



Inhaltsverzeichnis

1	Über diese Anleitung	4
1.1	Copyright.....	4
1.2	Übersicht: Dokumentations-Module für ecomatmobile-Geräte.....	5
1.3	CODESYS-Programmierhandbuch.....	5
1.4	Was bedeuten die Symbole und Formatierungen?	6
1.5	Wie ist diese Dokumentation aufgebaut?	7
1.6	Historie der Anleitung (CR0451).....	8
2	Sicherheitshinweise	9
2.1	Beachten!	9
2.2	Welche Vorkenntnisse sind notwendig?	10
2.3	Anlaufverhalten der Steuerung	10
3	Systembeschreibung	11
3.1	Angaben zum Gerät.....	11
3.1.1	Zubehör.....	11
3.2	Hardware-Beschreibung	12
3.2.1	Hardware-Aufbau	12
3.2.2	Status-LED	16
3.3	Schnittstellen-Beschreibung	17
3.3.1	CAN-Schnittstellen	17
3.4	Software	18
3.4.1	Software-Module für das Gerät	18
3.4.2	Programmierhinweise für CODESYS-Projekte.....	21
3.4.3	Betriebszustände.....	25
3.4.4	Leistungsgrenzen des Geräts.....	28
4	Konfigurationen	35
4.1	Laufzeitsystem einrichten	35
4.1.1	Laufzeitsystem neu installieren	36
4.1.2	Laufzeitsystem aktualisieren	37
4.1.3	Installation verifizieren	37
4.2	Programmiersystem einrichten	38
4.2.1	Programmiersystem manuell einrichten	38
4.2.2	Programmiersystem über Templates einrichten	42
4.3	Funktionskonfiguration, allgemein	43
4.3.1	Systemvariablen	43
4.4	Variablen	44
4.4.1	Retain-Variablen.....	44
4.4.2	Netzwerkvariablen.....	45
5	ifm-Funktionselemente	46
5.1	ifm-Bibliotheken für das Gerät CR0451	46
5.1.1	Mindestens erforderliche Bibliotheken.....	46
5.1.2	Bibliothek ifm_CR0451_Vxxyzz.LIB	47
5.1.3	Bibliothek ifm_CR0451_Init_Vxxyzz.LIB.....	47
5.1.4	Bibliothek ifm_PDMsmart_util_Vxxyzz.LIB	48
5.1.5	Bibliothek ifm_RAWCan_NT_Vxxyzz.LIB.....	48
5.1.6	Bibliothek ifm_CANOpen_NT_Vxxyzz.LIB	49
5.1.7	Bibliothek ifm_J1939_NT_Vxxyzz.LIB	50

Inhalt

5.2	ifm-Bausteine für das Gerät CR0451	51
5.2.1	Baustein-Ausgänge	52
5.2.2	Bausteine: RAW-CAN (Layer 2)	53
5.2.3	Bausteine: CANopen	79
5.2.4	Bausteine: SAE J1939	124
5.2.5	Bausteine: System	156
5.2.6	Bausteine: Grafik	174
6	Diagnose und Fehlerbehandlung	184
6.1	Diagnose	184
6.2	Fehler	184
6.3	Reaktion auf Fehlermeldungen	185
6.3.1	Beispielablauf für Reaktion auf Fehlermeldungen	185
6.4	CAN / CANopen: Fehler und Fehlerbehandlung	185
7	Anhang	186
7.1	Systemmerker	186
7.2	Fehler-Tabellen	188
7.2.1	Fehlermerker	188
7.2.2	Fehler: CAN / CANopen	188
8	Begriffe und Abkürzungen	189
9	Index	203
10	Notizen • Notes • Notes	207
11	ifm weltweit • ifm worldwide • ifm à l'échelle internationale	211

1 Über diese Anleitung

Inhalt

Copyright	4
Übersicht: Dokumentations-Module für ecomatmobile-Geräte	5
CODESYS-Programmierhandbuch	5
Was bedeuten die Symbole und Formatierungen?	6
Wie ist diese Dokumentation aufgebaut?	7
Historie der Anleitung (CR0451)	8

202

1.1 Copyright

6088

© 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™
- IO-Link® (→ www.io-link.com) ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- 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 Übersicht: Dokumentations-Module für ecomatmobile-Geräte

17405

Die Dokumentation für **ecomatmobile**-Geräte besteht aus folgenden Modulen:

1.	Datenblatt
Inhalt:	Technische Daten in Tabellenform
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0451 > [Technische Daten im PDF-Format]
2.	Montageanleitung / Betriebsanleitung
Inhalt:	Anleitung für Montage, elektrische Installation, (Inbetriebnahme*), Technische Daten
Quelle:	Anleitung wird mit dem Gerät mitgeliefert Auch zu finden auf der ifm -Homepage: → www.ifm.com > Land wählen > [Datenblattsuche] > CR0451 > [Betriebsanleitungen]
3.	Programmierhandbuch + Online-Hilfe
Inhalt:	Beschreibung der Konfiguration und der Funktionen der Geräte-Software
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0451 > [Betriebsanleitungen]
4.	Systemhandbuch "Know-How ecomatmobile"
Inhalt:	Hintergrundwissen zu folgenden Themen: • Übersicht Templates und Demo-Programme • CAN, CANopen • Ausgänge steuern • User-Flash-Speicher • Visualisierungen • Übersicht Dateien und Bibliotheken
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0451 > [Betriebsanleitungen]

*) Die in Klammern gesetzten Beschreibungen sind nur in den Anleitungen bestimmter Geräte enthalten.

1.3 CODESYS-Programmierhandbuch

17542

Im ergänzenden "Programmierhandbuch CODESYS V2.3" der 3S GmbH erhalten Sie weitergehende Informationen über die Nutzung des Programmiersystems.

Dieses Handbuch steht auf der **ifm**-Homepage als kostenloser Download zur Verfügung:

→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]

Handbücher und Online-Hilfen für **ecomatmobile** finden Sie auch hier:

→ **ecomatmobile**-DVD "Software, tools and documentation"

1.4 Was bedeuten die Symbole und Formatierungen?

203

Folgende Symbole oder Piktogramme verdeutlichen Ihnen unsere Hinweise in unseren Anleitungen:

 WARNUNG	
Tod oder schwere irreversible Verletzungen sind möglich.	
 VORSICHT	
Leichte reversible Verletzungen sind möglich.	
ACHTUNG	
Sachschaden ist zu erwarten oder möglich.	
	Wichtige Hinweise auf Fehlfunktionen oder Störungen
	Weitere Hinweise
► ...	Handlungsaufforderung
> ...	Reaktion, Ergebnis
→ ...	"siehe"
abc	Querverweis
123	Dezimalzahl
0x123	Hexadezimalzahl
0b010	Binärzahl
[...]	Bezeichnung von Tasten, Schaltflächen oder Anzeigen

1.5 Wie ist diese Dokumentation aufgebaut?

204
1508

Diese Dokumentation ist eine Kombination aus verschiedenen Anleitungstypen. Sie ist eine Lernanleitung für den Einsteiger, aber gleichzeitig auch eine Nachschlageanleitung für den versierten Anwender. Dieses Dokument richtet sich an die Programmierer der Anwendungen.

Und so finden Sie sich zurecht:

- Um gezielt zu einem bestimmten Thema zu gelangen, benutzen Sie bitte das Inhaltsverzeichnis.
- Mit dem Stichwortregister "Index" gelangen Sie ebenfalls schnell zu einem gesuchten Begriff.
- Am Anfang eines Kapitels geben wir Ihnen eine kurze Übersicht über dessen Inhalt.
- Abkürzungen und Fachbegriffe → Anhang.

Bei Fehlfunktionen oder Unklarheiten setzen Sie sich bitte mit dem Hersteller in Verbindung:

→ www.ifm.com > Land wählen > [Kontakt].

Wir wollen immer besser werden! Jeder eigenständige Abschnitt enthält in der rechten oberen Ecke eine Identifikationsnummer. Wenn Sie uns über Unstimmigkeiten unterrichten wollen, dann nennen Sie uns bitte diese Nummer zusammen mit Titel und Sprache dieser Dokumentation. Vielen Dank für Ihre Unterstützung!

Im Übrigen behalten wir uns Änderungen vor, so dass sich Abweichungen vom Inhalt der vorliegenden Dokumentation ergeben können. Die aktuelle Version finden Sie auf der **ifm**-Homepage:

→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

1.6 Historie der Anleitung (CR0451)

12111

Was hat sich wann in dieser Anleitung geändert? Ein Überblick:

Datum	Thema	Änderung
2012-07-24	Laufzeitsystem	Upgrade auf V02
2013-06-24	diverse	neue Dokumentenstruktur
2013-09-05	Laufzeitsystem	Upgrade auf V03
2014-03-24	Visualisierungsgrenzen	Angaben zum zulässigen Zeichenbereich
2014-04-29	FB CAN_REMOTE_RESPONSE	Beschreibung FB-Eingang ENABLE präzisiert
2014-05-12	Leistungsgrenzen CAN	Leistungsgrenzen ergänzt für CAN, CANopen und CAN J1939
2014-06-30	Name der Dokumentation	"Systemhandbuch" umbenannt zu "Programmierhandbuch"
2015-01-13	Dokumentationsstruktur Fehlercodes, Systemmerker	• Fehlermerker: nur noch im Anhang, Kapitel Systemmerker • CAN / CANopen Fehler und Fehlerbehandlung: nur noch im Systemhandbuch "Know-How" • Fehlercodes, EMCY-Codes: nun im Anhang, Kapitel Fehler-Tabellen
2015-03-10	Verfügbarer Speicher	Darstellung verbessert

2 Sicherheitshinweise

Inhalt

Beachten!	9
Welche Vorkenntnisse sind notwendig?	10
Anlaufverhalten der Steuerung	10
	213

2.1 Beachten!

6091
11212

Mit den in dieser Anleitung gegebenen Informationen, Hinweisen und Beispielen werden keine Eigenschaften zugesichert. Die abgebildeten Zeichnungen, Darstellungen und Beispiele enthalten weder Systemverantwortung noch anwendungsspezifische Besonderheiten.

- ▶ Die Sicherheit der Maschine/Anlage muss auf jeden Fall eigenverantwortlich durch den Hersteller der Maschine/Anlage gewährleistet werden.
- ▶ Beachten Sie die nationalen Vorschriften des Landes, in welchem die Maschine/Anlage in Verkehr gebracht werden soll!



WARNUNG

Bei Nichtbeachten der Hinweise in dieser Anleitung sind Sach- oder Körperschäden möglich! Die **ifm electronic gmbh** übernimmt hierfür keine Haftung.

- ▶ Die handelnde Person muss vor allen Arbeiten an und mit diesem Gerät die Sicherheitshinweise und die betreffenden Kapitel dieser Anleitung gelesen und verstanden haben.
- ▶ Die handelnde Person muss zu Arbeiten an der Maschine/Anlage autorisiert sein.
- ▶ Die handelnde Person muss für die auszuführende Arbeit über die erforderliche Ausbildung und Qualifikation verfügen.
- ▶ Beachten Sie die Technischen Daten der betroffenen Geräte!
Das aktuelle Datenblatt finden Sie auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Technische Daten im PDF-Format]
- ▶ Beachten Sie die Montage- und Anschlussbedingungen sowie die bestimmungsgemäße Verwendung der betroffenen Geräte!
→ mitgelieferte Montageanleitung oder auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]
- ▶ Beachten Sie die Korrekturen und Hinweise in den "Release-Notes" zur vorhandenen Hardware, Software und Dokumentation auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

2.2 Welche Vorkenntnisse sind notwendig?

215

Das Dokument richtet sich an Personen, die über Kenntnisse der Steuerungstechnik und SPS-Programmierkenntnisse mit IEC 61131-3 verfügen.

Zum Programmieren der SPS sollten die Personen zusätzlich mit der Software CODESYS vertraut sein.

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. Das Dokument enthält Angaben zum korrekten Umgang mit dem Produkt.

Lesen Sie dieses Dokument vor dem Einsatz, damit Sie mit Einsatzbedingungen, Installation und Betrieb vertraut werden. Bewahren Sie das Dokument während der gesamten Einsatzdauer des Gerätes auf.

Befolgen Sie die Sicherheitshinweise.

2.3 Anlaufverhalten der Steuerung

15233



WARNUNG

Gefahr durch unbeabsichtigtes und gefährliches Anlaufen von Maschinen- oder Anlagenteilen!

- ▶ Der Programmierer muss bei der Programmerstellung verhindern, dass nach Auftreten eines Fehlers (z.B. NOT-HALT) und der anschließenden Fehlerbeseitigung unbeabsichtigt Maschinen- oder Anlagenteile gefährlich anlaufen können!
⇒ Wiederanlaufsperr realisieren!
- ▶ Dazu im Fehlerfall die in Frage kommenden Ausgänge im Programm logisch abschalten!

Ein Wiederanlauf kann z.B. verursacht werden durch:

- Spannungswiederkehr nach Spannungsausfall
- Reset nach Watchdog-Ansprechen wegen zu langer Zykluszeit
- Fehlerbeseitigung nach NOT-HALT

So erreichen Sie sicheres Verhalten der Steuerung:

- ▶ Spannungsversorgung im Anwendungsprogramm überwachen.
- ▶ Im Fehlerfall alle relevanten Ausgänge im Anwendungsprogramm ausschalten.
- ▶ Aktuatoren, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Anwendungsprogramm überwachen (Feedback).
- ▶ Relaiskontakte, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Anwendungsprogramm überwachen (Feedback).
- ▶ Bei Bedarf im Anwendungsprojekt sicherstellen, dass verschweißte Relaiskontakte keine gefahrbringenden Bewegungen auslösen oder fortführen können.

6827

3 Systembeschreibung

Inhalt

Angaben zum Gerät	11
Hardware-Beschreibung.....	12
Schnittstellen-Beschreibung.....	17
Software	18
	975

3.1 Angaben zum Gerät

8316

Diese Anleitung beschreibt aus der Gerätefamilie für den mobilen Einsatz, **ecomatmobile** der **ifm electronic gmbh**:

- BasicDisplay: CR0451

Das Display gehört zur Familie der BasicController: CR040n, CR041n, CR043n.

3.1.1 Zubehör

8317

Zum BasicDisplay gibt es reichhaltiges Zubehör. Beispiele:

EC0402	Gehäuseabdeckung für BasicController: CR040n, CR041n, CR043n mit Öffnung für BasicDisplay: CR045n
EC0403	Montagerahmen für Fronttafel-Montage von CR0451
EC0405	RAMmount-Set für den Einsatz von CR0451 als Standgerät
EC0452	Verbindungsleitung für Stromversorgung und CAN zwischen Display und BasicController bei Verwendung der Gehäuseabdeckung EC0402
EC0453	Verbindungsleitung für Stromversorgung und CAN zwischen Display und BasicController und einem weiteren BasicController bei Verwendung der Gehäuseabdeckung EC0402
EC0454	Verbindungsleitung 5 m für Versorgung und CAN zwischen Display und BasicController
---	Software "Maintenance Tool" zum Aktualisieren von Firmware, Laufzeitsystem und Anwendungsprogramm → ecomatmobile -DVD "Software, tools and documentation"

Zubehör zum Artikel finden Sie auf der **ifm**-Homepage:

→ www.ifm.com > Land wählen > [Datenblattsuche] > Artikel-Nr. > [Zubehör]

3.2 Hardware-Beschreibung

Inhalt	
Hardware-Aufbau	12
Status-LED	16

14081

3.2.1 Hardware-Aufbau

Inhalt	
Verfügbarer Speicher	13
Schnittstellen	14
Farb-Display des CR0451	14
Bedien-Elemente des CR0451	14
Tasten-LEDs dimmbar	15
Funktionsrückmeldungen des Geräts	15
Anschluss auf Gehäuserückwand	15

15269

Schutzart IP 65

frontseitig, im eingebauten Zustand: Schutzart IP 67

Verfügbarer Speicher

13736

FLASH-Speicher

13053

FLASH-Speicher (nichtflüchtiger, langsamer Speicher) insgesamt im Gerät vorhanden	1 536 kByte
--	-------------

Davon sind folgende Speicherbereiche reserviert für ...

maximale Größe für das Anwendungsprogramm	512 kByte
Daten außerhalb des Anwendungsprogramms Daten mit FB FLASH_READ (→ Seite 158) lesen (bei Files: abzüglich 128 Byte für Header)	64 kByte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

SRAM

14027

SRAM (flüchtiger, schneller Speicher) insgesamt im Gerät vorhanden SRAM steht hier allgemein für alle Arten von flüchtigen, schnellen Speichern.	592 kByte
--	-----------

Davon sind folgende Speicherbereiche reserviert für ...

vom Anwendungsprogramm reservierte Daten	128 kByte
--	-----------

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

FRAM

2262

FRAM (nichtflüchtiger, schneller Speicher) insgesamt im Gerät vorhanden FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.	2 kByte
--	---------

Davon sind folgende Speicherbereiche reserviert für ...

im Anwendungsprogramm als VAR_RETAIN deklarierte Variablen	128 Byte
fest als remanent definierte Merker (%MB0...127)	128 Byte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

Schnittstellen

15383
15272

In diesem **ecomatmobile**-Gerät sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

CAN-Schnittstelle	CAN 1	CAN 2	CAN 3	CAN 4
voreingestellte Download-ID	ID 127	ID 126	ID 125	ID 124
CAN-Protokolle	CAN Layer 2	Schnittstelle nicht vorhanden	Schnittstelle nicht vorhanden	Schnittstelle nicht vorhanden
	CANopen			
	SAE J1939			

Standard-Baudrate = 250 kBit/s

 Alle CAN-Schnittstellen können zeitgleich mit allen CAN-Protokollen arbeiten. Die verwendeten IDs dürfen sich nicht beeinträchtigen!

Farb-Display des CR0451

8341

Bezeichnung	Daten
Technologie	TFT
Bildschirmdiagonale	2,8" (7,1 cm)
Seitenverhältnis	4:3
Auflösung	320 x 240 Pixel
Farbtiefe	8 Bit = 256 Farben darstellbar über definierte Farbpalette ► Bild als 256-Farben-Datei erstellen!
Hintergrundbeleuchtung	LED dimmbar in 1%-Schritten Einstellung a) temporär änderbar und b) voreinstellbar (gespeichert)

Bedien-Elemente des CR0451

8342

Das Display ist mit folgenden Bedien-Elementen ausgestattet:

- 4 Funktionstasten [F1]...[F4]
mit LEDs hinterleuchtet
- 1 Kreuzwippe
als Kombination von 4 unabhängigen Tasten
mit LEDs hinterleuchtet
- 1 Taste [OK]
mit LED hinterleuchtet

Alle Tasten arbeiten unabhängig voneinander.

Das Gerät erkennt mehrere gleichzeitig betätigte Tasten und wertet diese aus.

Tasten-LEDs dimmbar

8369

- Alle Bedien-Elemente sind mit LEDs hinterleuchtet.

Nacht-Design der Bedien-Elemente:

- Die LEDs für alle Bedien-Elemente sind nur gemeinsam dimmbar:
 - Merker KEY_BACKLIGHT_CTRL

Funktionsrückmeldungen des Geräts

15261

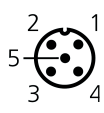
Weitere Funktionsrückmeldungen des Geräts:

- **Funktionsanzeige** mit 2-Farb-LED (rot/grün), → Kapitel **Status-LED** (→ Seite [16](#))
mehrfarbig programmierbar → FB **SET_LED** (→ Seite [170](#))

Anschluss auf Gehäuserückwand

8351

Stecker M12, A-codiert, für Versorgung und CAN:

Bild	Pin	Bezeichnung	Bemerkung
	1	n.c.	----
	2	VBB	8...32 V DC
	3	GND	Klemme 31
	4	CAN_H	
	5	CAN_L	

3.2.2 Status-LED

7998

Die Betriebszustände werden durch die integrierte Status-LED (Voreinstellung) angezeigt.

LED-Farbe	Blinkfrequenz	Beschreibung
aus	konstant aus	keine Betriebsspannung
Orange	kurzzeitig ein	INIT-Zustand, Reset-Checks
Grün	5 Hz	kein Laufzeitsystem geladen
Grün	2 Hz	RUN-Zustand: Anwendungsprogramm läuft
Grün	konstant ein	STOP-Zustand: Anwendungsprogramm angehalten
Rot	5 Hz	STOP-Zustand mit Fehler: Anwendungsprogramm angehalten Ursache: Unterspannung
Rot	10 Hz	STOP-Zustand mit Fehler: Anwendungsprogramm angehalten Ursache: Laufzeitüberschreitung des Anwendungsprogramms oder der Visualisierung: ▶ Anwendungsprogramm löschen! ▶ PowerOn-Reset ▶ Anwendungsprogramm neu ins Gerät laden
Rot	konstant ein	FATAL-ERROR: Anwendungsprogramm angehalten Ursache: Software-Watchdog ist ausgefallen ▶ PowerOn-Reset Wenn ohne Erfolg: ▶ Goto Bootloader ▶ PowerOn-Reset ▶ BasicSystem neu ins Gerät laden ▶ Anwendungsprogramm neu ins Gerät laden Wenn ohne Erfolg: ▶ Hardware-Fehler: Gerät an ifm einsenden!

Die Betriebszustände STOP und RUN können vom Programmiersystem geändert werden.

LED im Anwendungsprogramm steuern

15481

Mit **SET_LED** (→ Seite [170](#)) können im Anwendungsprogramm für den RUN-Zustand Frequenz und Farbe der Status-LED geändert werden.

! Der Einsatz des LED-Bausteins im Anwendungsprogramm ersetzt im RUN-Zustand die System-Voreinstellung der Status-LED.

3.3 Schnittstellen-Beschreibung

Inhalt

CAN-Schnittstellen	17
--------------------------	----

14098

3.3.1 CAN-Schnittstellen

Inhalt

CAN: Schnittstellen und Protokolle	17
--	----

14101

Anschlüsse und Daten → Datenblatt

CAN: Schnittstellen und Protokolle

15270
15271

Das Gerät wird mit nur einer CAN-Schnittstelle ausgerüstet.

Grundsätzlich kann die Schnittstelle mit folgenden Funktionen genutzt werden:

- RAW-CAN (Layer 2): CAN auf Ebene 2 (→ Kapitel **Bausteine: RAW-CAN (Layer 2)** (→ Seite [53](#)))
- CANopen-Master / CANopen-Slave (→ Kapitel **Bausteine: CANopen** (→ Seite [79](#)))
- CANopen-Netzwerkvariablen (via CODESYS) (→ Kapitel **Netzwerkvariablen** (→ Seite [45](#)))
- SAE J1939 (für Antriebsmanagement, → Kapitel **Bausteine: SAE J1939** (→ Seite [124](#)))
- Buslast-Erkennung
- Errorframe-Zähler
- Download-Schnittstelle
- 100 % Buslast ohne Paketverlust

15272

In diesem **ecomatmobile**-Gerät sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

CAN-Schnittstelle	CAN 1	CAN 2	CAN 3	CAN 4
voreingestellte Download-ID	ID 127	ID 126	ID 125	ID 124
CAN-Protokolle	CAN Layer 2	Schnittstelle nicht vorhanden	Schnittstelle nicht vorhanden	Schnittstelle nicht vorhanden
	CANopen			
	SAE J1939			

Standard-Baudrate = 250 kBit/s

 Alle CAN-Schnittstellen können zeitgleich mit allen CAN-Protokollen arbeiten. Die verwendeten IDs dürfen sich nicht beeinträchtigen!

3.4 Software

Inhalt	
Software-Module für das Gerät	18
Programmierhinweise für CODESYS-Projekte	21
Betriebszustände	25
Leistungsgrenzen des Geräts	28

14107

3.4.1 Software-Module für das Gerät

Inhalt	
Bootloader	19
Laufzeitsystem	19
Anwendungsprogramm	19
Bibliotheken	20

14110

Die Software in diesem Gerät setzt wie folgt auf der Hardware auf:

Software-Modul	Anwender kann das Modul ändern?	womit?
Anwendungsprogramm mit Bibliotheken	ja	CODESYS, MaintenanceTool
Laufzeitsystem (LZS) *)	Upgrade ja Downgrade ja	MaintenanceTool
Bootloader	nein	---
(Hardware)	nein	---

*) Die Laufzeitsystem-Versionsnummer muss der Target-Versionsnummer in der CODESYS-Zielsystemeinstellung entsprechen!
→ Kapitel **Target einrichten** (→ Seite [39](#))

Nachfolgend beschreiben wir diese Software-Module:

Bootloader

14111

Im Auslieferungszustand enthalten **ecomatmobile**-Controller nur den Bootloader.

Der Bootloader ist ein Startprogramm, mit dem das Laufzeitsystem und das Anwendungsprogramm auf dem Gerät nachgeladen werden können.

Der Bootloader enthält Grundroutinen...

- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Laufzeitsystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Laufzeitsystem

14112

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm.

→ Kapitel **Software-Module für das Gerät** (→ Seite [18](#))

Im Auslieferungszustand ist im Normalfall kein Laufzeitsystem im Controller geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Bootloader aktiv. Dieser stellt die minimalen Funktionen für den Laufzeitsystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. CAN).

Der Laufzeitsystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Anwendungsprogramm kann anschließend (auch mehrmals) in den Controller geladen werden, ohne das Laufzeitsystem zu beeinflussen.

Das Laufzeitsystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der **ifm electronic gmbh** heruntergeladen werden:

→ www.ifm.com > Land wählen > [Service] > [Download]

Anwendungsprogramm

15274
14118

Software, die speziell für die Anwendung vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

8340



WARNUNG

Für die sichere Funktion der Anwendungsprogramme, die vom Anwender erstellt werden, ist dieser selbst verantwortlich. Bei Bedarf muss er zusätzlich entsprechend der nationalen Vorschriften eine Abnahme durch entsprechende Prüf- und Überwachungsorganisationen durchführen lassen.

Die Visualisierungsseiten und darin eingebettete Grafiken sind Bestandteil des CODESYS-Anwendungsprogramms.

Bibliotheken

15409

ifm electronic bietet passend für jedes Gerät eine Reihe von Bibliotheken (*.LIB) an, die Programmmodule für das Anwendungsprogramm enthalten. Beispiele:

Bibliothek	Verwendung
ifm_CR0451_Vxxyzz.LIB	gerätespezifische Bibliothek Muss immer im Anwendungsprogramm enthalten sein!
ifm_RawCAN_NT_Vxxyzz.LIB	(optional) wenn eine CAN-Schnittstelle des Geräts mit CAN Layer 2 betrieben werden soll
ifm_CANopen_NT_Vxxyzz.LIB	(optional) wenn eine CAN-Schnittstelle des Geräts als CANopen-Master oder als CANopen-Slave betrieben werden soll
ifm_J1939_NT_Vxxyzz.LIB	(optional) wenn eine CAN-Schnittstelle des Geräts mit einer Motorsteuerung kommunizieren soll

Detailinformationen: → **ifm-Bibliotheken für das Gerät CR0451** (→ Seite [46](#))

3.4.2 Programmierhinweise für CODESYS-Projekte

Inhalt	
FB, FUN, PRG in CODESYS	22
Zykluszeit beachten!.....	22
Anwendungsprogramm erstellen.....	23
ifm-Maintenance-Tool nutzen.....	24
Verteilen des Anwendungsprogramms	24

7426

Hier erhalten Sie Tipps zum Programmieren des Geräts.

- Beachten Sie die Hinweise im CODESYS-Programmierhandbuch
 - www.ifm.com > Land wählen > [Datenblattsuche] > CR0451 > [Betriebsanleitungen],
 - **ecomatmobile**-DVD "Software, tools and documentation".

FB, FUN, PRG in CODESYS

15410

In CODESYS unterscheiden wir folgende Typen von Bausteinen (POUs):

FB = function block = Funktionsbaustein

- Ein FB kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein FB darf in einem Projekt mehrmals aufgerufen werden.
- Für jeden Aufruf muss eine Instanz deklariert werden.
- Erlaubt: Im FB aufrufen von FB und FUN.

FUN = function = Funktion

- Eine Funktion kann mehrere Eingänge, aber nur einen Ausgang haben.
- Der Ausgang ist vom gleichen Datentyp wie die Funktion selbst.

PRG = program = Programm

- Ein PRG kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein PRG darf in einem Projekt nur einmal aufgerufen werden.
- Erlaubt: im PRG aufrufen von PRG, FB und FUN.

! HINWEIS

Funktionsbausteine dürfen NICHT in Funktionen aufgerufen werden!

Sonst: Bei der Ausführung stürzt das Anwendungsprogramm ab.

Alle Bausteine (POUs) dürfen NICHT rekursiv aufgerufen werden, auch nicht indirekt!

Eine IEC-Anwendung darf maximal 8000 Bausteine (POUs) enthalten, in diesem Gerät maximal 512 Bausteine (POUs)!

Hintergrund:

Alle Variablen von Funktionen...

- werden beim Aufruf initialisiert und
- werden nach der Rückkehr zum Aufrufer ungültig.

Funktionsbausteine haben 2 Aufrufe:

- einen Initialisierungsaufruf und
- den eigentlichen Aufruf, um irgend etwas zu tun.

Folglich heißt das für den FB-Aufruf in einer Funktion:

- jedesmal erfolgt ein zusätzlicher Initialisierungsaufruf und
- die Daten des letzten Aufrufs gehen verloren.

Zykluszeit beachten!

8006

Bei den frei programmierbaren Geräten aus der Controller-Familie **ecomatmobile** stehen in einem großen Umfang Bausteine zur Verfügung, die den Einsatz der Geräte in den unterschiedlichsten Anwendungen ermöglichen.

Da diese Bausteine je nach Komplexität mehr oder weniger Systemressourcen belegen, können nicht immer alle Bausteine gleichzeitig und mehrfach eingesetzt werden.

ACHTUNG

Gefahr von zu tragem Verhalten des Geräts!

Zykluszeit darf nicht zu lang werden!

- ▶ Beim Erstellen des Anwendungsprogramms die oben aufgeführten Empfehlungen beachten und durch Austesten überprüfen.
- ▶ Bei Bedarf durch Neustrukturieren der Software und des Systemaufbaus die Zykluszeit vermindern.

