electronic • Kaohsiung City, 806, Taiwan R.O.C. • Tel. +886 7 3357778
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501-85-43
US	ifm efector inc. • Malvern, PA 19355 • Tel. +1 800-441-8246
VN	ifm electronic Vietnam Co., Ltd. • 700000 Ho Chi Minh City • Tel. +84-28-2253.6715
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 450 0412

Technische Änderungen behalten wir uns ohne vorherige Ankündigung vor.

We reserve the right to make technical alterations without prior notice.

Nous nous réservons le droit de modifier les données techniques sans préavis.