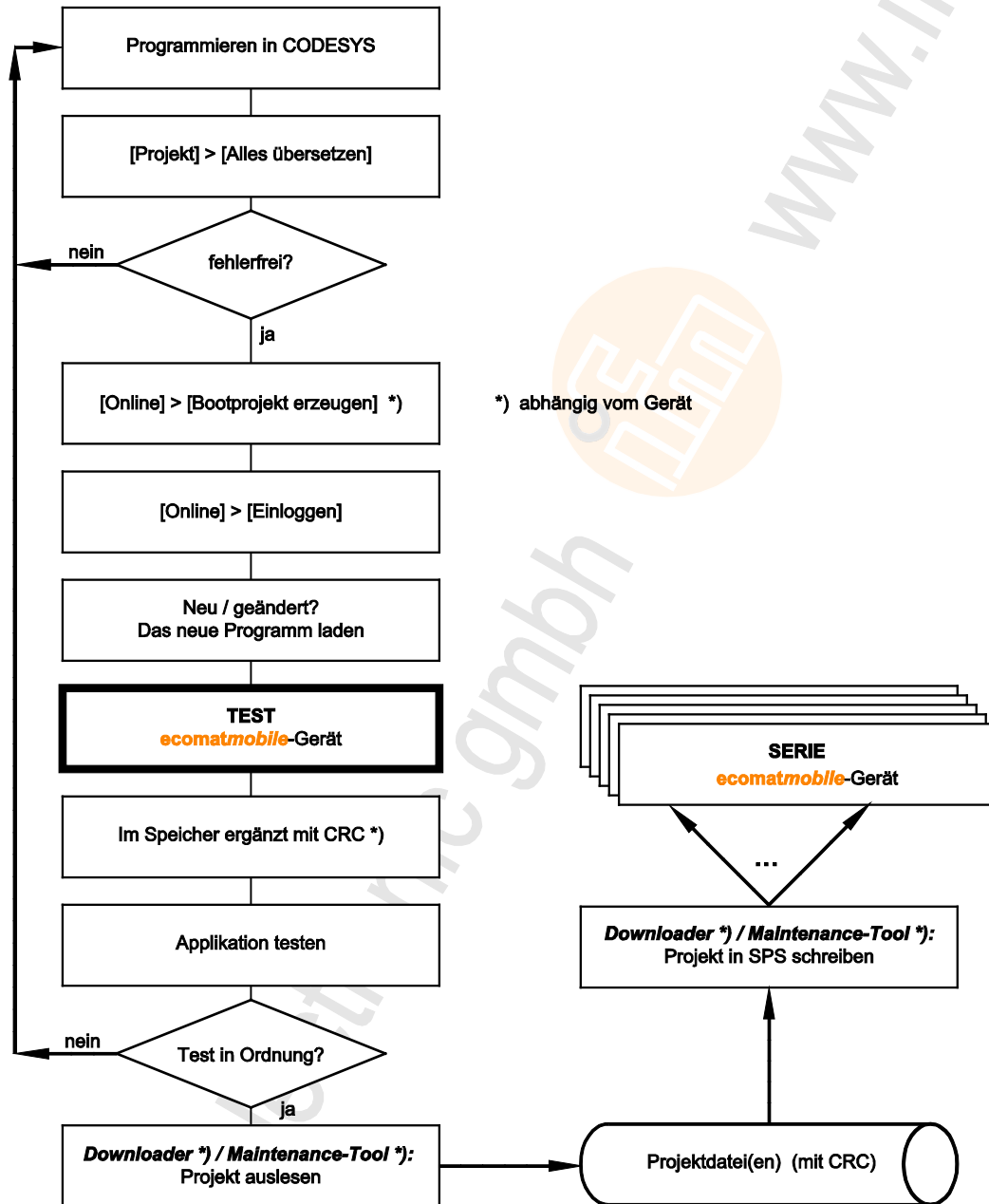
Anwendungsprogramm erstellen

8007

Das Anwendungsprogramm wird mit dem Programmiersystem CODESYS erstellt und während der Programmentwicklung mehrfach zum Testen in die Steuerung geladen:

In CODESYS: [Online] > [Einloggen] > das neue Programm laden.

Für jeden derartigen Download via CODESYS wird dazu der Quellcode neu übersetzt. Daraus resultiert, dass auch jedes Mal im Speicher der Steuerung eine neue Prüfsumme gebildet wird. Auch für Sicherheitssteuerungen ist dieses Verfahren bis zur Freigabe der Software zulässig.



Grafik: Erstellen und Verteilen der Software

ifm-Maintenance-Tool nutzen

8492

Das **ifm**-Maintenance-Tool dient dem einfachen Übertragen des Programmcodes vom Programmierplatz in das Gerät. Grundsätzlich kann jedes Anwendungsprogramm mit dem **ifm**-Maintenance-Tool auf die Geräte kopiert werden. Vorteil: Dazu ist kein Programmiersystem mit einer CODESYS-Lizenz erforderlich.

Hier finden Sie das aktuelle **ifm**-Maintenance-Tool:

- www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]
- **ecomatmobile**-DVD "Software, tools and documentation" im Register "R360 tools [D/E]"

Verteilen des Anwendungsprogramms

8493

Wir empfehlen folgenden Ablauf, wenn das Anwendungsprogramm auf Serienmaschinen kopiert wird und zum Einsatz kommt:

- **Sichern der Software**
Nach Abschluss der Programmentwicklung muss die letzte Version des in das Gerät geladenen Anwendungsprogramms mit dem **ifm**-Maintenance-Tool zunächst aus dem Gerät ausgelesen und auf einem Datenträger unter dem Namen `projektdatei.RESX` gespeichert werden. Nur dieses Verfahren gewährleistet, dass das Anwendungsprogramm mit den entsprechenden Prüfsummen gesichert ist.
- **Download der Software**
Um in der Serienproduktion alle Maschinen mit einer einheitlichen Software auszurüsten, darf nur diese Datei mit dem **ifm**-Maintenance-Tool in die Geräte geladen werden.
- Ein Fehler in den Daten dieser Datei wird durch die integrierte Prüfsumme beim erneuten Laden durch das **ifm**-Maintenance-Tool automatisch erkannt.

3.4.3 Betriebszustände

1075

Nach Anlegen der Versorgungsspannung kann sich das **ecomatmobile**-Gerät in einem von fünf möglichen Betriebszuständen befinden:

- BOOTLOADER
- INIT
- STOP
- RUN
- SYSTEM STOP

INIT-Zustand (Reset)

1076

Voraussetzung: ein gültiges Laufzeitsystem ist installiert.

Dieser Zustand wird nach jedem Power-On-Reset durchlaufen:

- > Das Laufzeitsystem wird initialisiert.
- > Verschiedene Checks werden durchgeführt, z.B. Warten auf gültige Versorgungsspannung.
- > Dieser nur temporäre Zustand wird vom RUN- oder STOP-Zustand abgelöst.
- > Die LED leuchtet orange.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- RUN
- STOP

STOP-Zustand

8288

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Aus dem INIT-Zustand, wenn kein Anwendungsprogramm geladen ist.
- Aus dem RUN-Zustand, wenn folgende Bedingung erfüllt ist:
 - STOP-Kommando kommt über die CODESYS-Schnittstelle.

Im STOP-Zustand:

- > Die Ausgänge des Geräts sind abgeschaltet.
- > Das Abarbeiten des Anwendungsprogramms ist angehalten.
- > Die LED leuchtet grün.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- RUN
- ERROR
- FATAL ERROR
- INIT (nach Power-On-Reset)

RUN-Zustand

8287

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Aus dem INIT-Zustand (Autostart), wenn folgende Bedingungen erfüllt sind:
 - Die Betriebsspannung hat einen Mindestwert erreicht. UND:
 - Das Anwendungsprogramm ist vorhanden.
- Aus dem STOP-Zustand:
 - durch das CODESYS-RUN-Kommando.
 - Die Betriebsspannung hat einen Mindestwert erreicht oder überschritten.

Im RUN-Zustand:

- > Das Laufzeitsystem läuft.
- > Das Anwendungsprogramm läuft.
- > Die LED blinkt grün mit 2 Hz.
Die LED kann vom Anwendungsprogramm abweichend gesteuert werden → FB **SET_LED**
(→ Seite [170](#)).

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- INIT (nach Power-On-Reset)
- STOP
- ERROR
- FATAL ERROR

ERROR-Zustand

8290

Wechsel in diesen Zustand in folgenden Fällen möglich:

- bei zu geringer Versorgungsspannung.

Im ERROR-Zustand:

- > Die Ausgänge des Geräts sind ausgeschaltet.
- > Das Abarbeiten des Anwendungsprogramms ist angehalten.
- > Systemparameter werden gespeichert.
- > Die LED blinkt rot mit 5 Hz.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- RUN
- STOP
- FATAL ERROR
- INIT (nach Power-On-Reset)

FATAL-ERROR-Zustand

8289

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Speicherfehler (RAM / Flash)
- Ausnahmezustand
- Laufzeitsystem-Fehler

Im FATAL-ERROR-Zustand:

- > Die Ausgänge des Geräts sind abgeschaltet.
- > Das Anwendungsprogramm ist beendet.
- > Das Laufzeitsystem ist beendet.
- > Die LED leuchtet rot.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:

- INIT (nach Power-On-Reset)

3.4.4 Leistungsgrenzen des Geräts

7358



Leistungsgrenzen des Geräts beachten! → Datenblatt

Verhalten des Watchdog

15277

Ein Watchdog überwacht in diesem Gerät die Programmlaufzeit der CODESYS-Anwendung.

Wird die maximale Watchdog-Zeit (Anwendungsprogramm: 100 ms; Visualisierung: 1 200 ms) überschritten:

- > Gerät geht in den Zustand "Timeout Error"
- > alle Prozesse werden angehalten (Reset)
- > alle Ausgänge werden ausgeschaltet
- > der Bildschirm wird dunkel
- > die Status-LED blinkt rot mit 10 Hz

Störung beseitigen:

- ▶ Anwendungsprogramm löschen!
- ▶ PowerOn-Reset
- ▶ Anwendungsprogramm neu ins Gerät laden

Sollte der beschriebene Watchdog ausfallen, dann:

- > ein zweiter Watchdog führt das Gerät in den Zustand "Fatal Error"
- > die Status-LED leuchtet rot

Störung beseitigen:

- ▶ PowerOn-Reset

Wenn ohne Erfolg:

- ▶ Goto Bootloader
- ▶ PowerOn-Reset
- ▶ Laufzeitsystem neu ins Gerät laden
- ▶ Anwendungsprogramm neu ins Gerät laden

Wenn ohne Erfolg:

- ▶ Hardware-Fehler: Gerät an **ifm** einsenden!

Visualisierungsgrenzen

Inhalt	
Bildskalierung	30
Begrenzungen für Visualisierungen	30
CODESYS-Visualisierungs-Elemente	31
Zeichenbereich	32
Texte.....	32
Bewegung von Visualisierungselementen	33

8337

Embedded-Displays, wie sie z.B. in **ecomatmobile**-Geräten verbaut sind, können den vollen Farbumfang von Bitmap-Grafiken nicht zur Verfügung stellen, weil nur eingeschränkte Leistungsreserven verfügbar sind. Folgende Vorbereitungen ermöglichen trotzdem Bitmap-Bilder im Gerät:

- richtige Auswahl der Motive,
- geschicktes Verschieben von Farben oder geschickte Zusammenstellung einer Farbpalette und
- das richtige Skalieren der Bitmaps vor dem Einsatz im Gerät.

→ **Leistungsgrenzen des Geräts** (→ Seite [28](#))

8465

Parameter	Begrenzung CR0451	Begrenzung CR0452
Datei-Typ	Bitmap (*.bmp)	
Dateiname	nur Kleinbuchstaben, Namenskonvention = 8.3	
Bildgröße	320 x 240 Pixel	480 x 272 Pixel
Farben	8 Bit = 28 Farben = 256 Farben darstellbar	
Speicherbedarf	≤ 76 kByte, abhängig vom Bildinhalt bei RLE-Komprimierung	

Tabelle: Bild-Vorgaben für das Startbild

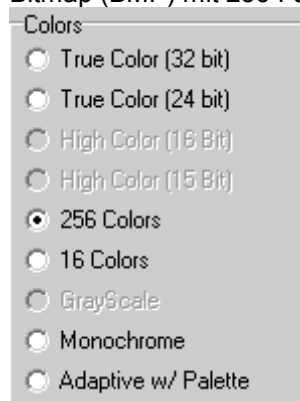
Die im Projekt eingesetzten Grafiken dürfen durchaus größer sein als die vorgegebene Bildgröße. Dann wird von dem Bild jedoch nur ein (wählbarer) Ausschnitt sichtbar sein.

8464

Farben:

Das Gerät unterstützt 2^8 (= 256) Farbabstufungen.

- Bitmap (BMP) mit 256 Farben erstellen:



Bildskalierung

3117

Wird im Gerät ein Bild geladen, welches nicht den Größen- oder den Farbanforderungen genügt, wird es vor der Anzeige auf die richtige Größe skaliert und die verwendeten Farben "überprüft".

Dieses Umrechnen eines Bildes muss bei jedem Aufruf des Bildes erneut durchgeführt werden. So verlängern sich oft die Bildwechsel-Zeiten erheblich. Abhilfe:

- ▶ Alle Umformungen des Bitmaps oder des Bildes zuvor auf dem Computer in einer Bildverarbeitung durchführen.
Nur für BasicDisplay: Die Anpassung an die Farbpalette erfolgt beim Einbinden des Bildes im Projekt durch CODESYS. Auf dem Gerät selbst werden keine Anpassungen vorgenommen (Größe, Skalierung, Farbe).
- ▶ Nur die passend gewandelten Bilder in der Visualisierung des Geräts speichern.

Begrenzungen für Visualisierungen

8319

Bezeichnung	Begrenzung
Länge von Zeichenketten	≤ 80 Zeichen
Länge von Pfadnamen	nicht relevant
Anzahl der Visualisierungsseiten	≤ 15
Anzahl der grafischen Objekte pro Visualisierungsseite	≤ 20
Anzahl der Bitmaps pro Projekt	≤ 256 abhängig von der Größe und dem verfügbaren Flash-Speicher für die Anwendung
Anzahl der Zeichensätze pro Projekt	Zeichensätze sind fest hinterlegt, nicht änderbar.
Anzahl der POUs ¹⁾ pro Projekt	≤ 512

¹⁾ POU (**P**rogram **O**rganization **U**nit) = Funktion, Funktionsbaustein oder Programmbaustein

Wegen begrenzten Speichers:

- Gruppierung von Elementen vermeiden.
- Visualisierung als Master-Folie vermeiden.
- Visualisierung von Master-Hintergrundfolien vermeiden.

CODESYS-Visualisierungselemente

453

Bitmap-Grafiken (BMP) → Kapitel **Visualisierungsgrenzen** (→ Seite [29](#))

! Nicht alle CODESYS-Funktionen können auf diesem Gerät erfolgreich arbeiten:

Visualisierungselement	Funktions-Sicherheit beim PDM	
Linienzug	o	Ein Linienzug soll aus maximal 5 Stützpunkten bestehen; nicht skalierbar Ein Linienzug soll keine Fläche umschließen.
Kurvenzug	--	wird nicht unterstützt
Rechteck	+	keine Probleme bekannt
abgerundetes Rechteck	--	wird nicht unterstützt
Kreis, Ellipse	+	keine Probleme bekannt
Polygon	o	Ein Polygon soll aus maximal 10 Stützpunkten bestehen; nicht skalierbar
Tortengrafik	--	wird nicht unterstützt
Visualisierung	--	wird nicht unterstützt
Schaltfläche	--	wird nicht unterstützt
Tabelle	--	wird nicht unterstützt
Scroll-Leiste	--	wird nicht unterstützt
Trendkurve	--	wird nicht unterstützt
Alarmtabelle	--	wird nicht unterstützt
Skalen	+	Skala als BMP-Datei erstellen
Zeigerinstrument	+	Zeigerinstrument als BMP-Datei mit überlagertem CoDeSys-Polygon darstellen
Balkendiagramm	+	Skala als BMP-Datei erstellen Wert als überlagertes CoDeSys-Rechteck darstellen
Histogramm	+	Skala als BMP-Datei erstellen Wert als überlagertes CoDeSys-Rechteck darstellen
Grafikdatei	+	bis 256 pro Projekt möglich: • BMP ¹⁾ • BMP RLE-komprimiert • TIFF ¹⁾ • JPEG ¹⁾
Grafik-Skaliermodus	o	wird unterstützt für Kreis, Ellipse, Rechteck, Linie, Linienzug, Polygon.
ActiveX-Element	--	wird nicht unterstützt
Zeigerdiagramm	--	wird nicht unterstützt
Edit-Steuerelement	--	wird nicht unterstützt

- + ohne Probleme einsetzbar
o begrenzt einsetzbar
-- nicht einsetzbar

¹⁾ Datei wird beim Einbinden ins Projekt in ein RLE-komprimiertes Bitmap umgewandelt.
Ab CODESYS-Version 2.3.9.24 wird beim Einbinden eines Bildes ein zusätzlicher Dialog geöffnet, mit dessen Hilfe die Datei an die vom Gerät verwendete Farbpalette angepasst wird. Bei der Farbumwandlung kann gewählt werden zwischen 'ähnlichste Farbe' ([Dithering] deaktivieren) und 'Dithering'.

Zeichenbereich

15987

- Der Ursprung (0,0) des virtuellen und physikalischen Zeichenbereichs markiert die linke obere Ecke.
- Virtueller Zeichenbereich (X- / Y-Koordinaten) = -32768...+32767 (vergrößert den physikalischen Zeichenbereich)
- Alle Objekte müssen in ihren Außenabmessungen auch nach Skalieren oder Verschieben innerhalb der Grenzen des virtuellen Zeichenbereichs bleiben! Ansonsten arbeitet die Visualisierung nicht mehr korrekt.
- Elemente im virtuellen Zeichenbereich werden nicht berechnet.

Texte

8436

- Die kleinste auf dem Gerät gut lesbare Schriftgröße ist 11 Punkt.
- Zulässige Schriftarten (Fonts):
 - Arial (Standard)
 - Lucida Console
- Zulässiger Schriftgrad [Pixel] und Schriftschnitt:
 - Arial: 11 (Standard), 16, 24, 32 (alle nur normal)
 - Lucida Console: 16, 24, 48*) (alle nur normal)
- *) Lucida Console in der Schriftgröße 48 enthält nur folgende Zeichen:
 - die Ziffern 0 bis 9
 - die Sonderzeichen + - . : %
 - das Leerzeichen.
- Zulässige Effekte:
 - ohne (Standard)
- Folgende Text-Skripte werden ignoriert:
 - westlich
 - hebräisch
 - arabisch
 - griechisch
 - türkisch
 - baltisch
 - mitteleuropäisch
 - kyrillisch
 - vietnamesisch

Bewegung von Visualisierungselementen

7392

Bild- und Text-Elemente können auf dem Display definiert bewegt werden.

Element-Bewegung	Beschreibung
Rotieren	Element um einen definierten Drehpunkt rotieren Rotationswinkel angeben - Rotationswinkel in [Grad] - Wert positiv = Rotation in Uhrzeigersinn - Wert negativ = Rotation entgegen Uhrzeigersinn
Verschieben	Element verschieben: - horizontal - vertikal - nur innerhalb des Zeichenbereichs - max. bis Verlassen des Zeichenbereichs
Relatives Verschieben von - Rechteck - Ellipse / Kreis	Jede Element-Kante kann über eine Variable vom Typ INT um angegebene Anzahl von Pixeln verschoben werden: - Grundposition der 4 Kanten = Null - neuer Wert verschiebt diese Kante um den angegebenen Wert Verschiebe-Richtung bei Wert > 0: - horizontale Kante nach unten - vertikale Kante nach rechts Verschiebe-Richtung bei Wert < 0: - horizontale Kante nach oben - vertikale Kante nach links

Grenzen für CAN in diesem Gerät

17975

FIFO (First In, First Out) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Einige RAW-CAN-Bausteine ermöglichen das Versenden und Empfangen mehrerer Nachrichten innerhalb eines SPS-Zyklus, da die Nachrichten in einem FiFo zwischengespeichert werden:

- CAN_TX..., → Bausteine: RAW-CAN Daten senden
- **CAN_RX_ENH_FIFO** (→ Seite [63](#))
- **CAN_RX_RANGE_FIFO** (→ Seite [67](#))

Die Anzahl der FiFo-Nachrichten ist begrenzt. Es gelten folgende Leistungsgrenzen der Geräte:

Gerät	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n
Kriterium		
max. FiFo senden - mit FB CAN_TX... - mit FB CAN_TX_ENH...	4 Nachrichten 16 Nachrichten	4 Nachrichten 16 Nachrichten
max. FiFo empfangen - mit FB CAN_RX_..._FIFO	32 Nachrichten	32 Nachrichten

Grenzen für CANopen in diesem Gerät

17976

Es gelten folgende Leistungsgrenzen der Geräte:

Gerät	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n
Kriterium		
max. Guarding-Fehler	32 Meldungen	128 Meldungen
max. SDO-Daten	2 048 Bytes	2 048 Bytes

Grenzen für CAN J1939 in diesem Gerät

17977

Es gelten folgende Leistungsgrenzen der Geräte:

Gerät	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n
Kriterium		
max. FiFo senden - mit FB J1939_TX - mit FB J1939_TX_ENH	4 Nachrichten 16 Nachrichten	4 Nachrichten 16 Nachrichten
max. FiFo empfangen - mit FB J1939_RX_FIFO	32 Nachrichten	32 Nachrichten
max. DTCs	64 Meldungen	64 Meldungen
max. Daten J1939	1 785 Bytes	1 785 Bytes

4 Konfigurationen

Inhalt

Laufzeitsystem einrichten	35
Programmiersystem einrichten	38
Funktionskonfiguration, allgemein	43
Variablen	44

Die in den jeweiligen Montage- und Installationsanweisungen oder dem **Anhang** (→ Seite [186](#)) dieser Dokumentation beschriebenen Gerätekonfigurationen stehen als Standardgeräte (Lagerware) zur Verfügung. Diese decken bei den meisten Anwendungen die geforderten Spezifikationen ab.

Entsprechend den Kundenanforderungen bei Serieneinsatz ist es aber auch möglich, dass andere Gerätekonfigurationen z.B. hinsichtlich der Zusammenstellung der Ein- und Ausgänge und der Ausführung der Analogkanäle eingesetzt werden.

4.1 Laufzeitsystem einrichten

Inhalt

Laufzeitsystem neu installieren	36
Laufzeitsystem aktualisieren	37
Installation verifizieren	37

14091

4.1.1 Laufzeitsystem neu installieren

14635
8486

Im Auslieferungszustand ist im Normalfall kein Laufzeitsystem im Gerät geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Bootloader aktiv. Dieser stellt die minimalen Funktionen für den Laufzeitsystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. RS232, CAN).

Der Laufzeitsystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Anwendungsprogramm kann anschließend (auch mehrmals) in das Gerät geladen werden, ohne das Laufzeitsystem zu beeinflussen.

Das Laufzeitsystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der **ifm electronic gmbh** heruntergeladen werden:

→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]

ACHTUNG

Gefahr von Datenverlust!

Bei Spannungsausfall während der Datenübertragung können Daten verloren gehen, so dass das Gerät nicht mehr funktionsfähig ist. Reparatur ist nur bei **ifm electronic** möglich.

► Für ständige Spannungsversorgung während der Datenübertragung sorgen!

! HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:

- des Laufzeitsystems (ifm_CR0451_Vxxyzz.RESX),
- der Steuerungskonfiguration (ifm_CR0451_Vxx.CFG),
- der Gerätebibliothek (ifm_CR0451_Vxxyzz.LIB) und
- der weiteren Dateien

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "CR0451") und die Software-Versionsnummer "xx" (z.B. "01") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

4368

! Folgende Dateien müssen ebenfalls geladen sein:

- die zum Projekt erforderlichen internen Bibliotheken (in IEC 61131 erstellt),
- die Konfigurationsdateien (*.CFG)
- und die Target-Dateien (*.TRG).

! Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CODESYS nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

Das Laufzeitsystem wird mit dem eigenständigen Programm "Maintenance Tool" in das Gerät übertragen. (Das Programm befindet sich auf der **ecomatmobile**-DVD "Software, tools and documentation" oder kann bei Bedarf von der **ifm**-Homepage heruntergeladen werden:

→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen])

Das Anwendungsprogramm wird im Normalfall über das Programmiersystem in das Gerät geladen. Es kann aber ebenfalls mit dem "Maintenance Tool" geladen werden, wenn es zuvor aus dem Gerät ausgelesen wurde.

4.1.2 Laufzeitsystem aktualisieren

13269

Auf dem Gerät ist bereits ein älteres Laufzeitsystem installiert. Nun möchten Sie das Laufzeitsystem auf dem Gerät aktualisieren?

14158

ACHTUNG

Gefahr von Datenverlust!

Beim Löschen oder Aktualisieren des Laufzeitsystems werden alle Daten und Programme auf dem Gerät gelöscht.

- ▶ Alle erforderlichen Daten und Programme sichern, bevor das Laufzeitsystem gelöscht oder aktualisiert wird!

3084

Immer, wenn es zu wesentlichen Verbesserungen in der Betriebssystem-Software oder des CODESYS-Laufzeitsystems kommt, gibt **ifm** davon eine neue Version heraus. Die Versionen werden fortlaufend durchnummeriert (V01, V02, V03, ...).

Welche neuen Zusatzfunktionen die neue Softwareversion enthält, entnehmen Sie bitte der jeweiligen Dokumentation. Beachten Sie, ob in der Dokumentation auf besondere Anforderungen an die Hardware-Version hingewiesen wird.

Wenn Sie im Besitz eines Gerätes mit einer älteren Version sind und wenn die Bedingungen für die Hardware und Ihr Projekt stimmen, können Sie Ihr Gerät durch Aktualisieren der Software auf den neuen Software-Stand bringen.

Prinzipiell gelten für diesen Vorgang die gleichen Hinweise, wie zuvor im Kapitel 'Laufzeitsystem neu installieren' gegeben wurden.

4.1.3 Installation verifizieren

14637

- ▶ Nach dem Laden des Laufzeitsystems in die Steuerung:
 - Prüfen, ob das Laufzeitsystem korrekt übertragen wurde!
 - Prüfen, ob sich das richtige Laufzeitsystem auf der Steuerung befindet!
- ▶ 1. Prüfung:

mit dem **ifm**-Maintenance-Tool prüfen, ob die richtige Laufzeitsystem-Version geladen wurde:

 - Name und Version des Laufzeitsystems im Gerät auslesen!
 - Diese Daten manuell mit den Soll-Daten vergleichen!
- ▶ 2. Prüfung (optional):

Im Anwendungsprogramm prüfen, ob die richtige Laufzeitsystem-Version geladen wurde:

 - Name und die Version des Laufzeitsystems im Gerät auslesen!
 - Diese Daten mit fest vorgegebenen Werten vergleichen!

Zum Auslesen der Daten dient folgender FB:

GET_SW_INFO (→ Seite [162](#))

liefert Informationen über die System-Software des Geräts:

- Software-Name,
- Software-Version,
- Build-Nummer,
- Build-Datum

4.2 Programmiersystem einrichten

Inhalt

Programmierensystem manuell einrichten	38
Programmierensystem über Templates einrichten	42

14461

4.2.1 Programmiersystem manuell einrichten

Inhalt

Target einrichten	39
Steuerungskonfiguration aktivieren	40
CAN-Schnittstellen deklarieren (z.B. CR1080)	41

3963

Target einrichten

13136
11379

Beim Erstellen eines neuen Projektes in CODESYS muss die dem Gerät entsprechende Target-Datei geladen werden.

- ▶ Im Dialog-Fenster [Zielsystem Einstellungen] im Menü [Konfiguration] die gewünschte Target-Datei wählen.
- > Die Target-Datei stellt für das Programmiersystem die Schnittstelle zur Hardware her.
- > Gleichzeitig mit Wahl des Targets werden automatisch einige wichtige Bibliotheken und die Steuerungskonfiguration geladen.
- ▶ Bei Bedarf im Fenster [Zielsystem Einstellungen] > Reiter [Netzfunktionen] > [Parameter-Manager unterstützen] und / oder [Netzvariablen unterstützen] aktivieren.
- ▶ Bei Bedarf geladene (3S-)Bibliotheken wieder entfernen oder durch weitere (ifm-)Bibliotheken ergänzen.
- ▶ Immer die passende Geräte-Bibliothek `ifm_CR0451_Vxxyyyz.LIB` manuell ergänzen!

! HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:

- des Laufzeitsystems (`ifm_CR0451_Vxxyyyz.RESX`),
- der Steuerungskonfiguration (`ifm_CR0451_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CR0451_Vxxyyyz.LIB`) und
- der weiteren Dateien

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "CR0451") und die Software-Versionsnummer "xx" (z.B. "01") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

4368

! Folgende Dateien müssen ebenfalls geladen sein:

- die zum Projekt erforderlichen internen Bibliotheken (in IEC 61131 erstellt),
- die Konfigurationsdateien (*.CFG)
- und die Target-Dateien (*.TRG).

i Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CODESYS nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

Steuerungskonfiguration aktivieren

10079

Die Steuerungskonfiguration wird automatisch zusammen mit dem Zielsystem geladen. Die Steuerungskonfiguration bildet den Inhalt der Datei CR0451.cfg in CODESYS ab. Der Programmierer hat dadurch einfachen Zugriff auf vordefinierte System- und Fehlermerker, Ein- und Ausgänge sowie die CAN-Schnittstellen des Geräts.

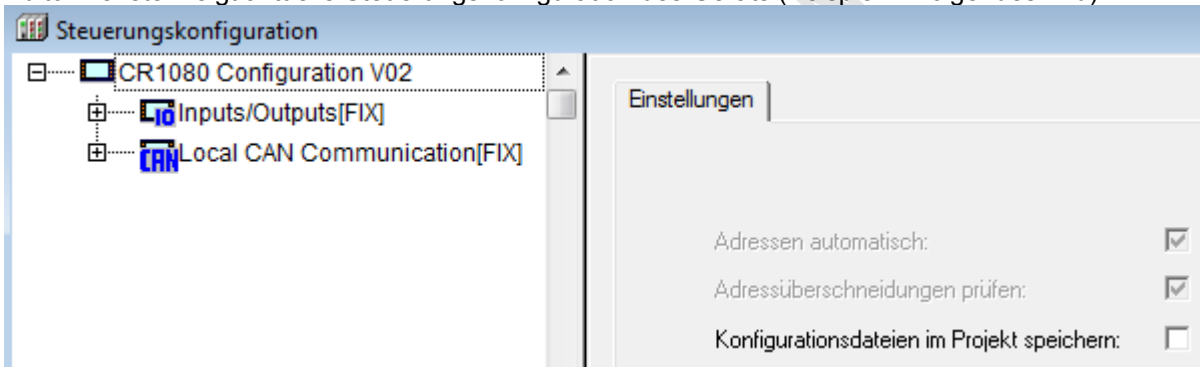
Um auf die Steuerungskonfiguration zuzugreifen (Bsp.: CR1080):

- ▶ In CODESYS den Reiter [Ressourcen] wählen:



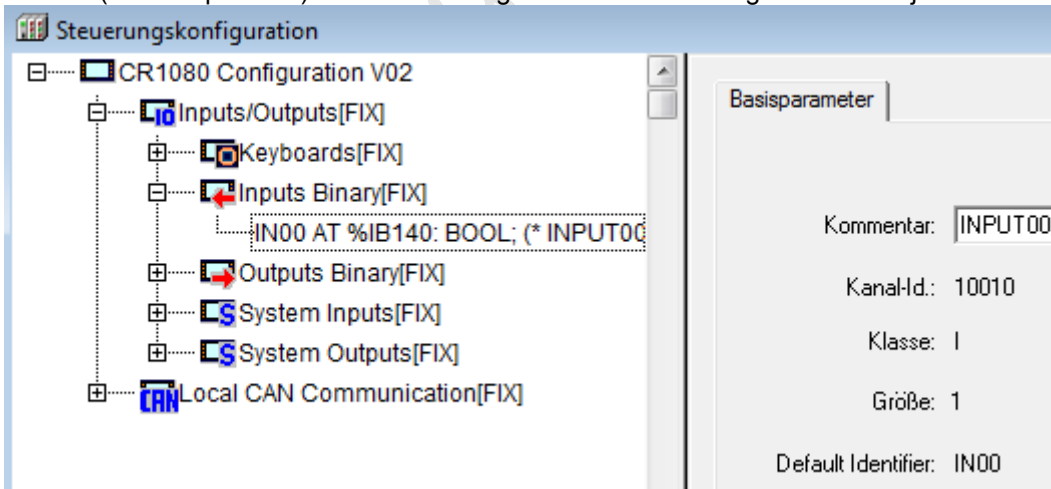
- ▶ Im Ressourcen-Baum: Doppelklick auf [Steuerungskonfiguration]

- > Editor-Fenster zeigt aktuelle Steuerungskonfiguration des Geräts (Beispiel → folgendes Bild):



Der Programmierer hat über die Steuerungskonfiguration einfachen Zugriff auf folgende Komponenten:

- System- und Fehlermerker
Je nach Anwendung und Anwendungsprogramm müssen diese Merker bearbeitet und ausgewertet werden. Der Zugriff erfolgt über deren symbolischen Namen.
- Struktur der Ein- und Ausgänge
Diese können im Fenster [Steuerungskonfiguration] (→ Bild unten) direkt symbolisch bezeichnet werden (sehr empfohlen!) und stehen als globale Variablen im gesamten Projekt zur Verfügung.

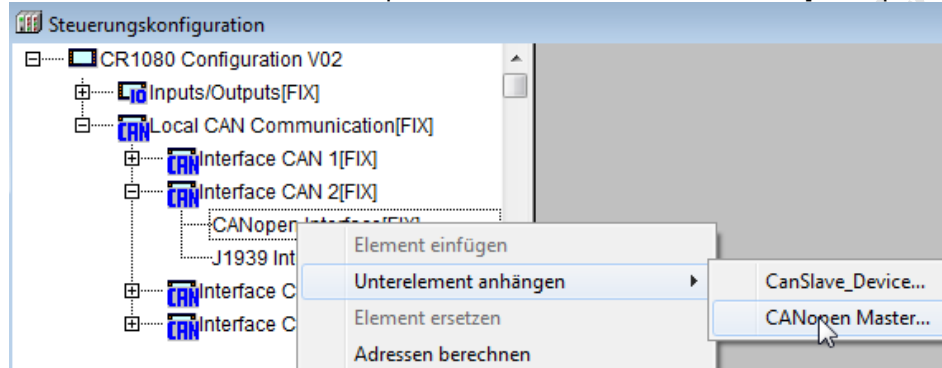


CAN-Schnittstellen deklarieren (z.B. CR1080)

10080

In der CODESYS-Steuerungskonfiguration müssen Sie nun die CAN-Schnittstelle(n) deklarieren.

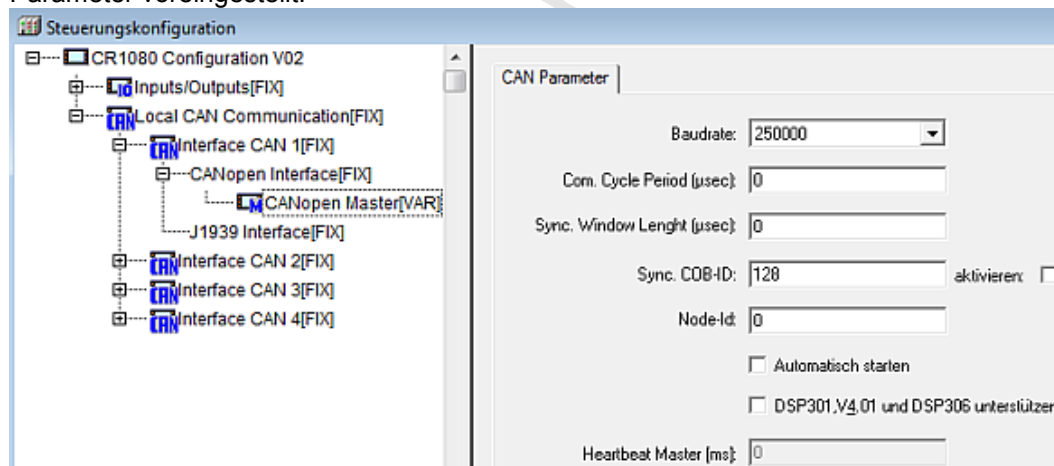
- ▶ Rechtsklick auf [CANopen Interface [FIX]] der gewünschten CAN-Schnittstelle.
- ▶ Klick auf [Unterelement anhängen].
- ▶ Auch wenn das Gerät als CANopen-Slave betrieben wird: Klick auf [CANopen Master...]:



Info

Wenn das Gerät als Slave betrieben wird, wäre die Auswahl [CanSlave_Device] ebenfalls möglich. Bei der insgesamt einfacheren Konfiguration als Master können auch alle CAN-Layer2- und Netzwerkvariablen-Funktionen genutzt werden.

- > Die CAN-Parameter der Steuerungskonfiguration erscheinen. Hier sind bereits einige CAN-Parameter voreingestellt:



- ▶ Wenn das Gerät via Netzwerkvariablen oder CAN_RX / CAN_TX auf CAN-Layer 2 oder als Slave betrieben wird:
 - ❗ Prüfen, ob für das Gerät die richtige Baudrate eingestellt ist (Baudrate muss für alle Teilnehmer identisch sein).
- ▶ Wenn das Gerät als CANopen-Master betrieben wird:
 - Alle Parameter-Einstellungen prüfen.
- ▶ Das Fenster [Steuerungskonfiguration] schließen.
- ▶ Mit Menü [Datei] > [speichern unter...] dem Projekt einen sinnvollen Namen geben und das Projekt im gewünschten Verzeichnis speichern.
- ▶ ❗ Im Anwendungsprogramm für jede CAN-Schnittstelle immer eine eigene Instanz des FBs **CANOPEN_ENABLE** (→ Seite 80) anlegen!

4.2.2 Programmiersystem über Templates einrichten

13745

ifm bietet vorgefertigte Templates (Programm-Vorlagen), womit Sie das Programmiersystem schnell, einfach und vollständig einrichten können.

970

-  Beim Installieren der **ecomatmobile**-DVD "Software, tools and documentation" wurden auch Projekte mit Vorlagen auf Ihrem Computer im Programmverzeichnis abgelegt:
...\\ifm electronic\\CoDeSys V...\\Projects\\Template_DVD_V...
- ▶ Die gewünschte dort gespeicherte Vorlage in CODESYS öffnen mit:
[Datei] > [Neu aus Vorlage...]
 - > CODESYS legt ein neues Projekt an, dem der prinzipielle Programmaufbau entnommen werden kann. Es wird dringend empfohlen, dem gezeigten Schema zu folgen.

4.3 Funktionskonfiguration, allgemein

Inhalt

Systemvariablen	43
-----------------------	----

3971

4.3.1 Systemvariablen

15576

Alle Systemvariablen (→ Kapitel **Systemmarker** (→ Seite [186](#))) liegen auf festen, nicht verschiebbaren Adressen.

4.4 Variablen

Inhalt

Retain-Variablen	44
Netzwerkvariablen	45

3130

In diesem Kapitel erfahren Sie mehr über den Umgang mit Variablen.

14486

Das Gerät unterstützt folgende Variablentypen:

Variable	Deklarationsort	Gültigkeitsbereich	Speicherverhalten
lokal	im Deklarationsteil des Bausteins	gilt nur im Baustein (POU), in dem sie konfiguriert wurde	flüchtig
lokal Retain			nicht flüchtig
global	in [Ressourcen] > [Globale Variablen] > [Globale_Variablen]	gilt in allen Bausteinen (POUs) dieses CODESYS-Projekts	flüchtig
global Retain			nicht flüchtig
Netzwerk	in [Ressourcen] > [Globale Variablen] > Deklarationsliste	Werte stehen allen CODESYS-Projekten im gesamten Netzwerk zur Verfügung, wenn die Variable in ihren Deklarationslisten enthalten ist.	flüchtig
Netzwerk Retain			nicht flüchtig



→ CODESYS-Programmierhandbuch
→ **ecomatmobile**-DVD "Software, tools and documentation"

4.4.1 Retain-Variablen

8672

Retain-Variablen werden automatisch in einen geschützten Speicherbereich gesichert und automatisch bei Neustart wieder geladen.

14166

Typische Einsätze für Retain-Variablen sind z.B.:

- Betriebsstunden, die zur Laufzeit der Maschine fortgeschrieben werden,
- Positionswerte von Inkrementalgebern,
- im Bildschirmgerät eingetragene Sollwerte,
- Maschinenparameter,

also alle Variablen, deren Werte beim Ausschalten des Geräts nicht verloren gehen dürfen.

Als Retain können alle Variablentypen, auch komplexe Strukturen (z.B. Timer), gekennzeichnet werden.

► Dazu in der Variablen-Deklaration das Kontrollfeld [RETAIN] aktivieren (→ Bild).

Sichern von Retain-Variablen

9853

Im Gerät werden auch die Daten vom Typ RETAIN zur Laufzeit nur im flüchtigen Speicher (RAM) abgelegt.

Um sie dauerhaft zu sichern, werden sie am Ende jedes Zyklus automatisch in den FRAM-Speicher ¹⁾ geschrieben.

¹⁾ FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

! HINWEIS

In diesem Gerät die folgenden Funktionen aus der 3S-Bibliothek SysLibPlcCtrl.lib NICHT einsetzen:

- FUN SysSaveRetains
- FUN SysRestoreRetains

Rücklesen von Retain-Variablen

9854

Nach dem Einschalten und vor dem ersten Programmzyklus schreibt das Gerät die gesicherten Daten einmalig automatisch zurück in den Arbeitsspeicher. Dazu müssen keine zusätzlichen Bausteine in das Anwendungsprogramm integriert werden.

! HINWEIS

In diesem Gerät die folgenden Funktionen aus der 3S-Bibliothek SysLibPlcCtrl.lib NICHT einsetzen:

- FUN SysSaveRetains
- FUN SysRestoreRetains

4.4.2 Netzwerkvariablen

15242
9856

Globale Netzwerkvariablen dienen dem Datenaustausch zwischen Controllern im Netzwerk. Die Werte von globalen Netzwerkvariablen stehen allen CODESYS-Projekten im gesamten Netzwerk zur Verfügung, wenn die Variablen in deren Deklarationslisten enthalten sind.

► Dazu folgende Bibliothek(en) in das CODESYS-Projekt einbinden:

- 3S_CANopenNetVar.lib
- ifm_NetVarLib_NT_Vxxyyzz.lib

5 ifm-Funktionselemente

Inhalt

ifm-Bibliotheken für das Gerät CR0451	46
ifm-Bausteine für das Gerät CR0451	51
	13586

Alle CODESYS-Funktionselemente (FBs, PRGs, FUNs) sind in Bibliotheken zusammengefasst. Nachfolgend zeigen wir Ihnen alle **ifm**-Bibliotheken, die Sie zusammen mit diesem Gerät nutzen können.

Anschließend finden Sie eine thematisch gegliederte Beschreibung der Funktionselemente.

5.1 ifm-Bibliotheken für das Gerät CR0451

Inhalt

Mindestens erforderliche Bibliotheken	46
Bibliothek ifm_CR0451_Vxxyzz.LIB	47
Bibliothek ifm_CR0451_Init_Vxxyzz.LIB	47
Bibliothek ifm_PDMsmart_util_Vxxyzz.LIB	48
Bibliothek ifm_RAWCan_NT_Vxxyzz.LIB	48
Bibliothek ifm_CANopen_NT_Vxxyzz.LIB	49
Bibliothek ifm_J1939_NT_Vxxyzz.LIB	50
	14235

Legende für ..._Vxxyzz.LIB:

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Hier finden Sie die für dieses Gerät passenden **ifm**-Funktionselemente aufgelistet, nach CODESYS-Bibliotheken sortiert.

5.1.1 Mindestens erforderliche Bibliotheken

15300

Wenn Sie die Programmierung dieses Geräts nicht über ein **ifm**-Template beginnen möchten, sollten Sie mindestens folgende Bibliotheken in Ihr Projekt einbinden:

ifm_CR0451_Vxxyzz.LIB	Gerätebibliothek
ifm_CR0451_Init_Vxxyzz.LIB	initialisiert den Gerätebildschirm

Für Netzwerkvariablen erforderliche Bibliotheken

15304

Wenn Sie mit Netzwerkvariablen arbeiten möchten, benötigen Sie zusätzlich folgende Bibliotheken:

ifm_NetVarLib_NT_Vxxyzz.LIB	Unterstützung von Netzwerkvariablen (ifm-Bibliothek)
3S_CANopenNetVar.LIB	Unterstützung von Netzwerkvariablen (3S-Bibliothek)

5.1.2 Bibliothek ifm_CR0451_Vxxyzz.LIB

15284

Dies ist die Geräte-Bibliothek.

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
FLASH_INFO (→ Seite 157)	liest die Informationen aus dem User-Flash-Speicher: • Name des Speicherbereichs (vom User vorgegeben), • Software-Version, • Startadresse (für einfaches Lesen mit IEC-Struktur)
FLASH_READ (→ Seite 158)	liest unterschiedliche Datentypen direkt aus dem Flash-Speicher in den RAM
GET_APP_INFO (→ Seite 159)	liefert Informationen über das im Gerät gespeicherte Anwendungsprogramm: • Name der Anwendung, • Version der Anwendung, • eindeutige CODESYS-Build-Nummer, • CODESYS-Build-Datum
GET_HW_INFO (→ Seite 160)	liefert Informationen über die Hardware des Geräts: • ifm-Artikelnnummer (z.B. CR0403), • Artikelbezeichnung, • eindeutige Seriennummer, • Hardware-Revision, • Produktionsdatum
GET_IDENTITY (→ Seite 161)	liest die im Gerät gespeicherte Kennung der Anwendung (wurde zuvor mit SET_IDENTITY (→ Seite 169) gespeichert)
GET_SW_INFO (→ Seite 162)	liefert Informationen über die System-Software des Geräts: • Software-Name, • Software-Version, • Build-Nummer, • Build-Datum
GET_SW_VERSION (→ Seite 163)	liefert Informationen über die Versionen der im Gerät gespeicherten Software: • Basic-System-Version, • Bootloader-Version, • SIS-Version, • Anwendungsprogramm-Version, • User-Flash-Version
MEM_ERROR (→ Seite 164)	meldet Fehler in einigen Parametern oder im Speicher (Re-)Initialisierung von Systemressourcen
MEMCPY (→ Seite 165)	schreibt und liest unterschiedliche Datentypen direkt in den Speicher
OHC (→ Seite 167)	parametrierbarer Betriebsstunden-Zähler (0...3)
SET_IDENTITY (→ Seite 169)	setzt eine anwendungsspezifische Programmkennung
SET_LED (→ Seite 170)	im Anwendungsprogramm Frequenz und Farbe der Status-LED ändern
SET_PASSWORD (→ Seite 172)	setzt Benutzerkennung für Zugangskontrolle bei Programm- und Speicher-Upload
TIMER_READ_US (→ Seite 173)	liest die aktuelle Systemzeit in [µs] aus Max-Wert = 1h 11min 34s 967ms 295µs

5.1.3 Bibliothek ifm_CR0451_Init_Vxxyzz.LIB

15286

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
BASICDISPLAY_INIT (→ Seite 181)	initialisiert im ersten SPS-Zyklus den Bildschirm des BasicDisplays

5.1.4 Bibliothek ifm_PDMsmart_util_Vxxyzz.LIB

15289

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
GET_TEXT_FROM_FLASH (→ Seite 175)	liest Texte vom Typ STRING mittels FLASH_READ (→ Seite 158) aus dem Flash-Speicher
NORM_DINT (→ Seite 177)	normiert einen Wert [DINT] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen
NORM_REAL (→ Seite 178)	normiert einen Wert [REAL] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen
PDM_PAGECONTROL (→ Seite 182)	steuert den Aufruf bestimmter Visualisierungsseiten
TOGGLE (→ Seite 179)	Setzen und Rücksetzen einer boolschen Variablen mit nur einem Eingangs-Bit

5.1.5 Bibliothek ifm_RAWCan_NT_Vxxyzz.LIB

14715

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CAN_ENABLE (→ Seite 54)	initialisiert die angegebene CAN-Schnittstelle parametriert die CAN-Baudrate
CAN_RECOVER (→ Seite 55)	das automatische BusOff-Handling aktivieren / deaktivieren bei Vorliegen eines BusOff die CAN-Schnittstelle wieder neu starten
CAN_REMOTE_REQUEST (→ Seite 76)	eine entsprechende Anforderung senden und die Antwort des anderen Gerätes als Ergebnis zurückliefern
CAN_REMOTE_RESPONSE (→ Seite 77)	stellt dem CAN-Controller im Gerät Daten zur Verfügung, die automatisch auf die Anfrage einer Remote-Nachricht gesendet werden
CAN_RX (→ Seite 60)	konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus
CAN_RX_ENH (→ Seite 61)	• konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus • Frame-Typ und Maske wählbar
CAN_RX_ENH_FIFO (→ Seite 63)	• konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus • Frame-Typ und Maske wählbar • mehrere CAN-Telegramme je Zyklus möglich
CAN_RX_RANGE (→ Seite 65)	• konfiguriert einen Bereich von Datenempfangsobjekten und liest den Empfangspuffer der Datenobjekte aus • Frame-Typ und Maske wählbar
CAN_RX_RANGE_FIFO (→ Seite 67)	• konfiguriert einen Bereich von Datenempfangsobjekten und liest den Empfangspuffer der Datenobjekte aus • Frame-Typ und Maske wählbar • mehrere CAN-Telegramme je Zyklus möglich
CAN_SETDOWNLOADID (→ Seite 56)	= Set CAN-Download-ID stellt den Download-Identifizier für die CAN-Schnittstelle ein
CAN_STATUS (→ Seite 57)	Informationen zum gewählten CAN-Bus abfragen: BAUDRATE, DOWNLOAD_ID, BUSOFF, WARNING_RX, WARNING_TX, VERSION, BUSLOAD und bei Bedarf zurücksetzen: BUSOFF, WARNING_RX, WARNING_TX
CAN_TX (→ Seite 70)	übergibt in jedem Aufruf ein CAN-Datenobjekt (Message) an die parametrierte CAN-Schnittstelle zur Übertragung
CAN_TX_ENH (→ Seite 71)	übergibt in jedem Aufruf ein CAN-Datenobjekt (Message) an die parametrierte CAN-Schnittstelle zur Übertragung CAN-spezifische Eigenschaften sind einstellbar
CAN_TX_ENH_CYCLIC (→ Seite 73)	übergibt zyklisch ein CAN-Datenobjekt (Message) an die parametrierte CAN-Schnittstelle zur Übertragung CAN-spezifische Eigenschaften sind einstellbar

5.1.6 Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

14914

Diese ifm-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CANOPEN_ENABLE (→ Seite 80)	initialisiert die angegebene CANopen-Master-Schnittstelle parametriert die CAN-Baudrate
CANOPEN_GETBUFFERFLAGS (→ Seite 82)	= CANopen Get Bufferflags liefert Informationen zu den Buffer-Flags Über optionale Eingänge können die Flags zurückgesetzt werden.
CANOPEN_GETEMCYMESSAGES (→ Seite 119)	= Get CANopen Emergency Messages listet alle Emergency-Nachrichten auf, die die Steuerung seit dem letzten Löschen der Nachrichten von anderen Knoten am Netz empfangen hat Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden.
CANOPEN_GETERRORREGISTER (→ Seite 121)	= Get CANopen Error-Register liest die Fehler-Register 0x1001 und 0x1003 der Steuerung aus Die Register können durch Setzen der entsprechenden Eingänge zurückgesetzt werden.
CANOPEN_GETGUARDHBERRLIST (→ Seite 115)	= Get CANopen-Guard and Heartbeat Error-List listet in einem Array alle Knoten auf, für die der Master einen Fehler erkannt hat: Guarding-Fehler, Heartbeat-Fehler Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden.
CANOPEN_GETGUARDHBSTATSLV (→ Seite 116)	= CANopen-Slave get Guard and Heartbeat State meldet der Steuerung im Slave-Betrieb folgende Zustände: Node-Guarding-Überwachung, Heartbeat-Überwachung Die gemeldeten Fehler können durch Setzen des entsprechenden Eingangs zurückgesetzt werden.
CANOPEN_GETNMTSTATESLAVE (→ Seite 89)	= CANopen-Slave get Network Management State meldet den Netzwerk-Betriebszustand des Knotens
CANOPEN_GETODCHANGEDFLAG (→ Seite 93)	= Get Object Directory Changed Flag meldet wenn bei einem bestimmten Objektverzeichnis-Eintrag der Wert geändert wurde
CANOPEN_GETSTATE (→ Seite 84)	= CANopen Set State Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz abfragen
CANOPEN_GETSYNCSTATE (→ Seite 111)	= CANopen get SYNC State • liest die Einstellung der SYNC-Funktionalität (aktiv / deaktiv), • liest den Fehlerzustand der SYNC-Funktionalität (SyncError)
CANOPEN_NMTSERVICES (→ Seite 90)	= CANopen Network Management Services aktualisiert den internen Knotenstatus sowie abhängig von den NMT-Kommando-Eingängen: • löst ein NMT-Kommando aus oder • löst die Initialisierung eines Knotens aus
CANOPEN_READOBJECTDICT (→ Seite 94)	= CANopen Read Object Directory liest Konfigurationsdaten aus dem Objektverzeichnis des Geräts
CANOPEN_SDOREAD (→ Seite 98)	= CANopen Read SDO liest ein "expedited SDO" = beschleunigtes Nachrichten-Objekt mit Servicedaten
CANOPEN_SDOREADBLOCK (→ Seite 100)	= CANopen Read SDO Block liest den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer
CANOPEN_SDOREADMULTI (→ Seite 102)	= CANopen Read SDO Multi liest den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz
CANOPEN_SDOWRITE (→ Seite 104)	= SDO Write schreibt ein "expedited SDO" = beschleunigtes Nachrichten-Objekt mit Servicedaten
CANOPEN_SDOWRITEBLOCK (→ Seite 106)	= CANopen Write SDO Block schreibt in den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer
CANOPEN_SDOWITEMULTI (→ Seite 108)	= CANopen Write SDO Multi schreibt in den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz

Baustein	Kurzbeschreibung
CANOPEN_SENDEMCYMESSAGE (→ Seite 122)	= CANOpen Send Emergency-Message versendet eine EMCY-Nachricht. Die Nachricht wird aus den entsprechenden Parametern zusammengebaut und ins Register 0x1003 eingetragen
CANOPEN_SETSTATE (→ Seite 86)	= CANOpen Set State Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz setzen
CANOPEN_SETSYNCSTATE (→ Seite 113)	= CANOpen Set SYNC State die SYNC-Funktionalität ein- und ausschalten
CANOPEN_WRITEOBJECTDICT (→ Seite 95)	= CANOpen Write Object Directory schreibt Konfigurationsdaten in das Objektverzeichnis des Geräts

5.1.7 Bibliothek ifm_J1939_NT_Vxxyzz.LIB

14912

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
J1939_DM1RX (→ Seite 149)	J1939 Diagnostic Message 1 RX empfängt Diagnosemeldungen DM1 oder DM2 von anderen ECUs
J1939_DM1TX (→ Seite 151)	J1939 Diagnostic Message 1 TX eine aktive Fehlermeldung an den CAN-Stack übergeben
J1939_DM1TX_CFG (→ Seite 154)	J1939 Diagnostic Message 1 TX configurable CAN-Stack sendet <u>keine</u> zyklischen DM1 "zero active faults"-Nachrichten
J1939_DM3TX (→ Seite 155)	J1939 Diagnostic Message 3 TX löscht inaktive DTCs (DM2) auf einem Gerät
J1939_ENABLE (→ Seite 125)	Initialisiert den J1939-Stack
J1939_GETDABYNAME (→ Seite 127)	= Get Destination Arbitrary Name anhand der Namensinformation die Ziel-Adresse eines oder mehrerer anderer Teilnehmer bestimmen
J1939_NAME (→ Seite 129)	dem Gerät einen Name geben, mit dem es sich im Netzwerk identifiziert
J1939_RX (→ Seite 136)	empfängt eine Single-Frame-Nachricht zeigt die zuletzt auf dem CAN-Bus gelesene Nachricht
J1939_RX_FIFO (→ Seite 137)	= J1939 RX with FIFO empfängt alle spezifizierten Nachrichten und liest sie nacheinander aus einem FIFO
J1939_RX_MULTI (→ Seite 139)	= J1939 RX Multiframe Message empfängt Multiframe-Nachrichten
J1939_SPEC_REQ (→ Seite 133)	= J1939 Specific Request fragt eine spezifizierte Nachricht bei einer anderen Steuerung an und empfängt sie
J1939_SPEC_REQ_MULTI (→ Seite 134)	= J1939 Specific Request Multiframe Message fragt eine spezifizierte Multiframe-Nachricht bei einer anderen Steuerung an und empfängt sie
J1939_STATUS (→ Seite 131)	zeigt relevante Informationen zum J1939-Stack
J1939_TX (→ Seite 141)	versendet einzelne Single-Frame-Nachrichten
J1939_TX_ENH (→ Seite 142)	= J1939 TX enhanced versendet einzelne Single-Frame-Nachrichten zusätzlich einstellbar: Sende-Priorität, Datenlänge
J1939_TX_ENH_CYCLIC (→ Seite 144)	= J1939 TX enhanced cyclic versendet zyklisch Single-Frame-Nachrichten zusätzlich einstellbar: Sende-Priorität, Datenlänge, Periode
J1939_TX_ENH_MULTI (→ Seite 146)	= J1939 TX enhanced Multiframe Message versendet einzelne Multiframe-Nachrichten

5.2 ifm-Bausteine für das Gerät CR0451

Inhalt

Baustein-Ausgänge	52
Bausteine: RAW-CAN (Layer 2)	53
Bausteine: CANopen	79
Bausteine: SAE J1939	124
Bausteine: System	156
Bausteine: Grafik	174

13988
3826

Hier finden Sie die Beschreibung der für dieses Gerät passenden **ifm**-Funktionselemente, nach Thema sortiert.

5.2.1 Baustein-Ausgänge

8354
7556

Einige Bausteine geben eine Ergebnis-Meldung RESULT zurück.

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1...31		Globale Rückgabewerte; Beispiele:
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
4	04	FB ist in der Bearbeitung – Daten werden zyklisch bearbeitet
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
6	06	FB ist in der Bearbeitung – Senden läuft noch
7	07	FB ist in der Bearbeitung – Remote für ID aktiv
8	08	Funktionsbaustein ist aktiv
14	0E	FB ist aktiv CANopen-Manager konfiguriert Devices und sendet SDOs
15	0F	FB ist aktiv CANopen-Manager ist gestartet
32...63		FB-spezifische Rückgabewerte
64...127		FB-spezifische Fehlermeldungen
128...255		Globale Fehlermeldungen; Beispiele:
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Manager konnte nicht gestartet werden
240	F0	Fehler: Mehrere modale Eingänge sind aktiv z.B. CANopen-NMT-Services
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich
247	F7	Fehler: Speicherüberschreitung (Länge größer als Array)
250	FA	Fehler: FiFo ist voll – Daten wurden verloren
252	FC	Fehler: CAN-Multiframe-Sendung ist fehlgeschlagen
253	FD	Fehler: CAN-Sendung ist fehlgeschlagen. Daten können nicht gesendet werden.
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

5.2.2 Bausteine: RAW-CAN (Layer 2)

Inhalt

Bausteine: RAW-CAN Status	53
Bausteine: RAW-CAN Daten empfangen	59
Bausteine: RAW-CAN Daten senden	69
Bausteine: RAW-CAN Remote	75

15051

Hier beschreiben wir die RAW-CAN-Funktionsbausteine (CAN Layer 2) der **ifm electronic** zur Nutzung im Anwendungsprogramm.

Bausteine: RAW-CAN Status

Inhalt

CAN_ENABLE	54
CAN_RECOVER	55
CAN_SETDOWNLOADID	56
CAN_STATUS	57

15049

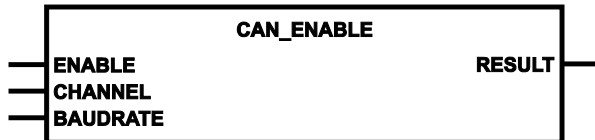
CAN_ENABLE

7492

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyzz.LIB`

Symbol in CODESYS:



Beschreibung

7494

Mit CAN_ENABLE wird die CAN-Hardware initialisiert. Ohne diesen Aufruf sind keine anderen Aufrufe im RAW-CAN möglich oder liefern einen Fehler zurück.

Zum Ändern der Baudrate ist folgender Ablauf erforderlich:

- ▶ Funktionsbaustein einen Zyklus lang auf ENABLE=FALSE halten.
- > Alle Protokolle werden zurückgesetzt.
- > Re-Initialisierung der CAN-Schnittstelle und auch der darauf laufenden CAN-Protokolle. Alle zum zyklischen Senden vorhandenen Informationen gehen dabei ebenfalls verloren und müssen neu aufgesetzt werden.
- > Beim erneuten ENABLE=TRUE wird die neue Baudrate übernommen.

Parameter der Eingänge

7495

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: CAN-Schnittstelle freigeben FALSE: CAN-Schnittstelle sperren
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
BAUDRATE	WORD := 250	Baudrate [kBit/s] zulässig = 20, 50, 100, 125, 250, 500, 800, 1000

Parameter der Ausgänge

8530

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	Funktionsbaustein ist aktiv
9 09	CAN ist nicht aktiv
242 F2	Fehler: Einstellung ist nicht möglich

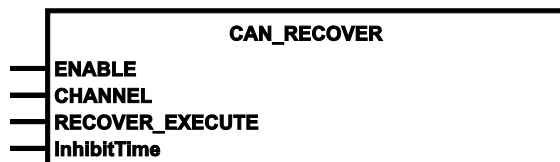
CAN_RECOVER

7512

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7513

CAN_RECOVER hat folgende Aufgaben:

- das automatische BusOff-Handling aktivieren / deaktivieren
 - bei Vorliegen eines BusOff die CAN-Schnittstelle wieder neu starten.
- > Wenn BusOff: CAN-Controller löscht alle Puffer (auch die Puffer der anderen Protokolle).

Wenn CAN_RECOVER nicht verwendet wird (ENABLE=FALSE):

- > Bei einem BusOff wird automatisch nach 1 s ein Recover versucht.
- > Nach 4 erfolglosen Recover-Versuchen in Folge wird die betroffene CAN-Schnittstelle deaktiviert.

Parameter der Eingänge

7514

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: KEIN automatisches Recover nach CAN-Busoff FALSE: Automatisches Recover nach CAN-Busoff
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
RECOVER_EXECUTE	BOOL	TRUE (nur 1 Zyklus lang): Neustart der CAN-Schnittstelle Busoff-Zustand beheben FALSE: Funktion wird nicht ausgeführt
InhibitTime (Parameter-Nutzung optional)	TIME := T#1s	Wartezeit zwischen BusOff und Neustart der CAN-Schnittstelle

CAN_SETDOWNLOADID

7516

= Set Download-ID

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyzz.LIB`

Symbol in CODESYS:



Beschreibung

7517

Die Download-ID ist zum Datenaustausch erforderlich bei der Verbindung zwischen dem Laufzeitsystem und der CODESYS-Entwicklungsumgebung. Die Download-ID wird voreingestellt beim Start des Geräts mit dem Wert aus der Hardware-Konfiguration gesetzt.

Mit CAN_SETDOWNLOADID kann dieser Wert im SPS-Programm (z.B. anhand bestimmter Eingänge) gesetzt werden. Die geänderte ID wird auch in die Hardware-Konfiguration geschrieben.

Parameter der Eingänge

7519

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DOWNLOAD_ID	BYTE	1...127 = Download-ID setzen 0 = Download-ID lesen

Parameter der Ausgänge

7520

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	Funktionsbaustein ist aktiv
242 F2	Fehler: Einstellung ist nicht möglich

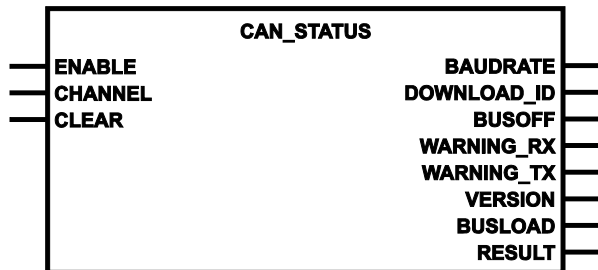
CAN_STATUS

7499

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7501

Mit CAN_STATUS können Informationen zum gewählten CAN-Bus abgefragt werden.

Ohne Hardware-Initialisierung können folgende Merker wieder auf FALSE gesetzt werden:

- BUSOFF
- WARNING_RX
- WARNING_TX

Parameter der Eingänge

7502

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
CLEAR	BOOL := FALSE	TRUE: Folgende Anzeigen zurücksetzen: • WARNING_RX • WARNING_TX • BUSOFF FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7504

Parameter	Datentyp	Beschreibung
BAUDRATE	WORD	aktuelle Baudrate des CANopen-Knotens in [kBaud]
DOWNLOAD_ID	BYTE	aktueller Download-ID
BUSOFF	BOOL	Fehler CAN-BUSOFF an der Schnittstelle
WARNING_RX	BOOL	Empfangs-Warnschwelle an der Schnittstelle überschritten
WARNING_TX	BOOL	Sende-Warnschwelle an der Schnittstelle überschritten
VERSION	DWORD	Version der ifm-CAN-Stack-Bibliothek
BUSLOAD	BYTE	mittlere Buslast in [%] zulässig: 0...100
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

Bausteine: RAW-CAN Daten empfangen

Inhalt	
CAN_RX	60
CAN_RX_ENH	61
CAN_RX_ENH_FIFO	63
CAN_RX_RANGE	65
CAN_RX_RANGE_FIFO	67

15050

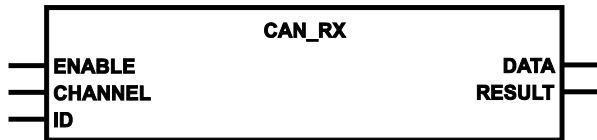
CAN_RX

7586

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyzz.LIB`

Symbol in CODESYS:



Beschreibung

7588

CAN_RX dient zum Empfang einer Nachricht.

Der FB beschränkt sich auf wenige Funktionen und hat nur geringen Speicherbedarf.

CAN_RX filtert auf den eingestellten Identifier. Wenn innerhalb eines Zyklus mehrere CAN-Nachrichten mit dem gleichen Identifier empfangen werden, steht nur die letzte / aktuellste Nachricht zur Verfügung.

Parameter der Eingänge

7589

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF

Parameter der Ausgänge

7590

Parameter	Datentyp	Beschreibung
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5 05	FB ist in der Bearbeitung – Empfangen läuft noch
9 09	CAN ist nicht aktiv
242 F2	Fehler: Einstellung ist nicht möglich

CAN_RX_ENH

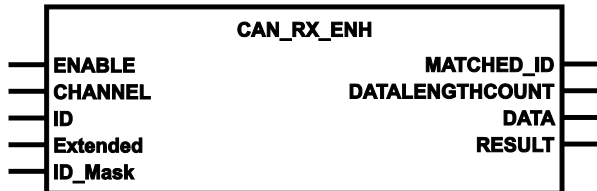
7606

= CAN RX enhanced

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyyzz.LIB

Symbol in CODESYS:



Beschreibung

7608

CAN_RX_ENH bietet (im Vergleich zu **CAN_RX** (→ Seite 60)) zusätzlich folgende Möglichkeiten:

- den Frame-Typ wählen (11 oder 29 Bit),
- eine Maske für die Auswertung des CAN-ID definieren.

Bit-Vergleich von ID und Maske:	Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.
---------------------------------	--

Mit der Maske können Sie mehrere Identifier als Filter definieren.

Beispiel:

ID =	0x100 = 0b0001 0000 0000
ID_MASK =	0x1F1 = 0b0001 1111 0001
Ergebnis	Die CAN-IDs mit folgendem Bitmuster werden ausgewertet: 0bxxx1 0000 xxx0 (x = beliebig), also für dieses Beispiel (alles in [hex]): 100, 102, 104, 106, 108, 10A, 10C, 10E, 300, 302, 304, 306, 308, 30A, 30C, 30E, 500, 502, 504, 506, 508, 50A, 50C, 50E, 700, 702, 704, 706, 708, 70A, 70C, 70E

Parameter der Eingänge

7609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)
ID_Mask (Parameter-Nutzung optional)	DWORD := 0	Filter-Maskierung zum Identifier: Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.

Parameter der Ausgänge

7613

Parameter	Datentyp	Beschreibung
MATCHED_ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALengthCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

CAN_RX_ENH_FIFO

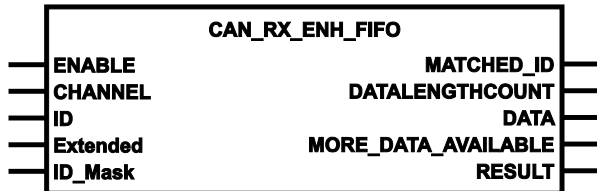
7615

= CAN RX enhanced with FIFO

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7616

CAN_RX_ENH_FIFO stellt (im Vergleich zu **CAN_RX_ENH** (→ Seite 61)) zusätzlich ein FiFo für die empfangenen Daten zur Verfügung. Somit können mehrere CAN-Telegramme innerhalb eines Zyklus empfangen werden.

! Wenn das FiFo voll ist, wird nicht überschrieben. Eingehende Nachrichten gehen dann verloren.

In diesem Fall:

- Mittels ENABLE den FB deaktivieren und wieder aktivieren.
- > Das FiFo wird gelöscht und kann von neuem befüllt werden.

Parameter der Eingänge

7609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)
ID_Mask (Parameter-Nutzung optional)	DWORD := 0	Filter-Maskierung zum Identifier: Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.

Parameter der Ausgänge

7617

Parameter	Datentyp	Beschreibung
MATCHED_ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALengthCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten im FiFo vorhanden FALSE: keine weiteren Daten im FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

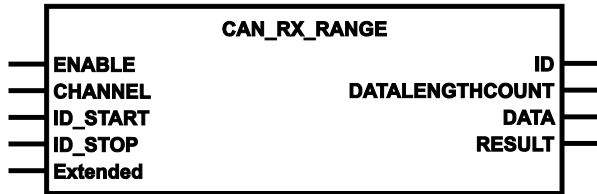
CAN_RX_RANGE

7592

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyzz.LIB`

Symbol in CODESYS:



Beschreibung

7594

CAN_RX_RANGE bietet folgende Einstellungen:

- den Nachrichten-Typ wählen (11 oder 29 Bit),
- einen Identifier-Bereich definieren.

CAN_RX filtert auf den eingestellten Identifier. Wenn innerhalb eines Zyklus mehrere CAN-Nachrichten mit dem gleichen Identifier empfangen werden, steht nur die letzte / aktuellste Nachricht zur Verfügung.

Parameter der Eingänge

7595

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID_START	DWORD	Anfangs-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame (2 ¹¹): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	End-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame (2 ¹¹): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)

Parameter der Ausgänge

7598

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALengthCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

CAN_RX_RANGE_FIFO

7601

= CAN RX Range with FIFO

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7603

CAN_RX_RANGE_FIFO arbeitet grundsätzlich wie **CAN_RX_RANGE** (→ Seite 65).

Zusätzlich stellt CAN_RX_RANGE_FIFO ein FiFo für die empfangenen Daten zur Verfügung. Somit können mehrere CAN-Telegramme innerhalb eines Zyklus empfangen werden.

⚠ Wenn das FiFo voll ist, wird nicht überschrieben. Eingehende Nachrichten gehen dann verloren.

In diesem Fall:

- ▶ Mit ENABLE die Funktion deaktivieren und wieder aktivieren.
- > Das FiFo wird gelöscht und kann von neuem befüllt werden.

Parameter der Eingänge

7595

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID_START	DWORD	Anfangs-Nummer des Datenobjekt-Identifizier-Bereichs: Normal Frame (2 ¹¹): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	End-Nummer des Datenobjekt-Identifizier-Bereichs: Normal Frame (2 ¹¹): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)

Parameter der Ausgänge

7604

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALengthCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten im FiFo vorhanden FALSE: keine weiteren Daten im FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

Bausteine: RAW-CAN Daten senden

Inhalt	
CAN_TX	70
CAN_TX_ENH.....	71
CAN_TX_ENH_CYCLIC	73
	15055

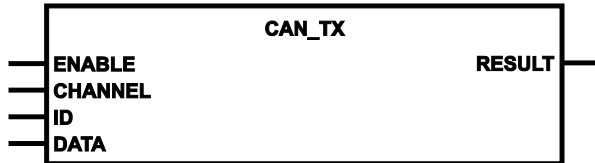
CAN_TX

7522

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyzz.LIB`

Symbol in CODESYS:



Beschreibung

7523

CAN_TX sendet eine Standard-Nachricht pro Zyklus.

Der FB beschränkt sich auf wenige Funktionen und hat nur geringen Speicherbedarf.

- > Bei mehrmaligem Aufruf derselben Instanz dieses FBs während eines Zyklus werden die Daten ebenfalls mehrmals versendet.

Bei den einfachen Funktionen CAN_TX und CAN_RX wird anhand des ID ermittelt, ob ein Standard- oder ein Extended-Frame versendet werden soll. Bei den Enhanced-Versionen wird dies über den Eingang EXTENDED festgelegt. Mit den einfachen Funktionen kann man folglich keine Extended-Frames im ID-Bereich 0...2047 versenden.

Parameter der Eingänge

7524

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7527

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
242 F2	Fehler: Einstellung ist nicht möglich
250 FA	Fehler: FiFo ist voll – Daten wurden verloren

CAN_TX_ENH

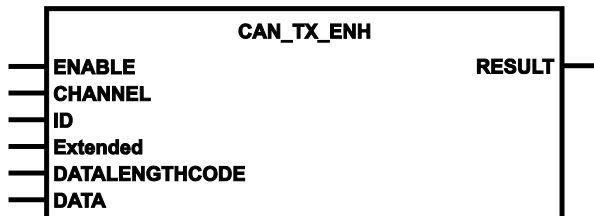
7558

= CAN TX enhanced

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7559

Zusätzliche Einstellmöglichkeiten bietet CAN_TX_ENH (für: enhanced). Hier können alle CAN-spezifischen Eigenschaften individuell eingestellt werden, z.B.:

- Handelt es sich um einen 11- oder 29-Bit-Identifizier?
- Die zusätzlichen Eingänge können voreingestellt werden, so dass **CAN_TX** (→ Seite 70) nicht erforderlich ist.
- > Bei mehrmaligem Aufruf derselben Instanz dieses FBs während eines Zyklus werden die Daten ebenfalls mehrmals versendet.

Parameter der Eingänge

7634

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇒ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)
DATALENGTHCODE	BYTE	= Data Length Code Anzahl der zu sendenden Daten-Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7527

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren



CAN_TX_ENH_CYCLIC

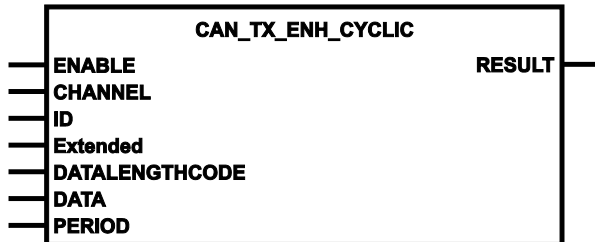
7568

= CAN TX enhanced Cyclic

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7569

CAN_TX_ENH_CYCLIC dient dem zyklischen Versand von CAN-Nachrichten.

Der FB entspricht ansonsten **CAN_TX_ENH** (→ Seite 71).

► Mit dem Parameter PERIOD die Periodendauer einstellen.

! Eine zu kurze Periodendauer kann zu einer hohen Buslast führen, was das Verhalten des Gesamtsystems beeinträchtigen könnte.

Parameter der Eingänge

7582

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)
DataLengthCode (Parameter-Nutzung optional)	BYTE := 8	Länge der zu sendenden Daten (0...8 Bytes)
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)
PERIOD	TIME	Periodendauer

Parameter der Ausgänge

7510

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich



Bausteine: RAW-CAN Remote

Inhalt

CAN_REMOTE_REQUEST	76
CAN_REMOTE_RESPONSE	77

15057

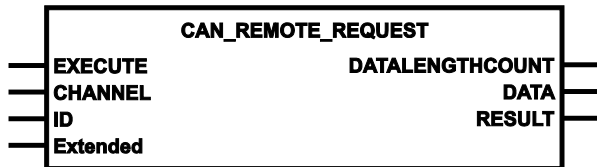
CAN_REMOTE_REQUEST

7625

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyzz.LIB`

Symbol in CODESYS:



Beschreibung

7627

Zur Anfrage einer Remote-Nachricht wird mit CAN_REMOTE_REQUEST eine entsprechende Anforderung versandt und die Antwort des anderen Gerätes als Ergebnis zurückgeliefert.

Parameter der Eingänge

7628

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)

Parameter der Ausgänge

7629

Parameter	Datentyp	Beschreibung
DATALENGTHCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5 05	FB ist in der Bearbeitung – Empfangen läuft noch
9 09	CAN ist nicht aktiv
242 F2	Fehler: Einstellung ist nicht möglich

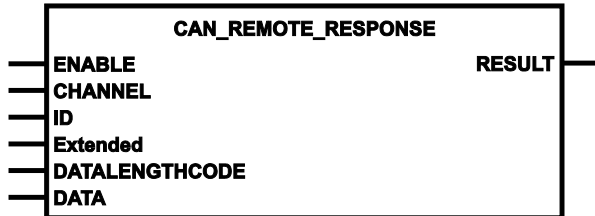
CAN_REMOTE_RESPONSE

7631

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_RawCAN_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7633

CAN_REMOTE_RESPONSE stellt dem CAN-Controller im Gerät Daten zur Verfügung, die automatisch auf die Anfrage einer Remote-Nachricht gesendet werden.

Dieser FB ist stark geräte-abhängig. Es kann nur eine begrenzte Anzahl von Remote-Nachrichten eingerichtet werden:

BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n	max. 40 Remote-Nachrichten
PDM360 NG: CR108n	max. 100 Remote-Nachrichten

Parameter der Eingänge

7634

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇒ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2 ¹¹ IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2 ²⁹ IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2 ²⁹ -1) FALSE: Normal Frame (ID = 0...2 ¹¹ -1)
DATALENGTHCODE	BYTE	= Data Length Code Anzahl der zu sendenden Daten-Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7636

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
6	06	FB ist in der Bearbeitung – Remote für ID nicht aktiv
7	07	FB ist in der Bearbeitung – Remote für ID aktiv



5.2.3 Bausteine: CANopen

Inhalt	
Bausteine: CANopen Status.....	79
Bausteine: CANopen Netzwerkmanagement.....	88
Bausteine: CANopen Objektverzeichnis	92
Bausteine: CANopen SDOs	97
Bausteine: CANopen SYNC.....	110
Bausteine: CANopen Guarding	114
Bausteine: CANopen Emergency.....	118

15059

Für CANopen stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

Bausteine: CANopen Status

Inhalt	
CANOPEN_ENABLE.....	80
CANOPEN_GETBUFFERFLAGS	82
CANOPEN_GETSTATE.....	84
CANOPEN_SETSTATE	86

15061

CANOPEN_ENABLE

7785

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7787

CANOPEN_ENABLE erlaubt es, den CANopen-Master ein- und auszuschalten.

- **!** Im Anwendungsprogramm für jede CAN-Schnittstelle immer eine eigene Instanz des FBs **CANOPEN_ENABLE** (→ Seite [80](#)) anlegen!

! Zur Vermeidung von Guarding- oder Heartbeat-Fehlern müssen zuvor die Knoten durch eine geeignete Sequenz "heruntergefahren" werden.

Wird der Master nach einem Stopp wieder gestartet, so müssen auch alle angeschlossenen Knoten wieder initialisiert werden.

Ohne CANOPEN_ENABLE wird der CANopen-Master automatisch gestartet, sofern dies in der Konfiguration gewählt wurde.

Die konfigurierte Baudrate wird nur übernommen, wenn zuvor nicht **CAN_ENABLE** (→ Seite [54](#)) aufgerufen wurde.

Parameter der Eingänge

7788

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := TRUE	TRUE: • CANopen für den gewählten Kanal freigeben • CANopen-Manager oder CANopen-Device starten entsprechend den Konfigurations-Einstellungen FALSE: • CANopen für den gewählten Kanal sperren • CANopen-Manager oder CANopen-Device beenden
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Baudrate (Parameter-Nutzung optional)	WORD := 0	Baudrate [kBit/s] zulässig = 20, 50, 100, 125, 250, 500, 800, 1 000 0 = Einstellung aus der Steuerungskonfiguration verwenden

Parameter der Ausgänge

7789

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
14	0E	FB ist aktiv CANopen-Manager konfiguriert Devices und sendet SDOs
15	0F	FB ist aktiv CANopen-Manager ist gestartet
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Manager konnte nicht gestartet werden
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_GETBUFFERFLAGS

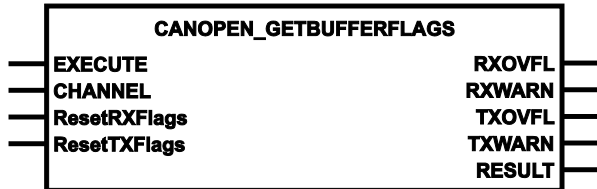
7890

= Get Buffer-Flags

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7892

CANOPEN_GETBUFFERFLAGS liefert Informationen zu den Buffer-Flags.

Über optionale Eingänge können die Flags zurückgesetzt werden.

Der Funktionsbaustein liefert den Zustand der Overflow-Flags zurück.

Parameter der Eingänge

7893

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ResetRXFlags (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Flag-Status am Ausgang ausgeben und anschließend zurücksetzen FALSE: Funktion wird nicht ausgeführt
ResetTXFlags (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Flag-Status am Ausgang ausgeben und anschließend zurücksetzen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7894

Parameter	Datentyp	Beschreibung
RXOVFL	BOOL	Zustand des RX-Overflow-Flags TRUE: Überlauf im Empfangspuffer FALSE: Kein Überlauf im Empfangspuffer
RXWARN	BOOL	Zustand des RX-Overflow-Warning-Flags TRUE: Füllstand im Empfangspuffer ist kritisch FALSE: Füllstand im Empfangspuffer ist unkritisch
TXOVFL	BOOL	Zustand des TX-Overflow-Flags TRUE: Überlauf im Sendepuffer FALSE: Kein Überlauf im Sendepuffer
TXWARN	BOOL	Zustand des TX-Overflow-Warning-Flags TRUE: Füllstand im Sendepuffer ist kritisch FALSE: Füllstand im Sendepuffer ist unkritisch
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_GETSTATE

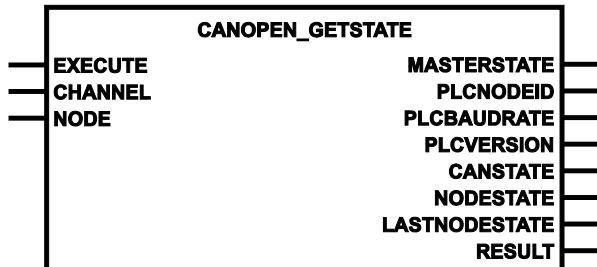
7865

= Get State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7867

Mit CANOPEN_GETSTATE können Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz abgefragt werden.

Parameter der Eingänge

7868

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	Node-ID = ID des Knotens (0...127) Gerät als CANopen-Master: Wert = 0: Nur die Statusinformationen des Geräts selbst werden an den Ausgängen zurückgeliefert. Die Ausgänge mit Informationen zu den Knoten sind ungültig. Wert nicht 0: Node-ID eines Knotens im Netzwerk. Für diesen sowie für das Gerät werden an den Ausgängen die Zustände zurückgeliefert. Gerät als CANopen-Slave: Wert = 0 (voreingestellt): Die Statusinformationen des Slave werden an den Ausgängen zurückgeliefert. Wert nicht 0: keine Aktion

Parameter der Ausgänge

7869

Parameter	Datentyp	Beschreibung
MASTERSTATE	BYTE	Master State = interner Zustand des Masters: 0 = 0x00 = Master läuft hoch 4 = 0x04 = Konfiguration der Knoten läuft 5 = 0x05 = normaler Betriebszustand des Masters 255 = 0xFF = PLC läuft als Slave
PLCNODEID	BYTE	PLC Node-ID = Node-ID der SPS, auf der das Programm ausgeführt wird Wert = 0...127 = 0x00...0x7F
PLCBAUDRATE	DWORD	Baudrate der SPS
PLCVERSION	DWORD	Version der SPS
CANSTATE	BYTE	Status des CANopen-Netzwerks Gerät als Master betrieben: Node-ID = 0 (Gerät selbst): 0 = 0x00 = OK 128 = 0x80 = BUSOFF Node-ID ≠ 0 (Knoten): 0 = 0x00 = OK 1 = 0x01 = Guard- oder Heartbeat-Fehler an Knoten 128 = 0x80 = BUSOFF Gerät als Slave betrieben: 0 = 0x00 = OK 1 = 0x01 = Guard- oder Heartbeat-Fehler 128 = 0x80 = BUSOFF
NODESTATE	BYTE	Node State = interner Knotenstatus eines Slaves aus Sicht des Masters. Der Knoten wird durch den Eingang NODEID bezeichnet. -1 = 0xFF = Rücksetzen nach ResetNode 1 = 0x01 = Warten auf BOOTUP 2 = 0x02 = Nach Empfang der BOOTUP-Nachricht 3 = 0x03 = noch nicht konfiguriert: STOPPED 4 = 0x04 = nach der Konfiguration mit SDOs: PRE-OPERATIONAL 5 = 0x05 = nach dem Starten des Knotens: OPERATIONAL 97 = 0x61 = optionaler Knoten 98 = 0x62 = anderer Gerätetyp als in 0x1000 konfiguriert 99 = 0x63 = Node-Guarding
LASTNODESTATE	BYTE	Last Node State = letzter Status des Knotens Knotenstatus nach CANopen (mit diesen Werten wird der Status auch in den entsprechenden Nachrichten vom Knoten her codiert). 0 0x00 BOOTUP 4 0x04 STOPPED 5 0x05 OPERATIONAL 127 0x7F PRE-OPERATIONAL
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_SETSTATE

7858

= Set State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7860

Mit CANOPEN_SETSTATE können Parameter des Masters, eines Slave-Devices oder eines Knotens im Netz gesetzt werden.

Die Behandlung des NMT-Zustands von Master, Knoten oder Device erfolgt im CAN-Stack oder über die Kommandos des FB **CANOPEN_NMTSERVICES** (→ Seite 90). Dabei werden gleichzeitig auch Zulässigkeitsprüfungen durchgeführt. Aus Konsistenzgründen sind deshalb hier für diesen Zweck keine Eingänge vorgesehen.

Parameter der Eingänge

7861

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	Node-ID = ID des Knotens (0...127) Gerät als CANopen-Master: Wert = 0: Die Änderungen beziehen sich nur auf das Gerät selbst. Wert nicht 0: Node-ID eines Knotens am Netzwerk, dessen Parameter verändert werden sollen. Nur für diesen Knoten (nicht für das Gerät) werden die angelegten Einstellungen übernommen. Gerät als CANopen-Slave: Im Slave-Mode kann über diesen Eingang der Node-ID des Slave gesetzt werden. Wert = 0: keine Aktion Wert nicht 0: Diesen Wert übernimmt der FB als neuen Node-ID des Geräts.
GlobalStart (Parameter-Nutzung optional)	BOOL := TRUE	Voraussetzung: FB muss unmittelbar nach dem Start des IEC-Programms aufgerufen werden. Diese Einstellung überschreibt die Einstellung aus der Konfiguration. TRUE: alle Teilnehmer gleichzeitig starten FALSE: alle Teilnehmer nacheinander starten
CfgTimeout (Parameter-Nutzung optional)	TIME := T#0ms	Konfigurations-Timeout für einen Knoten setzen: Wert = 0: keine Aktion – Konfigurationsdaten behalten Wert nicht 0: Daten aus der Konfiguration mit dem neuen Wert überschreiben
InitNoSdos (Parameter-Nutzung optional)	BOOL := FALSE	An den in NODE angegebenen Knoten beim Initialisieren... TRUE: keine Konfigurationsdaten senden FALSE: die konfigurierten SDOs senden

Parameter der Ausgänge

7862

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8 08	FB ist aktiv – noch nicht bearbeitet
242 F2	Fehler: Einstellung ist nicht möglich

Bausteine: CANopen Netzwerkmanagement

Inhalt

CANOPEN_GETNMTSTATESLAVE	89
CANOPEN_NMTSERVICES	90

15063

CANOPEN_GETNMTSTATESLAVE

7851

= Get Network Management State Slave

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7853

► Baustein nur verwenden, wenn das Gerät als CANopen-Slave betrieben wird!

An die Anwendung wird mit CANOPEN_GETNMTSTATESLAVE nur noch der Betriebszustand nach CANopen gemeldet sowie eine Fehlermeldung, falls ein ungültiger Zustandsübergang angefordert wurde.

Parameter der Eingänge

7854

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

Parameter der Ausgänge

7855

Parameter	Datentyp	Beschreibung
NMTSTATE	BYTE	Netzwerk-Betriebszustand des Knotens 0 = INIT 1 = OPERATIONAL 2 = PRE-OPERATIONAL 3 = STOPPED
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_NMTSERVICES

7843

= Network Management Services

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7844

CANOPEN_NMTSERVICES löst abhängig von seinen NMT-Kommando-Eingängen ein NMT-Kommando oder die Initialisierung eines Knotens aus.

 NMT = **Network-Management**

Der Funktionsbaustein aktualisiert den internen Knotenstatus. Sollte ein Zustandsübergang nach CANopen (→ **NMT-Status**) nicht erlaubt sein, so wird das Kommando nicht ausgeführt.

Ein CANopen-Device kann mit Hilfe des FB seinen CANopen-Status selbständig ändern:
Preoperational ↔ Operational

Parameter der Eingänge

7847

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x00...0x7F NODE = 0: Kommando gilt für alle Knoten im Netzwerk NODE = Node-ID des Geräts: Kommando gilt für das Gerät selbst
NMTSERVICE	BYTE	Netzwerk-Kommando 0 = Init Node (außer Master) 1 = Enter PRE-OPERATIONAL 2 = Start Node 3 = Reset Node 4 = Reset Kommunikation 5 = Stop Node
Timeout (Parameter-Nutzung optional)	TIME := T#0ms	Wartezeit des FB auf die Initialisierung Nach Zeitablauf bricht der FB das Warten ab. 0 = Wert aus der Konfiguration verwenden

Parameter der Ausgänge

7848

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
35	23	mindestens 1 SDO der Konfiguration war nicht erfolgreich
36	24	Knoten war bereits initialisiert
37	25	zur Initialisierung war Knoten nicht im Modus PRE-OPERATIONAL
043	2B	Master / Slave ist nicht initialisiert
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich

Bausteine: CANopen Objektverzeichnis

Inhalt	
CANOPEN_GETODCHANGEDFLAG	93
CANOPEN_READOBJECTDICT	94
CANOPEN_WRITEOBJECTDICT	95
	15065

CANOPEN_GETODCHANGEDFLAG

7927

= Get Object Directory Changed Flag

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7928

CANOPEN_GETODCHANGEDFLAG meldet wenn bei einem bestimmten Objektverzeichnis-Eintrag der Wert geändert wurde.

Parameter der Eingänge

7930

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis

Parameter der Ausgänge

7931

Parameter	Datentyp	Beschreibung
DATA	DWORD	Parameter-Wert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	FB ist aktiv – noch nicht bearbeitet
242 F2	Fehler: Einstellung ist nicht möglich

CANOPEN_READOBJECTDICT

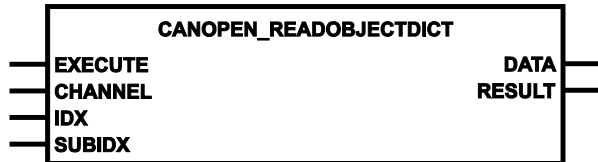
7933

= Read Object Directory

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyyzz.LIB

Symbol in CODESYS:



Beschreibung

7935

CANOPEN_READOBJECTDICT liest bis zu 4 Bytes Konfigurationsdaten aus dem Objektverzeichnis des Geräts zur Verwendung im Anwendungsprogramm.

Parameter der Eingänge

7936

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis

Parameter der Ausgänge

7937

Parameter	Datentyp	Beschreibung
DATA	DWORD	Parameter-Wert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	Funktionsbaustein ist noch nicht ausgeführt
40 28	Objektverzeichnis-Eintrag ist ungültig
242 F2	Fehler: Einstellung ist nicht möglich

CANOPEN_WRITEOBJECTDICT

7940

= Write Object Directory

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7942

CANOPEN_WRITEOBJECTDICT schreibt Konfigurationsdaten in das Objektverzeichnis der Steuerung.

ACHTUNG

Wichtige Systemeinstellungen können hierbei verfälscht werden, z.B.:

- Guarding-Zeiten
- Heartbeat-Zeiten

► Eingabe-Parameter sorgfältig prüfen!

Parameter der Eingänge

7943

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
DATA	DWORD	Parameter-Wert

Parameter der Ausgänge

7945

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
40	28	Objektverzeichnis-Eintrag ist ungültig
242	F2	Fehler: Einstellung ist nicht möglich

Bausteine: CANopen SDOs

Inhalt	
CANOPEN_SDOREAD	98
CANOPEN_SDOREADBLOCK.....	100
CANOPEN_SDOREADMULTI	102
CANOPEN_SDOWRITE	104
CANOPEN_SDOWRITEBLOCK.....	106
CANOPEN_SDOWRITEMULTI	108

2071

Hier finden Sie **ifm**-Bausteine für den Umgang von CANopen mit Service Data Objects (SDOs).

CANOPEN_SDOREAD

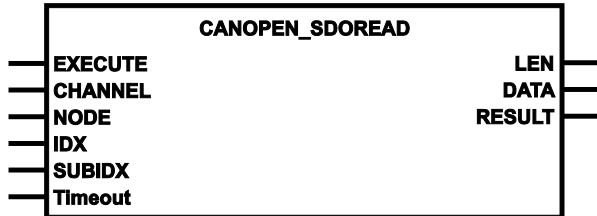
7791

= SDO Read

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7793

CANOPEN_SDOREAD ist ein einfacher Funktionsbaustein zur Bearbeitung von "Expedited SDOs", also SDOs mit maximal 4 Nutzdaten-Bytes. Diese Art bildet in der Regel einen Großteil der SDO-Kommunikation ab.

 Expedited SDO = beschleunigtes Nachrichten-Objekt mit Servicedaten

Wegen der auf max. 4 Nutzdaten-Bytes begrenzten Datenmenge lässt sich erheblich Speicherplatz sparen, da dieser FB nur 4 Bytes als Pufferspeicher vorhalten muss und selbst kein großes Daten-Array anlegt.

Parameter der Eingänge

7794

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

7795

Parameter	Datentyp	Beschreibung
LEN	BYTE	Anzahl der empfangenen Bytes (1...4)
DATA	DWORD	der empfangene Datenwert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich
255	FF	Pufferüberlauf – zu viele Daten-Bytes wurden empfangen

CANOPEN_SDOREADBLOCK

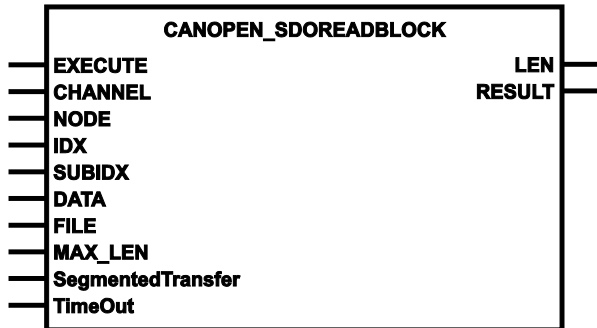
14942

= SDO Read Block

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

14943

CANOPEN_SDOREADBLOCK liest den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer.

- > Falls der Knoten keinen Blocktransfer unterstützt, schaltet der FB automatisch um auf den "Segmented Transfer". Per Eingang kann aber auch direkt auf den "Segmented Transfer" umgeschaltet werden.
- > Die COB-ID für den SDO wird aus der übergebenen Node-ID berechnet.

Multiframe-SDOs sind in der Länge grundsätzlich nicht begrenzt.

Für Systeme ohne File-System (z.B. BasicController CR04nn) gilt:

- ▶ Dem FB eine Adresse übergeben, auf die per Zeiger schreibend zugegriffen wird. Der durch Startadresse DATA und Datenanzahl MAX_LEN definierte Speicherbereich muss verfügbar sein!
- > Ist die Datenmenge größer als angegeben, wird der Transfer abgebrochen und per RESULT signalisiert.

Für Systeme mit File-System (z.B. PDM360NG CR108n) gilt:

- ▶ Dem FB den Pfad und Namen einer Datei übergeben, in welcher die Daten im Binärformat gespeichert werden sollen.
- > Zum Status der SDO-Übertragung informiert der Ausgang RESULT.

Parameter der Eingänge

14945

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens zulässig = 0x01...0x7F = 1...127  Die COB-ID des SDOs errechnet sich aus Knoten-ID + 0x600
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
DATA	DWORD	Adresse des Datenbereichs zum Speichern der empfangenen Daten  Eingang ist ohne Funktion bei Gerät mit Dateisystem (Linux).
FILE	STRING(80)	Pfad und Dateiname zum Speichern der empfangenen Daten im Binärformat  Eingang ist ohne Funktion bei Gerät ohne Dateisystem (BasicSystem).
MAX_LEN	DWORD	Maximal erlaubte Anzahl der Bytes, die empfangen werden dürfen
SegmentedTransfer (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Segmented SDO-Transfer FALSE: SDO-Blocktransfer
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

14951

Parameter	Datentyp	Beschreibung
LEN	DWORD	Anzahl der empfangenen Daten-Bytes
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
16 10	Übertragung läuft als segmentierter Download
17 11	Übertragung läuft als Block-Download
32 20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33 21	TIMEOUT ist abgelaufen
64 40	Fehler: Schreibzeiger ist außerhalb des zulässigen Datenbereichs
65 41	Fehler: Datei konnte nicht geöffnet werden
66 42	Fehler bei Schreiben auf Datei
242 F2	Fehler: Einstellung ist nicht möglich

CANOPEN_SDOREADMULTI

7806

= SDO Read Multi

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7808

CANOPEN_SDOREADMULTI liest den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz. Die COB-ID für das SDO wird nach CANopen-Konvention aus der übergebenen Node-ID berechnet.

Parameter der Eingänge

7809

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens zulässig = 0x01...0x7F = 1...127  Die COB-ID des SDOs errechnet sich aus Knoten-ID + 0x600
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

7810

Parameter	Datentyp	Beschreibung
LEN	DWORD	Anzahl der empfangenen Bytes zulässige Werte = 0x0000 0001...0x0000 0800 = 1...2 048
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	Pufferspeicher für Nutzdaten der SDO-Datenübertragung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
34	22	Toggle-Bit-Fehler
242	F2	Fehler: Einstellung ist nicht möglich
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

CANOPEN_SDOWNWRITE

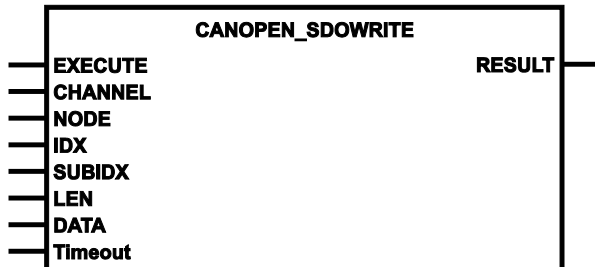
7825

= SDO Write

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzzz.LIB

Symbol in CODESYS:



Beschreibung

7826

CANOPEN_SDOWNWRITE ist ein einfacher Funktionsbaustein zur Bearbeitung von "Expedited SDOs", also SDOs mit maximal 4 Nutzdaten-Bytes. Diese Art bildet in der Regel einen Großteil der SDO-Kommunikation ab.

 Expedited SDO = beschleunigtes Nachrichten-Objekt mit Servicedaten

Wegen der auf max. 4 Nutzdaten-Bytes begrenzten Datenmenge lässt sich erheblich Speicherplatz sparen, da dieser FB nur 4 Bytes als Pufferspeicher vorhalten muss und selbst kein großes Daten-Array anlegt.

Parameter der Eingänge

7828

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	BYTE	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 1...4 = 0x01...0x04
DATA	ARRAY [0..3] OF BYTE	Datenbereich (1...4 Bytes)
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

7829

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_SDOWRITEBLOCK

14961

= SDO Write Block

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

14963

CANOPEN_SDOWRITEBLOCK schreibt in den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer.

Per FB-Eingang kann bei Bedarf auf den Segmented Transfer umgeschaltet werden.

- > Die COB-ID für den SDO wird aus der übergebenen Node-ID berechnet.
- > Zum Status der SDO-Übertragung informiert der Ausgang RESULT.

Multiframe-SDOs sind in der Länge grundsätzlich nicht begrenzt.

Für Systeme ohne File-System (z.B. BasicController CR04nn) gilt:

- Dem FB eine Adresse übergeben, auf die per Zeiger lesend zugegriffen wird.

Für Systeme mit File-System (z.B. PDM360NG CR108n) gilt:

- Dem FB den Pfad und Namen einer Datei übergeben, aus welcher die Daten im Binärformat gelesen werden sollen.

Parameter der Eingänge

14964

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE $\Rightarrow$ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens zulässig = 0x01...0x7F = 1...127  Die COB-ID des SDOs errechnet sich aus Knoten-ID + 0x600
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	DWORD	Anzahl der in DATA zu übergebenden Daten-Bytes zulässig = 1...2 048 = 0x0000 0001...0x0000 0800
DATA	DWORD	Adresse des Datenbereichs zum Lesen der zu sendenden Daten  Eingang ist ohne Funktion bei Gerät mit Dateisystem (Linux).
FILE	STRING(80)	Pfad und Dateiname zum Lesen der zu sendenden Daten im Binärformat  Eingang ist ohne Funktion bei Gerät ohne Dateisystem (BasicSystem).
SegmentedTransfer (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Segmented SDO-Transfer FALSE: SDO-Blocktransfer
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

14968

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen $\rightarrow$ folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
16	10	Übertragung läuft als segmentierter Download
17	11	Übertragung läuft als Block-Download
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
65	41	Fehler: Datei konnte nicht geöffnet werden
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_SDOWNITEMULTI

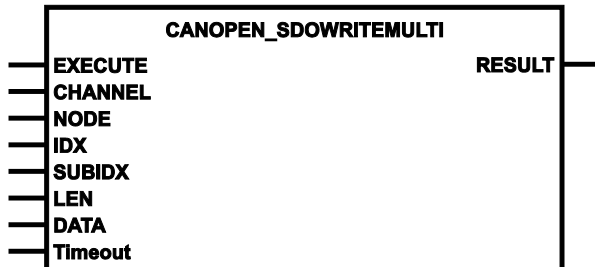
7832

= SDO Write Multi

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyyzz.LIB

Symbol in CODESYS:



Beschreibung

7834

CANOPEN_SDOWNITEMULTI schreibt den angegebenen Eintrag im Objektverzeichnis eines Knotens im Netz. Die COB-ID für den SDO wird nach CANopen-Konvention aus der übergebenen Node-ID berechnet.

Parameter der Eingänge

7835

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	DWORD	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 0000 0001...0000 0800 ₁₆ = 1...2 048 ₁₀
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	Pufferspeicher für Nutzdaten der SDO-Datenübertragung
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

7836

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
34	22	Toggle-Bit-Fehler
242	F2	Fehler: Einstellung ist nicht möglich

Bausteine: CANopen SYNC

Inhalt

CANOPEN_GETSYNCSTATE.....	111
CANOPEN_SETSYNCSTATE	113

15069

CANOPEN_GETSYNCSTATE

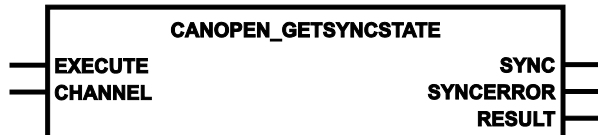
7871

= Get SYNC State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7872

CANOPEN_GETSYNCSTATE liest...

- die Einstellung der SYNC-Funktionalität (aktiv / deaktiv),
- den Fehlerzustand der SYNC-Funktionalität (SyncError).

Wenn die PLC als CANopen-Slave läuft, wird über diesen FB signalisiert, ob SYNC-Signale ausbleiben oder ob sie regelmäßig kommen.

Die Bearbeitung von synchronen PDOs usw. läuft im CAN-Stack. CANOPEN_GETSYNCSTATE liefert jedoch den Fehlerzustand, so dass das Anwendungsprogramm darauf entsprechend reagieren kann.

Parameter der Eingänge

7874

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

Parameter der Ausgänge

7875

Parameter	Datentyp	Beschreibung
SYNC	BOOL	Status der SYNC-Funktionalität TRUE: SYNC ist aktiviert. Im Master-Betrieb werden SYNC-Telegramme erzeugt entsprechend Einstellungen in Konfiguration sowie synchrone PDOs gesendet und empfangen. Im Slave-Betrieb werden SYNC-Telegramme empfangen und entsprechend bearbeitet. FALSE: SYNC ist nicht aktiv
SYNCERROR	BYTE	(Sync-Error) SYNC-Fehlermeldung 0 = kein Fehler >0 = SYNC-Error (Slave-Betrieb)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_SETSYNCSTATE

7883

= Set SYNC State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7884

Mit CANOPEN_SETSYNCSTATE wird die SYNC-Funktionalität ein- und ausgeschaltet.

Parameter der Eingänge

7886

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
SYNC	BOOL	Status der SYNC-Funktionalität TRUE: SYNC ist aktiviert: Im Master-Betrieb werden SYNC-Telegramme erzeugt entsprechend Einstellungen in Konfiguration sowie synchrone PDOs gesendet und empfangen. Im Slave-Betrieb werden SYNC-Telegramme empfangen und entsprechend bearbeitet. FALSE: SYNC ist nicht aktiv

Parameter der Ausgänge

7887

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	Funktionsbaustein ist noch nicht ausgeführt
38 26	SYNC konnte nicht aktiviert werden
242 F2	Fehler: Einstellung ist nicht möglich

Bausteine: CANopen Guarding

Inhalt

CANOPEN_GETGUARDHBERRLIST	115
CANOPEN_GETGUARDHBSTATSLV	116

15071

CANOPEN_GETGUARDHBERRLIST

7896

= Get Guard and Heartbeat Error-List

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7898

CANOPEN_GETGUARDHBERRLIST listet in einem Array alle Knoten auf, für die der Master einen Fehler erkannt hat:

- Guarding-Fehler
- Heartbeat-Fehler

Parameter der Eingänge

7899

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ResetList (Parameter-Nutzung optional)	BOOL := FALSE	Fehlerliste zurücksetzen TRUE: Die Fehlerliste sowie die Anzahl der fehlerhaften Knoten am Ausgang ausgeben und anschließend zurücksetzen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7900

Parameter	Datentyp	Beschreibung
N_NODES	WORD	Anzahl der Knoten mit Heartbeat- oder Guarding-Fehlern 0 = kein Knoten hat einen Guarding- oder Heartbeat-Fehler
NODEID	ARRAY [0..MAXGUARDERROR] OF BYTE	Liste der Knoten-IDs mit Heartbeat- oder Guarding-Fehler. Der jüngste Eintrag steht im Index 0. MAXGUARDERROR ist abhängig vom Gerät → Kapitel Leistungsgrenzen der Geräte (CANopen) (→ Seite 34)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	FB ist aktiv – noch nicht bearbeitet
242 F2	Fehler: Einstellung ist nicht möglich

CANOPEN_GETGUARDHBSTATSLV

7902

= Get Guard and Heartbeat State Slave

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7904

CANOPEN_GETGUARDANDHBSTATESLAVE meldet der Steuerung im Slave-Betrieb folgende Zustände:

- Node-Guarding-Überwachung
- Heartbeat-Überwachung

Dabei kann die Steuerung Heartbeat-Producer und Heartbeat-Consumer sein.

Parameter der Eingänge

7905

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Reset (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Die aktuellen Zustände an den Ausgängen ausgeben und anschließend auf "Kein Fehler" zurücksetzen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7906

Parameter	Datentyp	Beschreibung
GUARDSTATE	BYTE	Status des Nodeguarding: 0 = 0x00 = kein Fehler (oder: inaktiv) 1 = 0x01 = Zeitüberschreitung (Konfiguration) 127 = 0x7F = noch keine Guarding-Nachricht empfangen
PROD_HBSTATE	BYTE	Steuerung als Heartbeat-Producer: 0 = 0x00 = inaktiv 1 = 0x01 = aktiv
CONS_HBSTATE	BYTE	Steuerung als Heartbeat-Consumer: 0 = 0x00 = kein Fehler 1 = 0x01 = Zeitüberschreitung (Konfiguration) 127 = 0x7F = noch keine Heartbeat-Nachricht empfangen
CONS_HBCOBID	WORD	COB-ID der Heartbeat-Nachricht, auf die der Consumer-Heartbeat der Steuerung hört (Konfiguration)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

Bausteine: CANopen Emergency

Inhalt	
CANOPEN_GETEMCYMESSAGES.....	119
CANOPEN_GETERRORREGISTER.....	121
CANOPEN_SENDEMCMYMESSAGE	122

15073

CANOPEN_GETEMCYMESSAGES

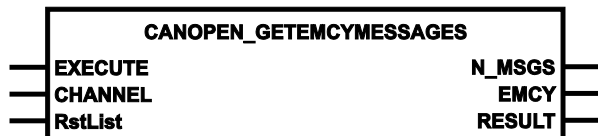
7921

= Get Emergency-Messages

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7923

CANOPEN_GETEMCYMESSAGES gibt alle Emergency-Nachrichten zurück, die die Steuerung seit dem letzten Löschen der Nachrichten von anderen Knoten am Netz empfangen hat.

Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden. Es werden maximal MAXEMCYMSGs Nachrichten gespeichert. Jede Nachricht enthält dabei als Info, von welchem Knoten sie gesendet wurde. Dabei steht die jüngste Nachricht im Index 0.

Parameter der Eingänge

7924

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
RstList (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Liste mit aufgelaufenen CAN-Nachrichten am Ausgang ausgeben und anschließend löschen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7925

Parameter	Datentyp	Beschreibung								
N_MSGS	DWORD	Anzahl der aufgelaufenen Nachrichten								
EMCY	ARRAY [0..MAXEMCYMSGS] OF T_EMCY	Emergency-Nachrichten Der jüngste Eintrag steht im Index 0. Struktur von T_EMCY: <table><tr><td>.NODEID</td><td>ID des Knotens von dem die Nachricht kam</td></tr><tr><td>.EEC</td><td>Emergency Error Code</td></tr><tr><td>.ER</td><td>Error Register</td></tr><tr><td>.MSEF</td><td>Manufacturer Specific Error Code</td></tr></table> MAXEMCYMSG = 10	.NODEID	ID des Knotens von dem die Nachricht kam	.EEC	Emergency Error Code	.ER	Error Register	.MSEF	Manufacturer Specific Error Code
.NODEID	ID des Knotens von dem die Nachricht kam									
.EEC	Emergency Error Code									
.ER	Error Register									
.MSEF	Manufacturer Specific Error Code									
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)								

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

CANOPEN_GETERRORREGISTER

7915

= Get Error-Register

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7917

CANOPEN_GETERRORREGISTER liest die Fehler-Register 0x1001 und 0x1003 der Steuerung aus.

Parameter der Eingänge

7918

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Reset_1001 (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Fehler-Register 0x1001 zurücksetzen FALSE: Funktion wird nicht ausgeführt
Reset_1003 (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Fehler-Register 0x1003 zurücksetzen Anzahl der Einträge auf 0 setzen FALSE: Funktion wird nicht ausgeführt Die Einträge bleiben unverändert.

Parameter der Ausgänge

7919

Parameter	Datentyp	Beschreibung
ER	BYTE	Inhalt des Fehler-Registers 0x1001
ERROR_FIELD	ARRAY [0..MAXERR] OF DWORD	Inhalt des Error-Registers 0x1003 Index 0 = Anzahl der gespeicherten Fehler Index 1...MAXERR = gespeicherte Fehler Der jüngste Fehler steht im Index 1 voreingestellt: MAXERR = 5
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	FB ist aktiv – noch nicht bearbeitet
242 F2	Fehler: Einstellung ist nicht möglich

CANOPEN_SENDEMCMYMESSAGE

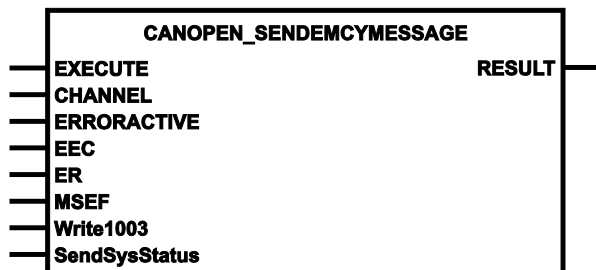
7908

= Send Emergency-Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CANopen_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7910

CANOPEN_SENDEMCMYMESSAGE versendet eine EMCY-Nachricht. Die Nachricht wird aus den entsprechenden Parametern zusammengebaut und ins Register 0x1003 eingetragen. Die COB-ID für die Emergency-Nachricht wird aus den Konfigurationsdaten ermittelt.

Parameter der Eingänge

7911

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ERRORACTIVE	BOOL	FALSE ⇒ TRUE (Flanke): sendet den anstehenden Fehler-Code TRUE ⇒ FALSE (Flanke): Wenn der Fehler NICHT mehr ansteht, wird nach einer Verzögerung von ca. 1 s eine Null-Fehlermeldung gesendet.
EEC	WORD	EEC = Emergency Error Code = Fehlermeldungsnummer
ER (Parameter-Nutzung optional)	BYTE := 0	0 = Wert aus dem Fehler-Register 0x1001 verwenden
MSEF	ARRAY [0..4] OF BYTE	MSEF = Manufacturer Specific Error Code = Zusätzlicher Fehler-Code, der vom Hersteller festgelegt wird. Wert kommt aus der Anwendung.
Write1003 (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Diese EMCY-Nachricht im Objekt 0x1003 eintragen FALSE: Funktion wird nicht ausgeführt
SendSysStatus (Parameter-Nutzung optional)	BOOL := FALSE	Send System-Status TRUE: Der Systemstatus wird überprüft und bei Vorliegen eines Fehlerstatus wird dieser ins Netzwerk übertragen. FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7912

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
39	27	kein Objekt 1001 ₁₆ in der Konfiguration
242	F2	Fehler: Einstellung ist nicht möglich



5.2.4 Bausteine: SAE J1939

Inhalt	
Bausteine: SAE J1939 Status	124
Bausteine: SAE J1939 Request	132
Bausteine: SAE J1939 Empfangen	135
Bausteine: SAE J1939 Senden	140
Bausteine: SAE J1939 Diagnose	148

2273

Für SAE J1939 stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

Bausteine: SAE J1939 Status

Inhalt	
J1939_ENABLE.....	125
J1939_GETDABYNAME	127
J1939_NAME	129
J1939_STATUS.....	131

15077

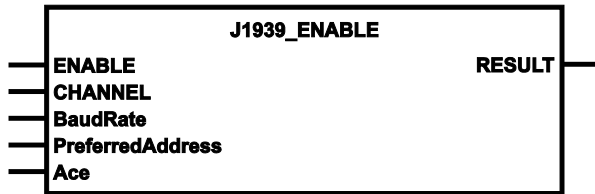
J1939_ENABLE

7641

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7642

Zur Initialisierung des J1939-Stack wird J1939_ENABLE auf TRUE=1 gesetzt.

- > Dieser FB startet auch die Soft-I/Os aus der CFG-Datei.
- > Eine andere Baudrate wird nur übernommen, wenn CAN_ENABLE nicht bereits aufgerufen wurde.

ACE = Address Claiming Enable = Freigabe Adressanforderung:

- Wenn ein ifm-Controller via J1939 mit nur einem Motorsteuergerät kommuniziert:
dann ACE = FALSE setzen.
- Wenn jedoch mehrere Motorsteuergeräte am selben Bus arbeiten:
dann ACE = TRUE setzen.
In diesem Fall müssen die Motorsteuergeräte das Address Claiming auch unterstützen!
Andernfalls riskieren Sie Adress-Überschneidungen mit nachfolgendem Systemausfall.

Parameter der Eingänge

7643

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: J1939-Kanal freigeben Ace=TRUE: Adressanforderung erfolgt FALSE: J1939-Kanal sperren
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
BaudRate (Parameter-Nutzung optional)	WORD := 250	Baudrate [kBit/s] zulässige Werte: 20, 50, 100, 125, 250, 500, 800, 1 000
PreferredAddress (Parameter-Nutzung optional)	BYTE = 252	Bevorzugte Quell-Adresse
Ace (Parameter-Nutzung optional)	BOOL := TRUE	Address Claiming Enable = Freigabe Adressanforderung TRUE: Adressanforderung freigegeben (Steuergerät ist selbst-konfigurierend) FALSE: Keine Adressanforderung

Parameter der Ausgänge

8542

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich



J1939_GETDABYNAME

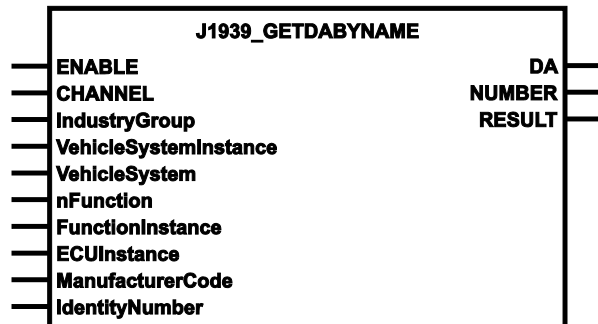
7664

= Get Destination Arbitrary Name

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7665

Über J1939_GETDABYNAME lässt sich anhand der Namensinformation die Ziel-Adresse eines oder mehrerer anderer Teilnehmer bestimmen.

- Wird an den optionalen Eingängen ein bestimmter Wert angelegt:
⇒ in der Ergebnisliste erscheinen nur die Teilnehmer, die diesen Wert besitzen.
- Wird an den optionalen Eingängen kein oder der voreingestellte Wert eingestellt:
⇒ bei der Filterung der Liste wird auf diesen Eintrag nicht geachtet.

Parameter der Eingänge

7667

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IndustryGroup (Parameter-Nutzung optional)	BYTE := 0xFF	Industry-Group = Industriegruppe des Geräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
VehicleSystemInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Fahrzeugsystems zulässige Werte = 0...15 = 0x00...0x0F 255 = 0xFF = Filter für alle
VehicleSystem (Parameter-Nutzung optional)	BYTE := 0xFF	Fahrzeugsystem zulässige Werte = 0...127 = 0x00...0x7F 255 = 0xFF = Filter für alle
nFunction (Parameter-Nutzung optional)	WORD := 0xFFFF	Funktionsnummer des Geräts zulässige Werte = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = Filter für alle
FunctionInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz der Funktion zulässige Werte = 0...31 = 0x00...0x1F 255 = 0xFF = Filter für alle
ECUInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Steuergeräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
ManufacturerCode (Parameter-Nutzung optional)	WORD := 0xFFFF	Hersteller-Code (muss bei SAE beantragt werden) zulässige Werte = 0...2047 ($2^{11}-1$) = 0x0000...0x07FF 65 535 = 0xFFFF = Filter für alle
IdentityNumber (Parameter-Nutzung optional)	DWORD := 0xFFFF FFFF	Seriennummer des Geräts (sollte nicht überschrieben werden) zulässige Werte = 0...2047 ($2^{11}-1$) 4 294 967 295 = 0xFFFF FFFF = Filter für alle

Parameter der Ausgänge

7668

Parameter	Datentyp	Beschreibung
DA	ARRAY [0..254] OF BYTE	Liste mit den gefundenen Teilnehmern 255 = Teilnehmer mit dieser Nummer nicht gefunden
NUMBER	BYTE	Anzahl der gefundenen Busteilnehmer
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8 08	Funktionsbaustein ist aktiv
242 F2	Fehler: Einstellung ist nicht möglich

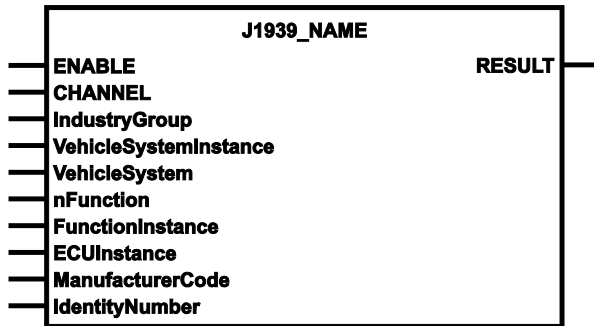
J1939_NAME

7646

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7648

Mit J1939_NAME kann dem Gerät ein Name gegeben werden, mit dem es sich im Netzwerk identifiziert.

Voreingestellt wird der Name der **ifm** verwendet.

Der Anwender hat die folgenden Möglichkeiten, den Namen des Gerätes zu ändern:

- ▶ die Informationen aus der CFG-Datei verwenden oder
- ▶ die gewünschten Daten mittels J1939_NAME überschreiben.
- Wird an den optionalen Eingängen kein oder der voreingestellte Wert eingestellt:
⇒ der voreingestellte Wert wird nicht überschrieben.

Die folgende Aufstellung zeigt die Zusammensetzung der 64-Bit-NAME-Information entsprechend SAE J1939-81:

Parameter	Datentyp	Beschreibung
Arbitrary Address Capable	1 Bit	beliebige Adresse verfügbar
Industry Group	3 Bit	Industriegruppe des Geräts
Vehicle System Instance	4 Bit	Instanz des Fahrzeugsystems
Vehicle System	7 Bit	Fahrzeugsystem
reserved	1 Bit	reserviert
Function	8 Bit	Funktion des Geräts
Function Instance	5 Bit	Instanz der Funktion
ECU Instance	3 Bit	Instanz der Steuerung
Manufacturer Code	11 Bit	Hersteller-Code (muss bei SAE beantragt werden)
Identify Number	21 Bit	Seriennummer des Geräts (sollte nicht überschrieben werden)

Tabelle: Zusammensetzung der 64-Bit-NAME-Information entsprechend SAE J1939-81

Parameter der Eingänge

7652

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: beliebige Adresse verfügbar FALSE: feste Adresse
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IndustryGroup (Parameter-Nutzung optional)	BYTE := 0xFF	Industry-Group = Industriegruppe des Geräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
VehicleSystemInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Fahrzeugsystems zulässige Werte = 0...15 = 0x00...0x0F 255 = 0xFF = Filter für alle
VehicleSystem (Parameter-Nutzung optional)	BYTE := 0xFF	Fahrzeugsystem zulässige Werte = 0...127 = 0x00...0x7F 255 = 0xFF = Filter für alle
nFunction (Parameter-Nutzung optional)	WORD := 0xFFFF	Funktionsnummer des Geräts zulässige Werte = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = Filter für alle
FunctionInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz der Funktion zulässige Werte = 0...31 = 0x00...0x1F 255 = 0xFF = Filter für alle
ECUInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Steuergeräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
ManufacturerCode (Parameter-Nutzung optional)	WORD := 0xFFFF	Hersteller-Code (muss bei SAE beantragt werden) zulässige Werte = 0...2047 ($2^{11}-1$) = 0x0000...0x07FF 65 535 = 0xFFFF = Filter für alle
IdentityNumber (Parameter-Nutzung optional)	DWORD := 0xFFFF FFFF	Seriennummer des Geräts (sollte nicht überschrieben werden) zulässige Werte = 0...2047 ($2^{11}-1$) 4 294 967 295 = 0xFFFF FFFF = Filter für alle

Parameter der Ausgänge

7661

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

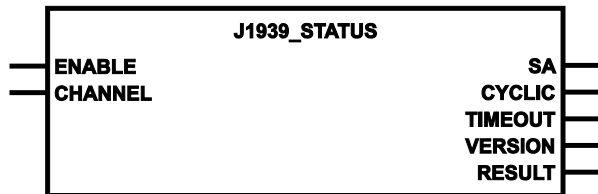
J1939_STATUS

7670

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7672

Mit J1939_STATUS können relevante Informationen zum J1939-Stack zurückgelesen werden.

Parameter der Eingänge

7673

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

Parameter der Ausgänge

7674

Parameter	Datentyp	Beschreibung
SA	BYTE	aktuelle Quell-Adresse (z.B. nach Adress-Claiming)
CYCLIC	WORD	Anzahl der zyklischen Nachrichten
TIMEOUT	BYTE	Quell-Adresse des Knotens, der Daten für Prozessabbild nicht rechtzeitig zur Verfügung gestellt hat 255 = 0xFF = alle Knoten haben rechtzeitig gesendet
VERSION	DWORD	Version der ifm-CAN-Stack-Bibliothek
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Protokoll ist aktiv
2	02	Protokoll ist inaktiv
3	03	Source-Adresse angefordert
4	04	Adresse verloren
242	F2	Fehler: Einstellung ist nicht möglich

Bausteine: SAE J1939 Request

Inhalt	
J1939_SPEC_REQ	133
J1939_SPEC_REQ_MULTI	134
	15079



J1939_SPEC_REQ

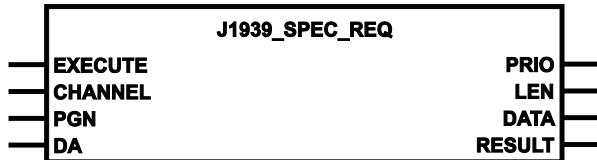
15023

= J1939 Specific Request

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

15026

J1939_SPECIFIC_REQUEST fragt eine spezifizierte Nachricht bei einer anderen Steuerung an und empfängt sie.

Beim Request einer Multiframe-Nachricht:

- der FB gibt die ersten 8 Bytes der Daten aus
- RESULT zeigt einen Fehler an

Parameter der Eingänge

15028

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = P arameter G roup N umber = Parameter-Gruppennummer zulässig = 1...??? = 0x00000001...0x???
DA	BYTE	J1939-Adresse des angefragten Geräts

Parameter der Ausgänge

15029

Parameter	Datentyp	Beschreibung
PRIO	BYTE	Nachrichten-Priorität in der PDU (Parameter Data Unit) zulässig = 0...7
LEN	WORD	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5 05	FB ist aktiv – noch keine Daten empfangen
64 40	Fehler: Multiframe empfangen
242 F2	Fehler: Einstellung ist nicht möglich

J1939_SPEC_REQ_MULTI

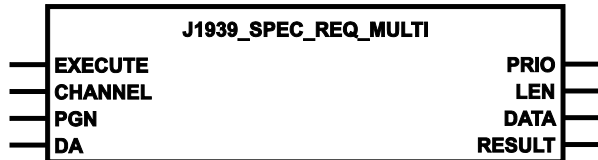
15033

= J1939 Specific Request Multiframe Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

15036

J1939_SPECIFIC_REQUEST fragt eine spezifizierte Multiframe-Nachricht bei einer anderen Steuerung an und empfängt sie.

Parameter der Eingänge

15037

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = P arameter G roup N umber = Parameter-Gruppennummer zulässig = 1...785 = 0x00000001...0x06F9
DA	BYTE	J1939-Adresse des angefragten Geräts

Parameter der Ausgänge

15038

Parameter	Datentyp	Beschreibung
PRIO	BYTE	Nachrichten-Priorität der PDU (Parameter Data Unit) zulässig = 0...7
LEN	WORD	Anzahl der zu übertragenden Daten-Bytes zulässig = 1...1785 = 0x0001...0x06F9
DATA	ARRAY [0..1784] OF BYTE	Empfangene Daten (1...1785 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5 05	FB ist aktiv – noch keine Daten empfangen
242 F2	Fehler: Einstellung ist nicht möglich

Bausteine: SAE J1939 Empfangen

Inhalt	
J1939_RX.....	136
J1939_RX_FIFO.....	137
J1939_RX_MULTI.....	139
	15081



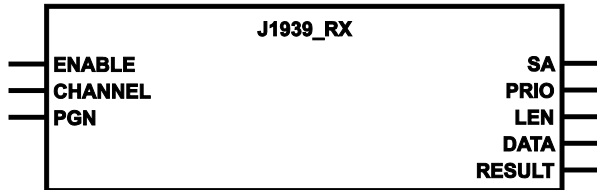
J1939_RX

7724

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7725

J1939_RX ist die einfachste Methode zum Empfangen von Single-Frame-Nachrichten. Es wird die zuletzt auf dem CAN-Bus gelesene Nachricht zurückgegeben.

Parameter der Eingänge

7726

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = Parameter Group Number = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF

! Die PGN = 0 wird nicht verwendet.

Parameter der Ausgänge

7727

Parameter	Datentyp	Beschreibung
SA	BYTE	Source Address des Senders
PRIO	BYTE	Nachrichten-Priorität der PDU (Parameter Data Unit) zulässig = 0...7
LEN	WORD	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
8 08	Funktionsbaustein ist noch nicht ausgeführt
242 F2	Fehler: Einstellung ist nicht möglich

J1939_RX_FIFO

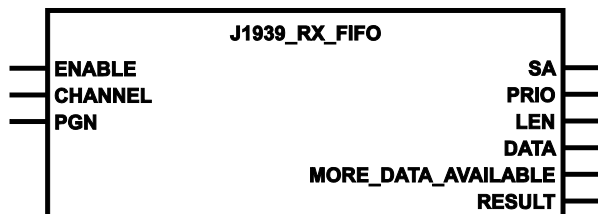
7732

= J1939 RX with FIFO

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7733

J1939_RX_FIFO ermöglicht es, alle spezifizierten Nachrichten zu empfangen und nacheinander aus einem FIFO zu lesen.

Parameter der Eingänge

7734

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = Parameter Group Number = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF

! Die PGN = 0 wird nicht verwendet.

Parameter der Ausgänge

7735

Parameter	Datentyp	Beschreibung
SA	BYTE	Source Address des Senders
PRIO	BYTE	Nachrichten-Priorität in der PDU (Parameter Data Unit) zulässig = 0...7
LEN	BYTE	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten im FiFo vorhanden FALSE: keine weiteren Daten im FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

J1939_RX_MULTI

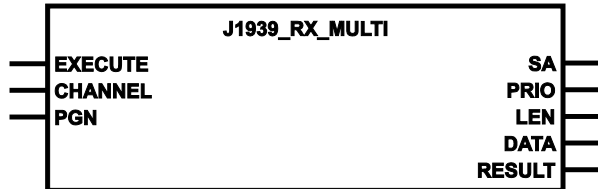
7736

= J1939 RX Multiframe Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7741

Mit J1939_RX_MULTI ist der Empfang von Multiframe-Nachrichten möglich.

Parameter der Eingänge

7743

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = P arameter G roup N umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF

! Die PGN = 0 wird nicht verwendet.

Parameter der Ausgänge

7744

Parameter	Datentyp	Beschreibung
SA	BYTE	S ource A ddress des Senders
PRIO	BYTE	Nachrichten-Priorität in der PDU (Parameter Data Unit) zulässig = 0...7
LEN	WORD	Anzahl der empfangenen Bytes zulässige Werte = 0000 0000...0000 06F9 ₁₆ = 0...1 785 ₁₀
DATA	ARRAY [0..1784] OF BYTE	Empfangene Daten (1...1785 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5 05	FB ist aktiv – noch keine Daten empfangen
242 F2	Fehler: Einstellung ist nicht möglich

Bausteine: SAE J1939 Senden

Inhalt	
J1939_TX	141
J1939_TX_ENH.....	142
J1939_TX_ENH_CYCLIC	144
J1939_TX_ENH_MULTI.....	146

15083

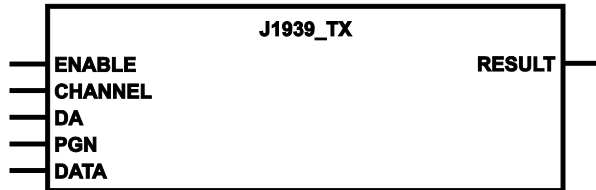
J1939_TX

7688

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7689

J1939_TX ist die einfachste Methode zum Versenden von Single-Frame-Nachrichten.

Parameter der Eingänge

7690

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = Destination Address = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
PGN	DWORD	PGN = Parameter Group Number = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7693

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
242 F2	Fehler: Einstellung ist nicht möglich
250 FA	Fehler: FiFo ist voll – Daten wurden verloren

J1939_TX_ENH

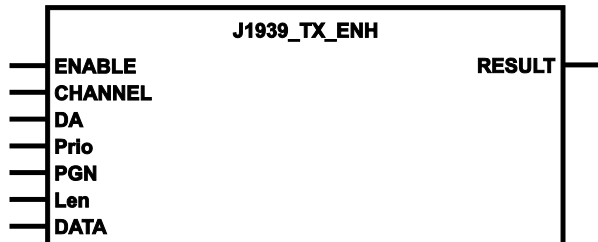
7696

= J1939 TX enhanced

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7697

Zusätzliche Einstellmöglichkeiten bietet J1939_TX_ENH (für: enhanced) für Single-Frame-Nachrichten:

- Sende-Priorität
- Datenlänge

Multi-Frame Nachrichten → **J1939_TX_ENH_MULTI** (→ Seite [146](#)).

Parameter der Eingänge

7702

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = Destination Address = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
Prio (Parameter-Nutzung optional)	BYTE := 3	Nachrichten-Priorität zulässige Werte = 0...7
PGN	DWORD	PGN = Parameter Group Number = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
Len (Parameter-Nutzung optional)	BYTE := 8	Anzahl der zu sendenden Bytes zulässige Werte = 0...8
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7969

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren



J1939_TX_ENH_CYCLIC

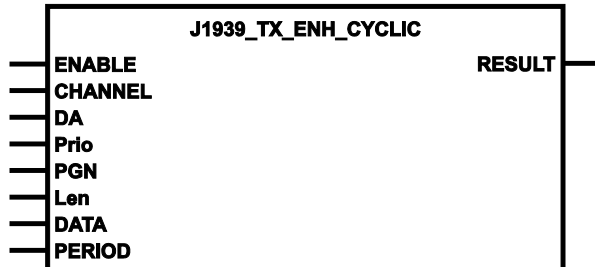
7716

= J1939 TX enhanced Cyclic

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyyzz.LIB

Symbol in CODESYS:



Beschreibung

7718

J1939_TX_ENH_CYCLIC dient dem zyklischen Versand von CAN-Nachrichten.

Der FB entspricht ansonsten **J1939_TX_ENH** (→ Seite [142](#)).

► Mit dem Parameter PERIOD die Periodendauer einstellen.

! Eine zu kurze Periodendauer kann zu einer hohen Buslast führen!
Die Buslast kann das Verhalten des Gesamtsystems beeinträchtigen.

Parameter der Eingänge

7719

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = Destination Address = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
Prio (Parameter-Nutzung optional)	BYTE := 3	Nachrichten-Priorität zulässige Werte = 0...7
PGN	DWORD	PGN = Parameter Group Number = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
Len (Parameter-Nutzung optional)	BYTE := 8	Anzahl der zu sendenden Bytes zulässige Werte = 0...8
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)
PERIOD	TIME	Periodendauer

Parameter der Ausgänge

7720

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich



J1939_TX_ENH_MULTI

7699

= J1939 TX enhanced Multiframe Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7705

Die Übertragung von Multi-Frame-Nachrichten erfolgt mit J1939_TX_ENH_MULTI.

Der FB entspricht **J1939_TX_ENH** (→ Seite [142](#)). Zusätzlich kann hier bestimmt werden, ob die Übertragung als BAM (**B**roadcast **A**nnounce **M**essage) erfolgen soll.

Parameter der Eingänge

7712

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = Destination Address = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
Prio (Parameter-Nutzung optional)	BYTE := 3	Nachrichten-Priorität zulässige Werte = 0...7
PGN	DWORD	PGN = Parameter Group Number = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
Len (Parameter-Nutzung optional)	WORD := 8	Anzahl der zu übertragenden Daten-Bytes zulässig = 1...1 785 = 0x0001...0x06F9
DATA	ARRAY [0..1784] OF BYTE	Zu sendende Daten (1...1785 Bytes)
Bam (Parameter-Nutzung optional)	BOOL := FALSE	BAM = Broadcast Announce Message = Nachricht an alle Teilnehmer TRUE: Multi-Frame-Übertragung als BAM Nachricht an alle Teilnehmer FALSE: automatisch; Nachricht nur an Zieladresse

Parameter der Ausgänge

7714

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
65	41	Fehler: senden ist nicht möglich
242	F2	Fehler: Einstellung ist nicht möglich



Bausteine: SAE J1939 Diagnose

Inhalt	
J1939_DM1RX	149
J1939_DM1TX.....	151
J1939_DM1TX_CFG	154
J1939_DM3TX.....	155

15085

J1939_DM1RX

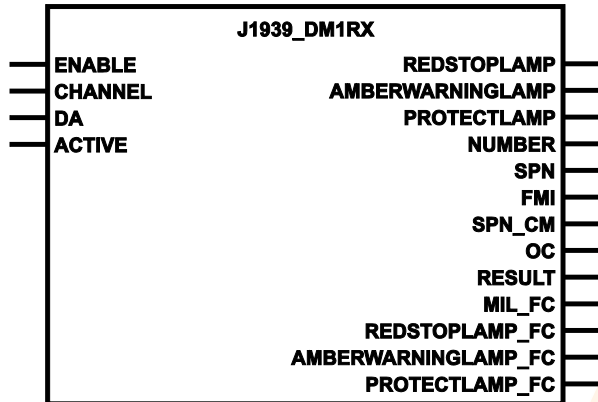
14977

= J1939 Diagnostic Message 1 RX

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7761

J1939_RX_DM1 empfängt Diagnosemeldungen DM1 oder DM2 von anderen ECUs.

Parameter der Eingänge

14979

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = Destination Address = Zieladresse der ECU, von der die DTCs geholt werden sollen. DA = 254: DTCs aus Gerät selbst lesen
ACTIVE	BOOL	TRUE: aktive DTCs (DM1) lesen FALSE: davor aktive DTCs (DM2) lesen

Parameter der Ausgänge

14980

Parameter	Datentyp	Beschreibung
REDSTOPLAMP	BOOL	Rote Stopp-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
AMBERWARNINGLAMP	BOOL	Gelbe Warn-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
PROTECTLAMP	BOOL	Schutz-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
NUMBER	BYTE	Anzahl der empfangenen DTCs (0...8)
SPN	WORD	Suspect Parameter Number
FMI	BYTE	Failure-Mode-Indicator zulässige Werte = 0...31 = 0x00...0x1F
SPN_CM	BOOL	Conversion Method
OC	BYTE	Occurrence Count = Ereigniszähler
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)
MIL_FC	BYTE	Status der elektronischen Komponente: Fehlfunktion-Anzeigelampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
REDSTOPLAMP_FC	BYTE	Status der elektronischen Komponente: Rote Stopp-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
AMBERWARNINGLAMP_FC	BYTE	Status der elektronischen Komponente: Gelbe Warnlampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
PROTECTLAMP_FC	BYTE	Status der elektronischen Komponente: Schutz-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	FB ist aktiv – keine Daten wurden empfangen
242	F2	Fehler: Einstellung ist nicht möglich

J1939_DM1TX

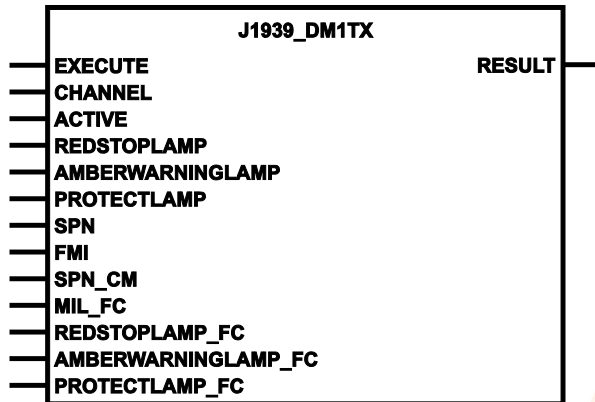
14993

= J1939 Diagnostic Message 1 TX

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

7747

Mit J1939_TX_DM1 (DM = **D**agnostic-**M**essage) kann die Steuerung nur eine aktive Fehlermeldung an den CAN-Stack übergeben.

- > Diese Meldung wird in die Hardware-Konfiguration gesichert
- > Meldung wird als aktiv markiert und sekundlich als DM1 gesendet.
- > Falls der Fehler bereits auftrat, wird der Ereignis-Zähler inkrementiert.
- ❗ Der Ereignis-Zähler wird vom CAN-Stack verwaltet.
- > Es erfolgt eine ODER-Verknüpfung aller Bits der Trouble-Codes. Sobald in einem der Trouble-Codes ein Bit gesetzt ist, ist es auch im Lampenstatus gesetzt.

Sobald eine Anfrage nach DM2 kommt, kann der CAN-Stack die entsprechenden Informationen aus der Hardware-Konfiguration auslesen und versenden.

- > Bei Eintreffen einer DM3-Nachricht werden alle nicht aktiven Fehler im Fehlerspeicher in der Hardware-Konfiguration gelöscht.

Parameter der Eingänge

14995

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ACTIVE	BOOL	TRUE: DTC ist aktiv Wird zyklisch (1x je Sekunde) als DM1 gesendet FALSE: DTC ist nicht mehr aktiv Wird in der Hardware-Konfiguration gesichert Wird bei Anfrage als DM2 gesendet
REDSTOPLAMP	BOOL	Rote Stopp-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
AMBERWARNINGLAMP	BOOL	Gelbe Warn-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
PROTECTLAMP	BOOL	Schutz-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
SPN	WORD	Suspect Parameter Number
FMI	BYTE	Failure-Mode-Indicator zulässige Werte = 0...31 = 0x00...0x1F
SPN_CM	BOOL	Conversion Method
MIL_FC	BYTE	Status der elektronischen Komponente: Fehlfunktion-Anzeigelampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
REDSTOPLAMP_FC	BYTE	Status der elektronischen Komponente: Rote Stopp-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
AMBERWARNINGLAMP_FC	BYTE	Status der elektronischen Komponente: Gelbe Warnlampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
PROTECTLAMP_FC	BYTE	Status der elektronischen Komponente: Schutz-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken

Parameter der Ausgänge

7750

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Daten wurden in Fehlerspeicher aktiv gekennzeichnet
242	F2	Fehler: Einstellung ist nicht möglich



J1939_DM1TX_CFG

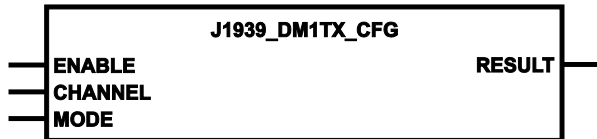
15424

= J1939 Diagnostic Message 1 TX configurable

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_V02.00.02.LIB oder höher

Symbol in CODESYS:



Beschreibung

15426

Ab Laufzeitsystem V03.00.03 sendet der CAN-Stack automatisch sekundliche DM1-Nachrichten, sobald der FB **J1939_ENABLE** (→ Seite [125](#)) für das betreffende CAN-Interface aufgerufen wurde.

- Den FB J1939_DM1TX_CFG nutzen, wenn dieses automatische, zyklische Senden von DM1-Nachrichten des CAN-Stacks nicht gewünscht ist.

Folgende Modi für die zyklische Sendung von DM1-Nachrichten stehen mit dem FB zur Verfügung:

MODE = 0 (voreingestellt)	Der CAN-Stack sendet normkonform, sekundlich DM1 "zero active faults"-Nachrichten. Manuelles Senden von DM1-Nachrichten über den FB J1939_DM1TX (→ Seite 151) ist möglich.
MODE = 1	Der CAN-Stack sendet keine zyklischen DM1 "zero active faults"-Nachrichten. Auf DM2-Anfragen wird automatisch geantwortet. manuelles Senden von DM1-Nachrichten über den FB J1939_DM1TX (→ Seite 151) ist möglich.
MODE = 2	Der CAN-Stack sendet keine zyklischen DM1 "zero active faults"-Nachrichten Der CAN-Stack sendet auch keine automatische Antwort auf DM2-Anfragen.

Parameter der Eingänge

15427

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
MODE	BYTE := 0	Betriebsart des Bausteins zulässig = 0...2 (→ Beschreibung des FBs)

Parameter der Ausgänge

15429

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	Funktionsbaustein-Ausführung ohne Fehler beendet
242 F2	Fehler: Einstellung ist nicht möglich

J1939_DM3TX

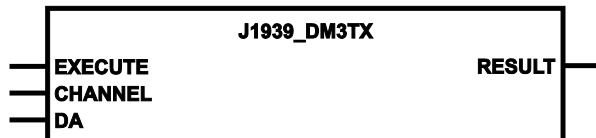
15002

= J1939 Diagnostic Message 3 TX

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_J1939_NT_Vxxyyzz.LIB

Symbol in CODESYS:



Beschreibung

15004

J1939_DM3TX (DM = **D**agnostic-**M**essage) ermöglicht das Löschen der inaktiven DTCs auf einem anderen Gerät.

- > Bei Eintreffen einer DM3-Nachricht werden alle nicht aktiven Fehler im Fehlerspeicher in der Hardware-Konfiguration gelöscht.

Parameter der Eingänge

15006

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇒ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = D estination A ddress = Zieladresse der ECU, auf der die DTCs gelöscht werden sollen. DA = 254: DTCs (DM2) im Gerät selbst löschen

Parameter der Ausgänge

15008

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich

5.2.5 Bausteine: System

Inhalt	
FLASH_INFO	157
FLASH_READ	158
GET_APP_INFO	159
GET_HW_INFO	160
GET_IDENTITY	161
GET_SW_INFO	162
GET_SW_VERSION	163
MEM_ERROR	164
MEMCPY	165
OHC	167
SET_IDENTITY	169
SET_LED	170
SET_PASSWORD	172
TIMER_READ_US	173

15067

Hier zeigen wir Ihnen **ifm**-Funktionen, mit denen Sie Folgendes erreichen:

- Speicherinhalte verwalten
- Informationen von Software und Hardware lesen
- diverse Daten und Parameter setzen oder lesen

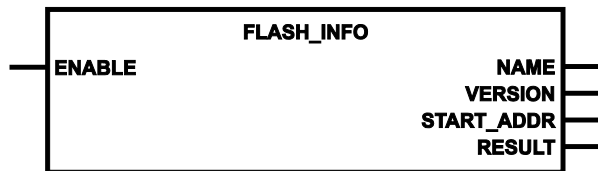
FLASH_INFO

11580

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxyxyz.LIB

Symbol in CODESYS:



Beschreibung

11588

FLASH_INFO liest die Informationen aus dem User-Flash-Speicher:

- Name des Speicherbereichs (vom User vorgegeben),
- Software-Version,
- Startadresse (für einfaches Lesen mit IEC-Struktur).

Parameter der Eingänge

11589

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert

Parameter der Ausgänge

11590

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name des Speicherbereichs (vom User vorgegeben)
VERSION	STRING(24)	Software-Version
START_ADDR	DWORD	Startadresse der Daten
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
157 9D	Software-Header ist ungültig (CRC-Fehler)

FLASH_READ

8147

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

11579

FLASH_READ ermöglicht das Lesen unterschiedlicher Datentypen direkt aus dem Flash-Speicher.

Der FB liest den Inhalt ab der Adresse von SRC aus dem Flash-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben sind.

- Die aus SRC + LEN resultierende Adresse muss $\leq 65\,408$ sein.
- Für die Zieladresse DST gilt:
 - ❗ Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Parameter der Eingänge

8148

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DST	DWORD	Startadresse im Zielspeicher ❗ Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	relative Anfangsadresse im Speicher zulässig = 0...65 407 = 0x0000 0000...0x0000 FF7F
LEN	WORD	Anzahl (≥ 1) der zu übertragenden Daten-Bytes

Parameter der Ausgänge

8152

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
152 98	unzulässiger Speicherbereich: • ungültige Quell-Adresse • ungültige Ziel-Adresse • ungültige Anzahl Bytes

GET_APP_INFO

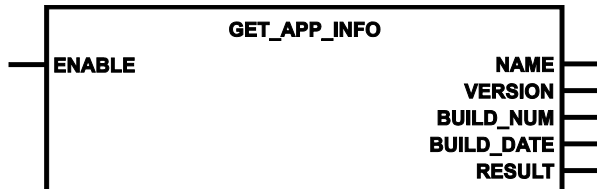
11581

= Get Application Information

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

11593

GET_APP_INFO liefert Informationen über das im Gerät gespeicherte Anwendungsprogramm:

- Name (= Dateiname des CODESYS-Projekts),
- Version (= aus CODESYS-Menü [Projekt] > [Projektinformation] > [Version]),
- eindeutige CODESYS-Build-Nummer,
- CODESYS-Build-Datum.

Parameter der Eingänge

11594

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

11595

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name der Anwendung
VERSION	STRING(24)	Version des Anwendungsprogramms
BUILD_NUM	STRING(24)	eindeutige CODESYS-Build-Nummer (z.B.: "45")
BUILD_DATE	STRING(24)	CODESYS-Build-Datum (z.B.: "20111006123800")
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

GET_HW_INFO

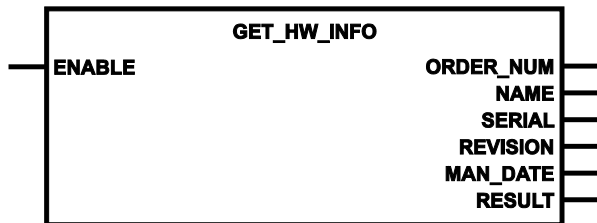
11582

= Get Hardware Information

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

11599

GET_HW_INFO liefert Informationen über die Hardware des Geräts:

- ifm-Artikelnummer (z.B. CR0403),
- Artikelbezeichnung,
- eindeutige Seriennummer,
- Hardware-Revision,
- Produktionsdatum.

Parameter der Eingänge

11600

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

11601

Parameter	Datentyp	Beschreibung
ORDER_NUM	STRING(24)	ifm-Artikelnummer (z.B.: CR0403)
NAME	STRING(24)	Artikelbezeichnung (z.B.: "BasicController 12/12")
SERIAL	STRING(24)	Seriennummer des Geräts (z.B.: "000045784")
REVISION	STRING(24)	Hardware-Revisionsstand des Geräts (z.B.: "V01.00.01")
MAN_DATE	STRING(24)	Herstellungsdatum des Geräts (z.B.: "20111007123800")
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

GET_IDENTITY

8166

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

15411

GET_IDENTITY liest die im Gerät gespeicherte Kennung (wurde zuvor mit **SET_IDENTITY** (→ Seite [169](#)) gespeichert).

Parameter der Eingänge

8167

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

8168

Parameter	Datentyp	Beschreibung
APP_IDENTITY	STRING(80)	Kennung der Anwendung als Zeichenkette von max. 80 Zeichen, z.B.: "Crane1704"
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
155	9B	Wert konnte nicht gelesen werden

GET_SW_INFO

11583

= Get Software Information

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

11596

GET_SW_INFO liefert Informationen über die System-Software des Geräts:

- Software-Name,
- Software-Version,
- Build-Nummer,
- Build-Datum.

Parameter der Eingänge

11597

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

11598

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name der System-Software (z.B.: "BasicSystem")
VERSION	STRING(24)	Version der System-Software (z.B.: "V02.00.03")
BUILD_NUM	STRING(24)	Build-Nummer der System-Software (z.B.: "45")
BUILD_DATE	STRING(24)	Build-Datum der System-Software (z.B.: "20111006123800")
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

GET_SW_VERSION

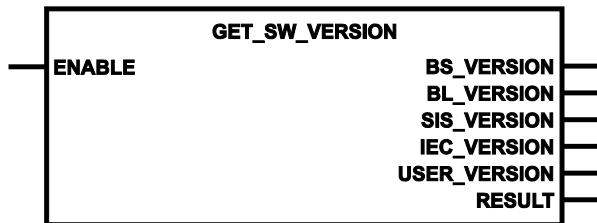
14763

= Get Software-Version

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

14765

GET_SW_VERSION liefert die Informationen zur Software im Gerät:

- BasicSystem-Version
- Bootloader-Version
- SIS-Version
- IEC-Anwendungsprogramm-Version
- IEC-User-Flash-Version

Parameter der Eingänge

14766

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

14767

Parameter	Datentyp	Beschreibung
BS_VERSION	STRING(24)	Basic-System-Version
BL_VERSION	STRING(24)	Bootloader-Version
SIS_VERSION	STRING(24)	SIS-Version (SIS = System Information Service)
IEC_VERSION	STRING(24)	IEC-Anwendungsprogramm-Version
USER_VERSION	STRING(24)	IEC-User-Flash-Version
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex	Beschreibung
0 00	FB ist inaktiv
1 01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

MEM_ERROR

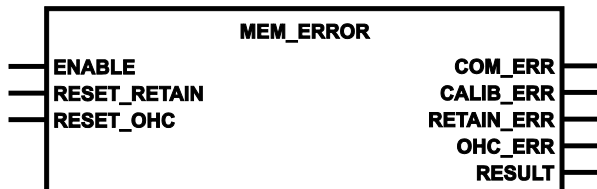
14770

= Memory Error

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

14772

MEM_ERROR meldet Fehler in einigen Parametern oder im Speicher.

Über entsprechende FB-Eingänge lassen sich die Speicherbereiche löschen.

Parameter der Eingänge

14773

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
RESET_RETAIN	BOOL	TRUE: Remanenten Retain-Speicher löschen FALSE: keine Änderung am Speicherinhalt
RESET_OHC	BOOL	TRUE: Remanenten OHC-Speicher löschen FALSE: keine Änderung am Speicherinhalt

Parameter der Ausgänge

14774

Parameter	Datentyp	Beschreibung
COM_ERR	BOOL	Download ID und Baudrate sind auf Voreinstellwerte gesetzt (Download-Parameter gingen verloren)
CALIB_ERR	BOOL	Kalibrierungswerte sind ungültig (analoge Eingänge, PWM-Ausgänge, Systemspannungen)
RETAIN_ERR	BOOL	Retain-Speicher ist ungültig (z.B. wegen zu starken Magnetfelds teilweise gelöscht)
OHC_ERR	BOOL	Betriebsstundenzähler OHC Werte sind ungültig (z.B. wegen zu starken Magnetfelds teilweise gelöscht)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

MEMCPY

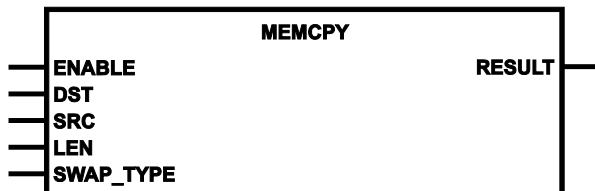
8160

= Memory Copy

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxyxyz.LIB

Symbol in CODESYS:



Beschreibung

412

MEMCPY ermöglicht das Schreiben und Lesen unterschiedlicher Datentypen direkt in den Speicher. Der FB schreibt den Inhalt ab der Adresse von SRC an die Adresse DST.

- Für die Adressen SRC und DST gilt:
 - ❗ Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
- > Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben wurden. Dadurch ist es auch möglich, genau ein Byte einer Word-Variablen zu übertragen.

Parameter der Eingänge

8162

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DST	DWORD	Startadresse im Zielspeicher ❗ Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	Quell-Adresse
LEN	WORD	Anzahl der zu kopierenden Bytes SWAP_TYPE=1: LEN muss Mehrfaches von 2 sein SWAP_TYPE=2: LEN muss Mehrfaches von 4 sein
SWAP_TYPE	BYTE	Byte-Reihenfolge tauschen: 0 = kein Tausch z.B.: 1A 2B 3C 4D ⇒ 1A 2B 3C 4D 1 = 2 Bytes tauschen (WORD, INT, ...) z.B.: 1A 2B 3C 4D ⇒ 2B 1A 4D 3C ❗ LEN muss ein Mehrfaches von 2 sein! 2 = 4 Bytes tauschen (DWORD, DINT, REAL, TIME, ...) z.B.: 1A 2B 3C 4D ⇒ 4D 3C 2B 1A ❗ LEN muss ein Mehrfaches von 4 sein!

Parameter der Ausgänge

8163

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
152	98	unzulässiger Speicherbereich: • ungültige Quell-Adresse • ungültige Ziel-Adresse • ungültige Anzahl Bytes
156	9C	unzulässige Werte: • ungültiger Wert für SWAP_TYPE • LEN passt nicht zu SWAP_TYPE

OHC

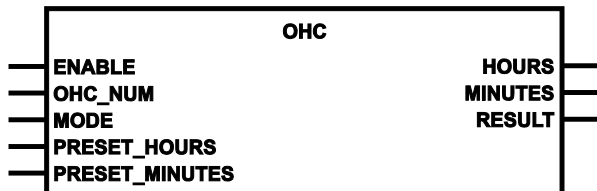
14777

= Operating Hours Counter

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

14778

OHC bietet 4 universell verwendbare Betriebsstundenzähler.

Aber bei Hardware-Stand < AD: nur 2 Betriebsstundenzähler möglich.

Gültiger Zählbereich: 0:00...4 294 967 295:59 Stunden (= 490 293 Jahre, 25 Tage, 15 Stunden)

! Falls Hardware-Stand des Geräts < AD:
Den Speicherbereich für OHC einmalig zurücksetzen:
► Im FB **MEM_ERROR** (→ Seite [164](#)) den Eingang RESET_OHC = TRUE setzen!
> Erst jetzt sind die Betriebsstundenzähler verwendbar.

Parameter der Eingänge

14779

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
OHC	BYTE	Operating Hours Counter = Betriebsstundenzähler Nummer des Zählers (0...3)
MODE	BYTE	Betriebsmodus des Zählers Zulässige Werte = 0 = Zähler anhalten 1 = Zählen fortsetzen beim zuletzt gespeicherten Wert 2 = Zähler zurücksetzen 3 = Zähler voreinstellen mit den folgenden Werten
PRESET_HOURS	DWORD	Voreinstellwert Stunden (0...4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
PRESET_MINUTES	BYTE	Voreinstellwert Minuten (0...59 = 0x00...0x3B)

Parameter der Ausgänge

14780

Parameter	Datentyp	Beschreibung
HOURS	DWORD	Zählerstand Stunden (0...4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
MINUTES	BYTE	Zählerstand Minuten (0...59 = 0x00...0x3B)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
130	82	Zählernummer in OHC_NUM ist ungültig
131	83	Voreinstellwert ist ungültig
132	84	Moduseinstellung ist ungültig
158	9E	Remanent-Speicher ist ungültig (CRC-Fehler)

SET_IDENTITY

8174

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

8535

SET_IDENTITY setzt eine anwendungsspezifische Programmkennung.

Mit dem FB kann durch das Anwendungsprogramm eine Programmkennung erzeugt werden.

- Diese Kennung kann zur Identifizierung des geladenen Programms ausgelesen werden:
 - über das Maintenance-Tool
 - im Anwendungsprogramm über den FB **GET_IDENTITY** (→ Seite [161](#))

Parameter der Eingänge

8175

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
APP_IDENT	STRING(80)	Kennung der Anwendung als Zeichenkette von max. 80 Zeichen, z.B.: "Crane1704" Rücksetzen mit APP_IDENT = ""

Parameter der Ausgänge

8176

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet

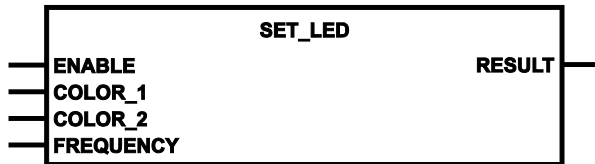
SET_LED

8052

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxyxyz.LIB

Symbol in CODESYS:



Beschreibung

8054

Mit SET_LED können im Anwendungsprogramm Frequenz und Farbe der Status-LED geändert werden.

⚠ Wird der Blinkmodus im Anwendungsprogramm geändert, gilt die Voreinstellung-Tabelle nicht mehr (→ Kapitel **Status-LED** (→ Seite [16](#))).

Parameter der Eingänge

8223

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
COLOR_1	BYTE	LED-Farbe für "eingeschaltet" Farbkonstante aus der Datenstruktur "System LED Color" zulässige Werte: 00 = LED_BLACK (= LED aus) 01 = LED_RED 02 = LED_GREEN 03 = LED_YELLOW
COLOR_2	BYTE	LED-Farbe für "ausgeschaltet" Farbkonstante aus der Datenstruktur "System LED Color" zulässige Werte: 00 = LED_BLACK (= LED aus) 01 = LED_RED 02 = LED_GREEN 03 = LED_YELLOW
FREQUENCY	BYTE	LED-Blinkfrequenz Frequenzkonstante aus der Datenstruktur "System LED Frequency"; zulässige Werte: 00 = LED_0HZ = dauernd EIN 01 = LED_05HZ = blinkt mit 0,5 Hz 02 = LED_1Hz = blinkt mit 1 Hz 04 = LED_2HZ = blinkt mit 2 Hz 10 = LED_5HZ = blinkt mit 5 Hz

Parameter der Ausgänge

8227

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez hex		Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
133	85	Wert für FREQUENCY ist ungültig
151	97	Wert für Farbe ist ungültig



SET_PASSWORD

8178

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

8179

SET_PASSWORD setzt Benutzerkennung für Programm- und Speicher-Upload mit dem Maintenance-Tool.

Ist die Benutzerkennung aktiv, kann durch das Maintenance-Tool das Anwendungsprogramm oder der Datenspeicher nur ausgelesen werden, wenn das richtige Passwort eingegeben wurde.

Wird an den Eingang PASSWORD ein Leer-String (Default-Zustand) übergeben, so wird ein zuvor gesetztes Passwort zurückgesetzt. Ein Upload des Anwendungsprogramms oder des Datenspeichers ist dann jederzeit möglich.

! Beim Laden eines neuen Anwendungsprogramms wird das Passwort wieder zurückgesetzt.

Parameter der Eingänge

8180

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
PASSWORD	STRING(16)	Benutzerkennung Wenn PASSWORD = "", dann ist Zugriff ohne Passworteingabe möglich.

Parameter der Ausgänge

8181

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert		Beschreibung
dez	hex	
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet

TIMER_READ_US

8219

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_CR0451_Vxyxyz.LIB

Symbol in CODESYS:



Beschreibung

660

TIMER_READ_US liest die aktuelle Systemzeit in [µs] aus.

Mit Anlegen der Versorgungsspannung bildet das Gerät einen Zeittakt, der in einem Register aufwärts gezählt wird. Dieses Register kann mittels des FB-Aufrufes ausgelesen werden und z.B. zur Zeitmessung genutzt werden.



Info

Der System-Timer läuft maximal bis zum Zählerwert 1h 11min 34s 967ms 295µs und startet anschließend wieder mit 0.

Parameter der Ausgänge

8220

Parameter	Datentyp	Beschreibung
TIME_US	DWORD	Aktuelle Systemzeit [µs]
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet

5.2.6 Bausteine: Grafik

Inhalt

Bausteine: Grafik Hilfe.....	174
Bausteine: Grafik Visualisierung	180
	15294

Bausteine: Grafik Hilfe

Inhalt

GET_TEXT_FROM_FLASH.....	175
NORM_DINT	177
NORM_REAL	178
TOGGLE.....	179
	15296

Hier finden Sie weitere **ifm**-Bausteine, die Sie bei der Programmierung der Grafikoberfläche unterstützen.

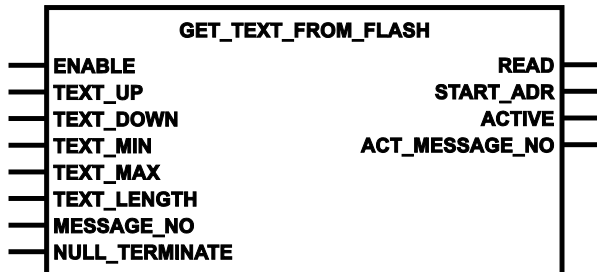
GET_TEXT_FROM_FLASH

3196

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_PDMsmart_UTIL_Vxxyzz.Lib

Symbol in CODESYS:



Beschreibung

11651

GET_TEXT_FROM_FLASH steuert **FLASH_READ** (→ Seite [158](#)) an, um Texte vom Typ STRING direkt auszulesen.

Im Gegensatz zum PDM360 und PDM360compact hat dieses Gerät kein Dateisystem. Daher bieten sich hier Flash-Speicher zum Ablegen von Textmeldungen an. Zum Auslesen dieser Speicherbereiche wird FLASH_READ benötigt.

Um nun gezielt einen oder auch mehrere Texte auszulesen, muss die Startadresse des Textes im Speicher berechnet werden. Diese Berechnung und auch das Setzen/Rücksetzen des ENABLE-Eingangs erfolgt in GET_TEXT_FROM_FLASH.

Die Organisation der Texte im Speicher muss nach folgenden Regeln erfolgen:

Textlänge

Die Textlänge sollte für alle Texte gleich sein und ist wegen der Displaygröße des Geräts auf jeweils maximal 30 Zeichen begrenzt.

Erstellung der Texte

Die Texte sollten mit einem Tabellenkalkulationsprogramm (z.B. Excel) erstellt und im CSV-Format gespeichert werden. Diese CSV-Datei kann mit dem ifm-Maintenance-Tool direkt in den gewünschten Speicherbereich geladen werden.

→ **ecomatmobile**-DVD "Software, tools and documentation".

Ein STRING wird automatisch vom Programmiersystem mit einem NULL-Byte abgeschlossen. Daher belegt ein Text mit 30 Zeichen 31 Bytes im Speicher. Der FB berücksichtigt das bei der Berechnung. Vom angegebenen Flash-Speicher des Geräts (→ Kapitel **Verfügbarer Speicher** (→ Seite [13](#))) 128 Bytes für den Header abziehen. Bei einer Textlänge von 30 Zeichen können somit $65\,408 / 31 = 2\,109$ Texte im Flash-Speicher gespeichert werden.

Parameter der Eingänge

3302

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
TEXT_UP	BOOL	Flanke FALSE ⇒ TRUE: nächsten Text lesen
TEXT_DOWN	BOOL	Flanke FALSE ⇒ TRUE: vorherigen Text lesen
TEXT_MIN	WORD	untere Grenze für MESSAGE_NO
TEXT_MAX	WORD	obere Grenze für MESSAGE_NO
TEXT_LENGTH	BYTE	Textlänge
MESSAGE_NO	WORD	Textnummer
NULL_TERMINATE	BOOL	TRUE: String hat Null-Terminierung FALSE: String hat keine Null-Terminierung

Parameter der Ausgänge

15596

Parameter	Datentyp	Beschreibung
READ	BOOL	Kommando Lesen ► Dieses Signal auf den Eingang ENABLE des FBs FLASH_READ legen!
START_ADR	WORD	berechnete Startadresse ► Dieses Signal auf den Eingang SRC des FBs FLASH_READ legen!
ACTIV	BOOL	TRUE: FB ist aktiv (wenn Eingang ENABLE = 1)
ACT_MESSAGE_NO	WORD	aktuelle Textnummer

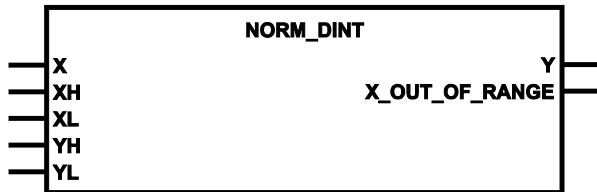
NORM_DINT

13240

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_PDMsmart_UTIL_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

3307

NORM_DINT normiert einen Wert innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen.

Der FB normiert einen Wert vom Typ DINT, der innerhalb der Grenzen XH und XL liegt, auf einen Ausgangswert innerhalb der Grenzen YH und YL. Der FB wird z.B. bei der Erzeugung von PWM-Werten aus analogen Eingangsgrößen genutzt.

HINWEIS

- ▶ Der Eingangswert für X muss sich im definierten Bereich zwischen XL und XH befinden! Der FB prüft NICHT den Wert X auf Plausibilität. Außerhalb dieses Wertebereiches wird der Ausgang X_OUT_OF_RANGE gesetzt.
- ▶ Das Ergebnis der Berechnung $(XH-XL) \cdot (YH-YL)$ muss im Wertebereich des Datentyps DINT (- 2 147 483 648...2 147 483 647) bleiben!
- > Bedingt durch die Rundungsfehler können Abweichungen beim normierten Wert um 1 auftreten.
- > Werden die Grenzen (XH/XL oder YH/YL) invertiert angegeben, erfolgt auch die Normierung invertiert.

Parameter der Eingänge

3308

Parameter	Datentyp	Beschreibung
X	DINT	aktueller Eingangswert
XH	DINT	obere Grenze des Eingangswertebereich
XL	DINT	untere Grenze des Eingangswertebereich
YH	DINT	obere Grenze des Ausgangswertebereich
YL	DINT	untere Grenze des Ausgangswertebereich

Parameter der Ausgänge

3309

Parameter	Datentyp	Beschreibung
Y	DINT	normierter Wert
X_OUT_OF_RANGE	BOOL	Eingangswert X ist außerhalb des definierten Wertebereichs XL/XH

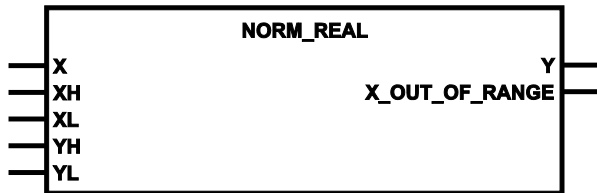
NORM_REAL

13244

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_PDMsmart_UTIL_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

3310

NORM_REAL normiert einen Wert innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen.

Der FB normiert einen Wert vom Typ REAL, der innerhalb der Grenzen XH und XL liegt, auf einen Ausgangswert innerhalb der Grenzen YH und YL. Der FB wird z.B. bei der Erzeugung von PWM-Werten aus analogen Eingangsgrößen genutzt.

! HINWEIS

- ▶ Der Eingangswert für X muss sich im definierten Bereich zwischen XL und XH befinden! Der FB prüft NICHT den Wert X auf Plausibilität. Außerhalb dieses Wertebereiches wird der Ausgang X_OUT_OF_RANGE gesetzt.
- ▶ Das Ergebnis der Berechnung $(XH-XL) \cdot (YH-YL)$ muss im Wertebereich des Datentyps REAL ($-3,402823466 \cdot 10^{38} \dots 3,402823466 \cdot 10^{38}$) bleiben!
- > Bedingt durch die Rundungsfehler können Abweichungen beim normierten Wert um 1 auftreten.
- > Werden die Grenzen (XH/XL oder YH/YL) invertiert angegeben, erfolgt auch die Normierung invertiert.

Parameter der Eingänge

3311

Parameter	Datentyp	Beschreibung
X	REAL	aktueller Eingangswert
XH	REAL	obere Grenze des Eingangswertebereich
XL	REAL	untere Grenze des Eingangswertebereich
YH	REAL	obere Grenze des Ausgangswertebereich
YL	REAL	untere Grenze des Ausgangswertebereich

Parameter der Ausgänge

3312

Parameter	Datentyp	Beschreibung
Y	REAL	normierter Wert
X_OUT_OF_RANGE	BOOL	Eingangswert X ist außerhalb des definierten Wertebereichs XL/XH

TOGGLE

13248

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek ifm_PDMsmart_UTIL_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

3304

TOGGLE ermöglicht das Setzen und Rücksetzen einer boolschen Variablen mit nur einem Eingangs-Bit.

Die erste steigende Flanke am Eingang IN setzt den Ausgang OUT auf 'TRUE'.

Die nächste steigende Flanke setzt den Ausgang wieder zurück auf 'FALSE'.

usw.

Parameter der Eingänge

3305

Parameter	Datentyp	Beschreibung
IN	BOOL	Flanke FALSE ⇒ TRUE: Setzen / Rücksetzen des Ausgangs

Parameter der Ausgänge

3306

Parameter	Datentyp	Beschreibung
OUT	BOOL	1. Flanke an IN ⇒ TRUE 2. Flanke an IN ⇒ FALSE 3. Flanke an IN ⇒ TRUE ...

Bausteine: Grafik Visualisierung

Inhalt

BASICDISPLAY_INIT	181
PDM_PAGECONTROL	182

15298

Hier finden Sie **ifm**-Bausteine für folgende Zwecke:

- Gerätebildschirm initialisieren
- Visualisierungsseiten aufrufen



BASICDISPLAY_INIT

9310

Baustein-Typ = Funktion (FUN) vom Typ BOOL

Baustein ist enthalten in Bibliothek ifm_CRRnnnn_Init_Vxxyzz.LIB

Symbol in CODESYS:



Beschreibung

9312

Die Funktion BasicDisplay_Init initialisiert im ersten SPS-Zyklus den Bildschirm des BasicDisplays. Ohne diese Initialisierung bleibt der Bildschirm dunkel.

Die Funktion erfordert keinerlei Parametrierung.

- Die Funktion nur im ersten SPS-Zyklus aufrufen!
Danach den Aufruf überspringen.
→ folgendes Beispiel

Beispiel: BasicDisplay_Init

9314

► Das Programm (PRG) INIT_DISPLAY erzeugen. ► Die Funktion (FUN) BASICDISPLAY_INIT so aufrufen, dass sie nur im ersten SPS-Zyklus durchlaufen wird.	<pre> 0001 PROGRAM INIT_DISPLAY 0002 VAR 0003 DisplayInit: BOOL := TRUE; 0004 END_VAR 0005 0001 DisplayInit → SKIP 0002 BasicDisplay_Init 0003 SKIP FALSE → DisplayInit </pre>
► Im Programm PLC_PRG das Programm INIT_DISPLAY aufrufen (wenn vorhanden, zusammen mit den anderen Initialisierungen).	<pre> 0001 PROGRAM PLC_PRG 0002 VAR 0003 0004 0005 END_VAR 0006 0007 0001 INIT_DISPLAY </pre>

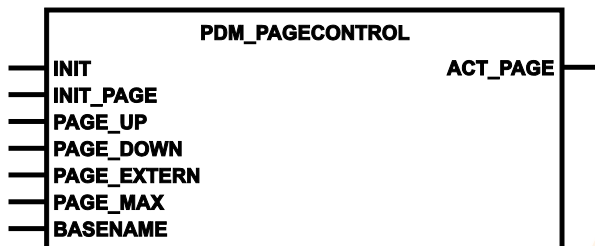
PDM_PAGECONTROL

3186

Baustein-Typ = Programm (PRG)

Baustein ist enthalten in Bibliothek	für folgende Geräte verfügbar:
ifm_PDM_UTIL_Vxxyyzz.LIB	- PDM360: CR1050, CR1051 - PDM360compact: CR1052, CR1053, CR1055, CR1056
ifm_PDMng_UTIL_Vxxyyzz.LIB	- PDM360 NG: CR108n
ifm_PDMsmart_UTIL_Vxxyyzz.LIB	- BasicDisplay: CR045n - PDM360smart: CR1070, CR1071

Symbol in CODESYS:



Beschreibung

3294

PDM_PAGECONTROL steuert den Aufruf bestimmter Visualisierungsseiten. Der Aufruf und die Rückgabe der Visualisierungsseiten erfolgt in CODESYS über die Systemvariable CurrentVisu (vom Typ STRING[40]).

Mit dem Programm kann wahlweise eine bestimmte Visualisierungsseite aufgerufen oder schrittweise in den Visualisierungen geblättert werden.

Das Programm lässt sich optimal nutzen, wenn die Namen aller Visualisierungen dem gleichen Schema entsprechen, also einer Kombination aus einem Basisnamen, gefolgt von einer 5-stelligen Zahl (ab Bibliotheks-Version V04.00.07; davor: 3-stellig *).

Beispiel BASENAME = PAGE:

Visualisierungsname = PAGE00001, PAGE00002, PAGE00003, usw.

Für den Basisnamen sind 1...35 Großbuchstaben (keine Sonderzeichen) zulässig. Die Nummerierung der Visualisierungen sollte lückenlos erfolgen. Das Programm setzt den endgültigen Visualisierungsnamen aus dem Parameter BASENAME und der Nummer zusammen oder liest die Nummer aus dem aktuellen Visualisierungsnamen aus und stellt sie im Ausgangsparameter ACT_PAGE zur Verfügung.

Anstatt die Visualisierungen mit Basisnamen und laufender Nummer zu benennen, kann jede Visualisierung auch individuell benannt werden, z.B.: SERVICE1, MOTORDATA2, CONFIGURATION3. Die Programmierung ist in diesem Fall aber aufwendiger, weil Basisname und Visualisierungsnummer einzeln zugewiesen werden müssen. Ein schrittweises Blättern ist nur noch sehr eingeschränkt möglich.

Verwenden Sie als BASENAME den Buchstaben P, dann ist Ihr Programm kompatibel mit den **ifm**-Templates.

***) Beachten Sie die neue 5-stellige Nummerierung auch bei der Namensgebung Ihrer bereits bestehenden Visualisierungsseiten!**

Parameter der Eingänge

3293

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Display wird initialisiert mit der in INIT_PAGE angegebenen Initialisierung FALSE: im weiteren Programmablauf
INIT_PAGE	WORD	Visualisierungsnummer, die mit INIT aufgerufen werden soll
PAGE_UP	BOOL	Flanke FALSE ⇒ TRUE: inkrementiert die Visualisierungsnummer
PAGE_DOWN	BOOL	Flanke FALSE ⇒ TRUE: dekrementiert die Visualisierungsnummer
PAGE_EXTERN	WORD	Angegebene Visualisierungsseite wird direkt aufgerufen (unabhängig von PAGE_UP / PAGE_DOWN) Sobald PAGE_EXTERN = ACT_PAGE, dann PAGE_EXTERN wieder auf "0" setzen!
PAGE_MAX	WORD	Maximale Anzahl der anwählbaren Visualisierungsseiten
BASENAME	STRING[35]	Gemeinsamer Namensbestandteil der Visualisierungsseite. Die Nummerierung der Visualisierungsseiten erfolgt durch die Namensgebung, z.B. "P00001". Hierbei gelten: - "P" = BASENAME (nur Großbuchstaben!) - "00001" = Visualisierungsnummer (5-stellig!)

Parameter der Ausgänge

3295

Parameter	Datentyp	Beschreibung
ACT_PAGE	WORD	aktuelle Visualisierungsnummer

6 Diagnose und Fehlerbehandlung

Inhalt

Diagnose	184
Fehler	184
Reaktion auf Fehlermeldungen	185
CAN / CANopen: Fehler und Fehlerbehandlung	185

19598

Das Laufzeitsystem (LZS) überprüft das Gerät durch interne Fehler-Checks:

- in der Startphase (Reset-Phase)
- während der Ausführung des Anwendungsprogramms

→ Kapitel **Betriebszustände** (→ Seite [25](#))

So wird eine möglichst hohe Betriebssicherheit gewährleistet.

6.1 Diagnose

19601

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche → Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:

- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.

Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.

Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

6.2 Fehler

19602

Ein Fehler ist die Unfähigkeit einer Einheit, eine geforderte Funktion auszuführen.

Kein Fehler ist diese Unfähigkeit während vorbeugender Wartung oder anderer geplanter Handlungen oder aufgrund des Fehlers externer Mittel.

Ein Fehler ist oft das Resultat eines Ausfalls der Einheit selbst, kann aber ohne vorherigen Ausfall bestehen.

In der ISO 13849-1 ist mit "Fehler" der "zufällige Fehler" gemeint.

6.3 Reaktion auf Fehlermeldungen

8504

Es liegt grundsätzlich in der Verantwortung des Programmierers, auf die Fehlermeldungen im Anwendungsprogramm zu reagieren. Über die Fehlermeldung erhält man eine Fehlerbeschreibung.

- > Das System setzt die Fehlermeldung zurück, sobald der fehlerauslösende Zustand nicht mehr vorliegt.

6.3.1 Beispielablauf für Reaktion auf Fehlermeldungen

8505

Das Laufzeitsystem schreibt zyklisch den Systemmarker TEMPERATURE.

Das Anwendungsprogramm erkennt die Gerätetemperatur durch Abfrage der INT-Variable.

Falls für die Anwendung zulässige Werte über- oder unterschritten werden:

- > Das Anwendungsprogramm schaltet die Ausgänge ab.
 - Die Ursache des Fehlers beheben.
- > Anwendungsprogramm erkennt den wieder normalen Temperaturwert:
Die Maschine / Anlage darf neu gestartet oder weiter betrieben werden.

6.4 CAN / CANopen: Fehler und Fehlerbehandlung

19604

→ Systemhandbuch "Hintergrundwissen ecomatmobile"

→ Kapitel **CAN / CANopen: Fehler und Fehlerbehandlung**

7 Anhang

Inhalt

Systemmerker	186
Fehler-Tabellen	188

1664

Hier stellen wir Ihnen – ergänzend zu den Angaben in den Datenblättern – zusammenfassende Tabellen zur Verfügung.

7.1 Systemmerker

8376

8440



Die zu den Systemmerkern gehörenden Merkeradressen können sich bei einer Erweiterung der Steuerungskonfiguration ändern.

- Für die Programmierung nur die Symbolnamen der Systemmerker nutzen!

Systemmerker (Symbolname)		Typ	Beschreibung
KEY_F1		BOOL	Funktionstaste 1
KEY_F2		BOOL	Funktionstaste 2
KEY_F3		BOOL	Funktionstaste 3
KEY_F4		BOOL	Funktionstaste 4
KEY_UP		BOOL	Navigationstaste [▲]
KEY_DOWN		BOOL	Navigationstaste [▼]
KEY_LEFT		BOOL	Navigationstaste [◀]
KEY_RIGHT		BOOL	Navigationstaste [▶]
KEY_OK		BOOL	Navigationstaste [OK]
KEY_CHANGED	BOOL	r	Impuls: Tastenstatus hat sich geändert (= eine Taste wurde gedrückt oder losgelassen)
KEY_PRESSED	BOOL	r	eine Taste wurde gedrückt
RT_F1		BOOL	Impuls: Funktionstaste 1 betätigt
RT_F2		BOOL	Impuls: Funktionstaste 2 betätigt
RT_F3		BOOL	Impuls: Funktionstaste 3 betätigt
RT_F4		BOOL	Impuls: Funktionstaste 4 betätigt
RT_UP		BOOL	Impuls: Navigationstaste [▲] betätigt
RT_DOWN		BOOL	Impuls: Navigationstaste [▼] betätigt
RT_LEFT		BOOL	Impuls: Navigationstaste [◀] betätigt
RT_RIGHT		BOOL	Impuls: Navigationstaste [▶] betätigt
RT_OK		BOOL	Impuls: Navigationstaste [OK] betätigt
FT_F1		BOOL	Impuls: Funktionstaste 1 losgelassen
FT_F2		BOOL	Impuls: Funktionstaste 2 losgelassen
FT_F3		BOOL	Impuls: Funktionstaste 3 losgelassen
FT_F4		BOOL	Impuls: Funktionstaste 4 losgelassen
FT_UP		BOOL	Impuls: Navigationstaste [▲] losgelassen
FT_DOWN		BOOL	Impuls: Navigationstaste [▼] losgelassen
FT_LEFT		BOOL	Impuls: Navigationstaste [◀] losgelassen

Systemmerker (Symbolname)	Typ	Beschreibung
FT_RIGHT	BOOL	Impuls: Navigationstaste [►] losgelassen
FT_OK	BOOL	Impuls: Navigationstaste [OK] losgelassen
KEY_BACKLIGHT_CTRL	BYTE	Tastenhinterleuchtung (0...100 %) voreingestellt = 100 %
SCREEN_BACKLIGHT_CTRL	BYTE	LCD-Hinterleuchtung (0...100 %) voreingestellt = 100 %
SUPPLY_VOLTAGE_VBBS	WORD	Versorgungsspannung an VBBS in [mV]
SUPPLY_VOLTAGE_VU	WORD	interne Versorgungsspannung in [mV]
TEMPERATURE	INT	Temperatur im Gerät [°C]



7.2 Fehler-Tabellen

Inhalt	
Fehlermerker	188
Fehler: CAN / CANopen	188

19606

7.2.1 Fehlermerker

19608

→ Kapitel **Systemmerker** (→ Seite [186](#))

7.2.2 Fehler: CAN / CANopen

19610
19604

→ Systemhandbuch "Hintergrundwissen ecomatmobile"
→ Kapitel **CAN / CANopen: Fehler und Fehlerbehandlung**

EMCY-Codes: CANx

13094

 Die Angaben für CANx gelten für jede der CAN-Schnittstellen.

EMCY-Code Objekt 0x1003		Objekt 0x1001	herstellerspezifische Informationen					Beschreibung
Byte 0 [hex]	Byte 1 [hex]	Byte 2 [hex]	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
00	80	11	---	---	---	---	---	CANx Monitoring SYNC-Error (nur Slave)
00	81	11	---	---	---	---	---	CANx Warngrenze (> 96)
10	81	11	---	---	---	---	---	CANx Empfangspuffer Überlauf
11	81	11	---	---	---	---	---	CANx Sendepuffer Überlauf
30	81	11	---	---	---	---	---	CANx Guard-/Heartbeat-Error (nur Slave)

EMCY-Codes: System

8413

EMCY-Code Objekt 0x1003		Objekt 0x1001	herstellerspezifische Informationen					Beschreibung
Byte 0 [hex]	Byte 1 [hex]	Byte 2 [hex]	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
00	31	05						Versorgungsspannung
00	42	09						Übertemperatur

8 Begriffe und Abkürzungen

A

Adresse

Das ist der „Name“ des Teilnehmers im Bus. Alle Teilnehmer benötigen eine unverwechselbare, eindeutige Adresse, damit der Austausch der Signale fehlerfrei funktioniert.

Anleitung

Übergeordnetes Wort für einen der folgenden Begriffe:

Montageanleitung, Datenblatt, Benutzerinformation, Bedienungsanleitung, Gerätehandbuch, Installationsanleitung, Onlinehilfe, Systemhandbuch, Programmierhandbuch, usw.

Anwendungsprogramm

Software, die speziell für die Anwendung vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

Architektur

Spezifische Konfiguration von Hardware- und/oder Software-Elementen in einem System.

B

Baud

Baud, Abk.: Bd = Maßeinheit für die Geschwindigkeit bei der Datenübertragung. Baud ist nicht zu verwechseln mit "bits per second" (bps, Bit/s). Baud gibt zwar die Anzahl von Zustandsänderungen (Schritte, Takte) pro Sekunde auf einer Übertragungsstrecke an. Aber es ist nicht festgelegt, wie viele Bits pro Schritt übertragen werden. Der Name Baud geht auf den französischen Erfinder J. M. Baudot zurück, dessen Code für Telexgeräte verwendet wurde.

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

Bestimmungsgemäße Verwendung

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

Bootloader

Im Auslieferungszustand enthalten **ecomatmobile**-Controller nur den Bootloader.

Der Bootloader ist ein Startprogramm, mit dem das Laufzeitsystem und das Anwendungsprogramm auf dem Gerät nachgeladen werden können.

Der Bootloader enthält Grundroutinen...

- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Laufzeitsystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Bus

Serielle Datenübertragung mehrerer Teilnehmer an derselben Leitung.

C

CAN

CAN = **C**ontroller **A**rea **N**etwork

CAN gilt als Feldbussystem für größere Datenmengen, das prioritätengesteuert arbeitet. Es gibt mehrere höhere Protokolle, die auf CAN aufsetzen, z. B. 'CANopen' oder 'J1939'.

CAN-Stack

CAN-Stack = Software-Komponente, die sich um die Verarbeitung von CAN-Telegramme kümmert.

CiA

CiA = CAN in Automation e.V.

Anwender- und Herstellerorganisation in Erlangen, Deutschland. Definitions- und Kontrollorgan für das CANopen-Protokoll.

Homepage → www.can-cia.org

CiA DS 304

DS = **D**raft **S**tandard

CANopen-Geräteprofil für sichere Kommunikation

CiA DS 401

DS = **D**raft **S**tandard

CANopen-Geräteprofil für digitale und analoge E/A-Baugruppen

CiA DS 402

DS = **D**raft **S**tandard

CANopen-Geräteprofil für Antriebe

CiA DS 403

DS = **D**raft **S**tandard

CANopen-Geräteprofil für Bediengeräte

CiA DS 404

DS = **D**raft **S**tandard

CANopen-Geräteprofil für Messtechnik und Regler

CiA DS 405

DS = **D**raft **S**tandard

CANopen-Spezifikation der Schnittstelle zu programmierbaren Steuerungen (IEC 61131-3)

CiA DS 406

DS = **D**raft **S**tandard

CANopen-Geräteprofil für Drehgeber / Encoder

CiA DS 407

DS = **D**raft **S**tandard

CANopen-Anwendungsprofil für den öffentlichen Nahverkehr

COB-ID

COB = **C**ommunication **O**bject = Kommunikationsobjekt

ID = **I**dentifier = Kennung

ID eines CANopen-Kommunikationsobjekts

Entspricht dem Identifier der CAN-Nachricht, mit der das Kommunikationsobjekt über den CAN-Bus gesendet wird.

CODESYS

CODESYS® ist eingetragene Marke der 3S – Smart Software Solutions GmbH, Deutschland. 'CODESYS for Automation Alliance™' vereinigt Firmen der Automatisierungsindustrie, deren Hardware-Geräte alle mit dem weit verbreiteten IEC 61131-3 Entwicklungswerkzeug CODESYS® programmiert werden.

Homepage → www.codesys.com

CSV-Datei

CSV = **C**omma **S**eparated **V**alues (auch: **C**haracter **S**eparated **V**alues)

Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten.

Die Dateinamen-Erweiterung lautet .csv.

Beispiel: Quell-Tabelle mit Zahlenwerten:

Wert 1.0	Wert 1.1	Wert 1.2	Wert 1.3
Wert 2.0	Wert 2.1	Wert 2.2	Wert 2.3
Wert 3.0	Wert 3.1	Wert 3.2	Wert 3.3

Daraus entsteht folgende CSV-Datei:

```
Wert 1.0;Wert 1.1;Wert 1.2;Wert 1.3
Wert 2.0;Wert 2.1;Wert 2.2;Wert 2.3
Wert 3.0;Wert 3.1;Wert 3.2;Wert 3.3
```

D

Datentyp

Abhängig vom Datentyp können unterschiedlich große Werte gespeichert werden.

Datentyp	min. Wert	max. Wert	Größe im Speicher
BOOL	FALSE	TRUE	8 Bit = 1 Byte
BYTE	0	255	8 Bit = 1 Byte
WORD	0	65 535	16 Bit = 2 Bytes
DWORD	0	4 294 967 295	32 Bit = 4 Bytes
SINT	-128	127	8 Bit = 1 Byte
USINT	0	255	8 Bit = 1 Byte
INT	-32 768	32 767	16 Bit = 2 Bytes
UINT	0	65 535	16 Bit = 2 Bytes
DINT	-2 147 483 648	2 147 483 647	32 Bit = 4 Bytes
UDINT	0	4 294 967 295	32 Bit = 4 Bytes
REAL	-3,402823466 • 10 ³⁸	3,402823466 • 10 ³⁸	32 Bit = 4 Bytes
ULINT	0	18 446 744 073 709 551 615	64 Bit = 8 Bytes
STRING			number of char. + 1

DC

Direct Current = Gleichstrom

Diagnose

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche →Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:

- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.

Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.

Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

Dither

to dither (engl.) = schwanken / zittern.

Dither ist ein Bestandteil der →PWM-Signale zum Ansteuern von Hydraulik-Ventilen. Für die elektromagnetischen Antriebe von Hydraulik-Ventilen hat sich herausgestellt, dass sich die Ventile viel besser regeln lassen, wenn das Steuersignal (PWM-Impulse) mit einer bestimmten Frequenz der PWM-Frequenz überlagert wird. Diese Dither-Frequenz muss ein ganzzahliger Teil der PWM-Frequenz sein.

DLC

Data Length Code = bei CANopen die Anzahl der Daten-Bytes in einer Nachricht.

Für →SDO: DLC = 8

DRAM

DRAM = **D**ynamic **R**andom **A**ccess **M**emory.

Technologie für einen elektronischen Speicherbaustein mit wahlfreiem Zugriff (Random Access Memory, RAM). Das speichernde Element ist dabei ein Kondensator, der entweder geladen oder entladen ist. Über einen Schalttransistor wird er zugänglich und entweder ausgelesen oder mit neuem Inhalt beschrieben. Der Speicherinhalt ist flüchtig: die gespeicherte Information geht bei fehlender Betriebsspannung oder zu später Wiederauffrischung verloren.

DTC

DTC = **D**iagnostic **T**rouble **C**ode = Fehler-Code

Beim Protokoll J1939 werden Störungen und Fehler über zugeordnete Nummern – den DTCs – verwaltet und gemeldet.

E

ECU

(1) **E**lectronic **C**ontrol **U**nit = Steuergerät oder Mikrocontroller

(2) **E**ngine **C**ontrol **U**nit = Steuergerät eines Motors

EDS-Datei

EDS = **E**lectronic **D**ata **S**heet = elektronisch hinterlegtes Datenblatt, z.B. für:

- Datei für das Objektverzeichnis im CANopen-Master,
- CANopen-Gerätebeschreibungen.

Via EDS können vereinfacht Geräte und Programme ihre Spezifikationen austauschen und gegenseitig berücksichtigen.

Embedded Software

System-Software, Grundprogramm im Gerät, praktisch das →Laufzeitsystem.

Die Firmware stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm. Die Firmware wird vom Hersteller der Steuerung als Teil des Systems geliefert und kann vom Anwender nicht verändert werden.

EMCY

Abkürzung für Emergency (engl.) = Notfall

Nachricht im CANopen-Protokoll, mit der Fehler gemeldet werden.

EMV

EMV = **E**lektro-**M**agnetische **V**erträglichkeit.

Gemäß der EG-Richtlinie (2004/108/EG) zur elektromagnetischen Verträglichkeit (kurz EMV-Richtlinie) werden Anforderungen an die Fähigkeit von elektrischen und elektronischen Apparaten, Anlagen, Systemen oder Bauteilen gestellt, in der vorhandenen elektromagnetischen Umwelt zufriedenstellend zu arbeiten. Die Geräte dürfen ihre Umgebung nicht stören und dürfen sich von äußerlichen elektromagnetischen Störungen nicht ungünstig beeinflussen lassen.

Ethernet

Ethernet ist eine weit verbreitete, herstellernerneutrale Netzwerktechnologie, mit der Daten mit einer Geschwindigkeit von 10 bis 10 000 Millionen Bit pro Sekunde (Mbps) übertragen werden können. Ethernet gehört zu der Familie der sogenannten „bestmöglichen Datenübermittlung“ auf einem nicht exklusiven Übertragungsmedium. 1972 entwickelt, wurde das Konzept 1985 als IEEE 802.3 spezifiziert.

EUC

EUC = **E**quipment **U**nder **C**ontrol (kontrollierte Einrichtung).

EUC ist eine Einrichtung, Maschine, Gerät oder Anlage, verwendet zur Fertigung, Stoffumformung, zum Transport, zu medizinischen oder anderen Tätigkeiten (→ IEC 61508-4, Abschnitt 3.2.3). Das EUC umfasst also alle Einrichtungen, Maschinen, Geräte oder Anlagen, die →Gefährdungen verursachen können und für die sicherheitsgerichtete Systeme erforderlich sind.

Falls eine vernünftigerweise vorhersehbare Aktivität oder Inaktivität zu durch das EUC verursachten Gefährdungen mit unvertretbarem Risiko führt, sind Sicherheitsfunktionen erforderlich, um einen sicheren Zustand für das EUC zu erreichen oder aufrecht zu erhalten. Diese Sicherheitsfunktionen werden durch ein oder mehrere sicherheitsgerichtete Systeme ausgeführt.

F

Fehlanwendung

Das ist die Verwendung eines Produkts in einer Weise, die vom Konstrukteur nicht vorgesehen ist. Eine Fehlanwendung führt meist zu einer →Gefährdung von Personen oder Sachen.

Vor vernünftigerweise, vorhersehbaren Fehlanwendungen muss der Hersteller des Produkts in seinen Benutzerinformationen warnen.

FiFo

FIFO (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Flash-Speicher

Flash-ROM (oder Flash-EPROM oder Flash-Memory) kombiniert die Vorteile von Halbleiterspeicher und Festplatten. Die Daten werden allerdings wie bei einer Festplatte blockweise in Datenblöcken zu 64, 128, 256, 1024, ... Byte zugleich geschrieben und gelöscht.

Vorteile von Flash-Speicher

- Die gespeicherten Daten bleiben auch bei fehlender Versorgungsspannung erhalten.
- Wegen fehlender beweglicher Teile ist Flash geräuschlos, unempfindlich gegen Erschütterungen und magnetische Felder.

Nachteile von Flash-Speicher

- Begrenzte Zahl von Schreib- bzw. Löschvorgängen, die eine Speicherzelle vertragen kann:
 - Multi-Level-Cells: typ. 10 000 Zyklen
 - Single-Level-Cells: typ. 100 000 Zyklen
- Da ein Schreibvorgang Speicherblöcke zwischen 16 und 128 kByte gleichzeitig beschreibt, werden auch Speicherzellen beansprucht, die gar keiner Veränderung bedürfen.

FRAM

FRAM, oder auch FeRAM, bedeutet **F**erroelectric **R**andom **A**ccess **M**emory. Der Speicher- und Löschvorgang erfolgt durch eine Polarisationsänderung in einer ferroelektrischen Schicht.

Vorteile von FRAM gegenüber herkömmlichen Festwertspeichern:

- nicht flüchtig,
- kompatibel zu gängigen EEPROMs, jedoch:
- Zugriffszeit ca. 100 ns,
- fast unbegrenzt viele Zugriffszyklen möglich.

H

Heartbeat

Heartbeat (engl.) = Herzschlag.

Die Teilnehmer senden regelmäßig kurze Signale. So können die anderen Teilnehmer prüfen, ob ein Teilnehmer ausgefallen ist.

HMI

HMI = **H**uman **M**achine **I**nterface = Mensch-Maschine-Schnittstelle

I

ID – Identifier

ID = **I**dentifier = Kennung

Name zur Unterscheidung der an einem System angeschlossenen Geräte / Teilnehmer oder der zwischen den Teilnehmern ausgetauschten Nachrichtenpakete.

IEC 61131

Norm: Grundlagen Speicherprogrammierbarer Steuerungen

- Teil 1: Allgemeine Informationen
- Teil 2: Betriebsmittelanforderungen und Prüfungen
- Teil 3: Programmiersprachen
- Teil 5: Kommunikation
- Teil 7: Fuzzy-Control-Programmierung

IEC-User-Zyklus

IEC-User-Zyklus = SPS-Zyklus im CODESYS-Anwendungsprogramm.

IP-Adresse

IP = Internet **P**rotocol = Internet-Protokoll.

Die IP-Adresse ist eine Nummer, die zur eindeutigen Identifizierung eines Internet-Teilnehmers notwendig ist. Zur besseren Übersicht wird die Nummer in 4 dezimalen Werten geschrieben, z. B. 127.215.205.156.

ISO 11898

Norm: Straßenfahrzeuge – CAN-Protokoll

- Teil 1: Bit-Übertragungsschicht und physikalische Zeichenabgabe
- Teil 2: High-speed medium access unit
- Teil 3: Fehlertolerante Schnittstelle für niedrige Geschwindigkeiten
- Teil 4: Zeitgesteuerte Kommunikation
- Teil 5: High-speed medium access unit with low-power mode

ISO 11992

Norm: Straßenfahrzeuge – Austausch von digitalen Informationen über elektrische Verbindungen zwischen Zugfahrzeugen und Anhängfahrzeugen

- Teil 1: Bit-Übertragungsschicht und Sicherungsschicht
- Teil 2: Anwendungsschicht für die Bremsausrüstung
- Teil 3: Anwendungsschicht für andere als die Bremsausrüstung
- Teil 4: Diagnose

ISO 16845

Norm: Straßenfahrzeuge – Steuergerätenetz (CAN) – Prüfplan zu Konformität

J

J1939

→ SAE J1939

K

Klemme 15

Klemme 15 ist in Fahrzeugen die vom Zündschloss geschaltete Plusleitung.

L

Laufzeitsystem

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm.

→ Kapitel **Software-Module für das Gerät** (→ Seite [18](#))

LED

LED = **L**ight **E**mitting **D**iode = Licht aussendende Diode.

Leuchtdiode, auch Luminiszenzdiode, ein elektronisches Element mit hoher, farbiger Leuchtkraft auf kleinem Volumen bei vernachlässigbarer Verlustleistung.

Link

Ein Link ist ein Querverweis zu einer anderen Stelle im Dokument oder auf ein externes Dokument.

LSB

Least **S**ignificant **B**it/Byte = Niederwertigstes Bit/Byte in einer Reihe von Bit/Bytes.

M

MAC-ID

MAC = **M**anufacturer's **A**ddress **C**ode

= Hersteller-Seriennummer.

→ ID = **I**dentifier = Kennung

Jede Netzwerkkarte verfügt über eine so genannte MAC-Adresse, ein unverwechselbarer, auf der ganzen Welt einzigartiger Zahlencode – quasi eine Art Seriennummer. So eine MAC-Adresse ist eine Aneinanderreihung von 6 Hexadezimalzahlen, etwa "00-0C-6E-D0-02-3F".

Master

Wickelt die komplette Organisation auf dem →Bus ab. Der Master entscheidet über den zeitlichen Buszugriff und fragt die →Slaves zyklisch ab.

MMI

MMI = **M**ensch-**M**aschine-**I**nterface

→ **HMI** (→ Seite [194](#))

MRAM

MRAM = **M**agnetoresistive **R**andom **A**ccess **M**emory

Die Informationen werden mit magnetischen Ladungselementen gespeichert. Dabei wird die Eigenschaft bestimmter Materialien ausgenutzt, die ihren elektrischen Widerstand unter dem Einfluss magnetischer Felder ändern.

Vorteile von MRAM gegenüber herkömmlichen Festwertspeichern:

- nicht flüchtig (wie FRAM), jedoch:
- Zugriffszeit nur ca. 35 ns,
- unbegrenzt viele Zugriffszyklen möglich.

MSB

Most **S**ignificant **B**it/Byte = Höchstwertiges Bit/Byte einer Reihe von Bits/Bytes.

N

NMT

NMT = **N**etwork **M**anagement = Netzwerk-Verwaltung (hier: im CANopen-Protokoll).
Der NMT-Master steuert die Betriebszustände der NMT-Slaves.

Node

Node (engl.) = Knoten. Damit ist ein Teilnehmer im Netzwerk gemeint.

Node Guarding

Node (engl.) = Knoten, hier: Netzwerkteilnehmer

Guarding (engl.) = Schutz

Parametrierbare, zyklische Überwachung von jedem entsprechend konfigurierten →Slave. Der →Master prüft, ob die Slaves rechtzeitig antworten. Die Slaves prüfen, ob der Master regelmäßig anfragt. Somit können ausgefallene Netzwerkteilnehmer schnell erkannt und gemeldet werden.

O

Obj / Objekt

Oberbegriff für austauschbare Daten / Botschaften innerhalb des CANopen-Netzwerks.

Objektverzeichnis

Das **Objektverzeichnis** OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OBV

Das **Objektverzeichnis** OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OPC

OPC = **O**LE for **P**rocess **C**ontrol = Objektverknüpfung und -einbettung für Prozesssteuerung
Standardisierte Software-Schnittstelle zur herstellerunabhängigen Kommunikation in der Automatisierungstechnik

OPC-Client (z.B. Gerät zum Parametrieren oder Programmieren) meldet sich nach dem Anschließen am OPC-Server (z.B. Automatisierungsgerät) automatisch bei diesem an und kommuniziert mit ihm.

operational

Operational (engl.) = betriebsbereit

Betriebszustand eines CANopen-Teilnehmers. In diesem Modus können →SDOs, →NMT-Kommandos und →PDOs übertragen werden.

P

PC-Karte

→ PCMCIA-Karte

PCMCIA-Karte

PCMCIA = Personal Computer Memory Card International Association, ein Standard für Erweiterungskarten mobiler Computer.

Seit der Einführung des Cardbus-Standards 1995 werden PCMCIA-Karten auch als PC-Karte (engl.: PC Card) bezeichnet.

PDM

PDM = **P**rocess and **D**ialog **M**odule = **P**rozess- und **D**ialog-**M**onitor.

Gerät zur Kommunikation des Bedieners mit der Maschine / Anlage.

PDO

PDO = **P**rocess **D**ata **O**bject = Nachrichten-Objekt mit Prozessdaten.

Die zeitkritischen Prozessdaten werden mit Hilfe der "Process Data Objects" (PDOs) übertragen. Die PDOs können beliebig zwischen den einzelnen Knoten ausgetauscht werden (PDO-Linking).

Zusätzlich wird festgelegt, ob der Datenaustausch ereignisgesteuert (asynchron) oder synchronisiert erfolgen soll. Je nach der Art der zu übertragenden Daten kann die richtige Wahl der Übertragungsart zu einer erheblichen Entlastung des →CAN-Bus führen.

Dem Protokoll entsprechend, sind diese Dienste nicht bestätigte Dienste: es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

PDU

PDU = **P**rotocol **D**ata **U**nit = Protokoll-Daten-Einheit.

Die PDU ist ein Begriff aus dem →CAN-Protokoll →SAE J1939. Sie bezeichnet einen Bestandteil der Ziel- oder Quelladresse.

PES

Programable **e**lectronic **s**ystem = Programmierbares elektronisches System ...

- zur Steuerung, zum Schutz oder zur Überwachung,
- auf der Basis einer oder mehrerer programmierbarer Geräte,
- einschließlich aller Elemente dieses Systems, wie Ein- und Ausgabegeräte.

PGN

PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer

PGN = PDU Format (PF) + PDU Source (PS)

Die Parameter-Gruppennummer ist ein Begriff aus dem →CAN-Protokoll →SAE J1939. Sie fasst die Teiladressen PF und PS zusammen.

PID-Regler

Der PID-Regler (proportional–integral–derivative controller) besteht aus folgenden Anteilen:

- P = Proportional-Anteil
- I = Integral-Anteil
- D = Differential-Anteil (jedoch nicht beim Controller CR04nn, CR253n).

Piktogramm

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln (→ Kapitel **Was bedeuten die Symbole und Formatierungen?** (→ Seite [6](#))).

Pre-Op

Pre-Op = PRE-OPERATIONAL mode (engl.) = Zustand vor 'betriebsbereit'.

Betriebszustand eines CANopen-Teilnehmers. Nach dem Einschalten der Versorgungsspannung geht jeder Teilnehmer automatisch in diesem Zustand. Im CANopen-Netz können in diesem Modus nur →SDOs und →NMT-Kommandos übertragen werden, jedoch keine Prozessdaten.

Prozessabbild

Mit Prozessabbild bezeichnet man den Zustand der Ein- und Ausgänge, mit denen die SPS innerhalb eines →Zyklus arbeitet.

- Am Zyklus-Beginn liest die SPS die Zustände aller Eingänge in das Prozessabbild ein. Während des Zyklus kann die SPS Änderungen an den Eingängen nicht erkennen.
- Im Laufe des Zyklus werden die Ausgänge nur virtuell (im Prozessabbild) geändert.
- Am Zyklus-Ende schreibt die SPS die virtuellen Ausgangszustände auf die realen Ausgänge.

PWM

PWM = Puls-Weiten-Modulation

Bei dem PWM-Ausgangssignal handelt es sich um ein getaktetes Signal zwischen GND und Versorgungsspannung.

Innerhalb einer festen Periode (PWM-Frequenz) wird das Puls-/Pausenverhältnis variiert. Durch die angeschlossene Last stellt sich je nach Puls-/Pausenverhältnis der entsprechende Effektivstrom ein.

R

ratiometrisch

Ratio (lat.) = Verhältnis

Messungen können auch ratiometrisch erfolgen = Verhältnismessung. Wenn das Ausgangssignal eines Sensors proportional zu seiner Versorgungsspannung ist, kann durch ratiometrische Messung (= Messung im Verhältnis zur Versorgung) der Einfluss von Schwankungen der Versorgung reduziert, im Idealfall sogar beseitigt werden.

→ Analogeingang

RAW-CAN

RAW-CAN bezeichnet das reine →CAN-Protokoll, das ohne ein zusätzliches Kommunikationsprotokoll auf dem CAN-Bus (auf ISO/OSI-Schicht 2) arbeitet. Das CAN-Protokoll ist international nach →ISO 11898-1 definiert und garantiert zusätzlich in →ISO 16845 die Austauschbarkeit von CAN-Chips.

remanent

Remanente Daten sind gegen Datenverlust bei Spannungsausfall geschützt.

Z.B. kopiert das →Laufzeitsystem die remanenten Daten automatisch in einen →Flash-Speicher, sobald die Spannungsversorgung unter einen kritischen Wert sinkt. Bei Wiederkehr der Spannungsversorgung lädt das Laufzeitsystem die remanenten Daten zurück in den Arbeitsspeicher. Dagegen sind die Daten im Arbeitsspeicher einer Steuerung flüchtig und bei Unterbrechung der Spannungsversorgung normalerweise verloren.

ro

ro = read only (engl.) = nur lesen

Unidirektionale Datenübertragung: Daten können nur gelesen werden, jedoch nicht verändert.

RTC

RTC = **Real Time Clock** = Echtzeituhr

Liefert (batteriegepuffert) aktuell Datum und Uhrzeit. Häufiger Einsatz beim Speichern von Fehlermeldungsprotokollen.

rw

rw = read/write (engl.) = lesen und schreiben

Bidirektionale Datenübertragung: Daten können sowohl gelesen als auch verändert werden.

S

SAE J1939

Das Netzwerkprotokoll SAE J1939 beschreibt die Kommunikation auf einem →CAN-Bus in Nutzfahrzeugen zur Übermittlung von Diagnosedaten (z.B. Motordrehzahl, Temperatur) und Steuerungsinformationen.

Norm: Recommended Practice for a Serial Control and Communications Vehicle Network

- Teil 2: Agricultural and Forestry Off-Road Machinery Control and Communication Network
- Teil 3: On Board Diagnostics Implementation Guide
- Teil 5: Marine Stern Drive and Inboard Spark-Ignition Engine On-Board Diagnostics Implementation Guide

- Teil 11: Physical Layer – 250 kBits/s, Shielded Twisted Pair
- Teil 13: Off-Board Diagnostic Connector
- Teil 15: Reduced Physical Layer, 250 kBits/s, Un-Shielded Twisted Pair (UTP)
- Teil 21: Data Link Layer
- Teil 31: Network Layer
- Teil 71: Vehicle Application Layer
- Teil 73: Application Layer – Diagnostics
- Teil 81: Network Management Protocol

SD-Card

Eine SD Memory Card (Kurzform für **Secure Digital** Memory Card; deutsch: Sichere digitale Speicherkarte) ist ein digitales Speichermedium, das nach dem Prinzip der →Flash-Speicherung arbeitet.

SDO

SDO = **S**ervice **D**ata **O**bject = Nachrichten-Objekt mit Servicedaten.

Das SDO dient dem Zugriff auf Objekte in einem CANopen-Objektverzeichnis. Dabei fordern 'Clients' die gewünschten Daten von 'Servern' an. Die SDOs bestehen immer aus 8 Bytes.

Beispiele:

- Automatische Konfiguration aller →Slaves über SDOs beim Systemstart.
 - Auslesen der Fehlernachrichten aus dem →Objektverzeichnis.
- Jedes SDO wird auf Antwort überwacht und wiederholt, wenn sich innerhalb der Überwachungszeit der Slave nicht meldet.

Selbsttest

Testprogramm, das aktiv Komponenten oder Geräte testet. Das Programm wird durch den Anwender gestartet und dauert eine gewisse Zeit. Das Ergebnis davon ist ein Testprotokoll (Log-Datei), aus dem entnommen werden kann, was getestet wurde und ob das Ergebnis positiv oder negativ ist.

Slave

Passiver Teilnehmer am Bus, antwortet nur auf Anfrage des →Masters. Slaves haben im Bus eine eindeutige →Adresse.

Steuerungskonfiguration

Bestandteil der CODESYS-Bedienoberfläche.

- ▶ Programmierer teilt dem Programmiersystem mit, welche Hardware programmiert werden soll.
- > CODESYS lädt die zugehörigen Bibliotheken.
- > Lesen und schreiben der Peripherie-Zustände (Ein-/Ausgänge) ist möglich.

stopped

stopped (engl.) = angehalten

Betriebszustand eines CANopen-Teilnehmers. In diesem Modus werden nur →NMT-Kommandos übertragen.

Symbole

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln (→ Kapitel **Was bedeuten die Symbole und Formatierungen?** (→ Seite [6](#))).

Systemvariable

Variable, auf die via IEC-Adresse oder Symbolname aus der SPS zugegriffen werden kann.

T

Target

Das Target enthält für CODESYS die Hardware-Beschreibung des Zielgeräts, z.B.: Ein- und Ausgänge, Speicher, Dateiablageorte.

Entspricht einem elektronischen Datenblatt.

TCP

Das **T**ransmission **C**ontrol **P**rotocol ist Teil der Protokollfamilie TCP/IP. Jede TCP/IP-Datenverbindung hat einen Sender und einen Empfänger. Dieses Prinzip ist eine verbindungsorientierte Datenübertragung. In der TCP/IP-Protokollfamilie übernimmt TCP als verbindungsorientiertes Protokoll die Aufgabe der Datensicherheit, der Datenflusssteuerung und ergreift Maßnahmen bei einem Datenverlust. (vgl.: →UDP)

Template

Template (englisch = Schablone) ist eine Vorlage, die mit Inhalten gefüllt werden kann.

Hier: Eine Struktur von vorkonfigurierten Software-Elementen als Basis für ein Anwendungsprogramm.

U

UDP

UDP (**U**ser **D**atagram **P**rotocol) ist ein minimales, verbindungsloses Netzprotokoll, das zur Transportschicht der Internetprotokollfamilie gehört. Aufgabe von UDP ist es, Daten, die über das Internet übertragen werden, der richtigen Anwendung zukommen zu lassen.

Derzeit sind Netzwerkvariablen auf Basis von →CAN und UDP implementiert. Die Variablenwerte werden dabei auf der Basis von Broadcast-Nachrichten automatisch ausgetauscht. In UDP sind diese als Broadcast-Telegramme realisiert, in CAN als →PDOs.

Dem Protokoll entsprechend, sind diese Dienste nicht bestätigte Dienste: es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

V

Verwendung, bestimmungsgemäß

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

W

Watchdog

Der Begriff Watchdog (englisch; Wachhund) wird verallgemeinert für eine Komponente eines Systems verwendet, die die Funktion anderer Komponenten beobachtet. Wird dabei eine mögliche Fehlfunktion erkannt, so wird dies entweder signalisiert oder geeignete Programm-Verzweigungen eingeleitet. Das Signal oder die Verzweigungen dienen als Auslöser für andere kooperierende Systemkomponenten, die das Problem lösen sollen.

wo

wo = write only (engl.) = nur schreiben

Unidirektionale Datenübertragung: Daten können nur verändert werden, jedoch nicht gelesen.

Z

Zykluszeit

Das ist die Zeit für einen Zyklus. Das SPS-Programm läuft einmal komplett durch.

Je nach ereignisgesteuerten Verzweigungen im Programm kann dies unterschiedlich lange dauern.

9 Index

A

Adresse	189
Angaben zum Gerät	11
Anhang	186
Anlaufverhalten der Steuerung	10
Anleitung	189
Anschluss auf Gehäuserückwand	15
Anwendungsprogramm	19, 189
Anwendungsprogramm erstellen	23
Architektur	189

B

BASICDISPLAY_INIT	181
Baud	189
Baustein-Ausgänge	52
Bausteine	
CANopen	79
CANopen Emergency	118
CANopen Guarding	114
CANopen Netzwerkmanagement	88
CANopen Objektverzeichnis	92
CANopen SDOs	97
CANopen Status	79
CANopen SYNC	110
Grafik	174
Grafik Hilfe	174
Grafik Visualisierung	180
RAW-CAN (Layer 2)	53
RAW-CAN Daten empfangen	59
RAW-CAN Daten senden	69
RAW-CAN Remote	75
RAW-CAN Status	53
SAE J1939	124
SAE J1939 Diagnose	148
SAE J1939 Empfangen	135
SAE J1939 Request	132
SAE J1939 Senden	140
SAE J1939 Status	124
System	156
Beachten!	9
Bedien-Elemente des CR0451	14
Begrenzungen für Visualisierungen	30
Beispiel	
BasicDisplay_Init	181
Beispielablauf für Reaktion auf Fehlermeldungen	185
Bestimmungsgemäße Verwendung	189
Betriebsstundenzähler	167
Betriebszustände	25
Bewegung von Visualisierungselementen	33
Bibliothek ifm_CANopen_NT_Vxxyzz.LIB	49
Bibliothek ifm_CR0451_Init_Vxxyzz.LIB	47
Bibliothek ifm_CR0451_Vxxyzz.LIB	47
Bibliothek ifm_J1939_NT_Vxxyzz.LIB	50
Bibliothek ifm_PDMsmart_util_Vxxyzz.LIB	48
Bibliothek ifm_RAWCan_NT_Vxxyzz.LIB	48
Bibliotheken	20
für Netzwerkvariablen erforderlich	46
mindestens erforderlich	46

Bildskalierung	30
Bootloader	19, 189
Bus	189

C

CAN	190
Schnittstellen und Protokolle	17
CAN / CANopen	
Fehler und Fehlerbehandlung	185
CAN_ENABLE	54
CAN_RECOVER	55
CAN_REMOTE_REQUEST	76
CAN_REMOTE_RESPONSE	77
CAN_RX	60
CAN_RX_ENH	61
CAN_RX_ENH_FIFO	63
CAN_RX_RANGE	65
CAN_RX_RANGE_FIFO	67
CAN_SETDOWNLOADID	56
CAN_STATUS	57
CAN_TX	70
CAN_TX_ENH	71
CAN_TX_ENH_CYCLIC	73
CANOPEN_ENABLE	80
CANOPEN_GETBUFFERFLAGS	82
CANOPEN_GETEMCYMESSAGES	119
CANOPEN_GETERRORREGISTER	121
CANOPEN_GETGUARDBERRLIST	115
CANOPEN_GETGUARDBSTATSLV	116
CANOPEN_GETNMTSTATESLAVE	89
CANOPEN_GETODCHANGEDFLAG	93
CANOPEN_GETSTATE	84
CANOPEN_GETSYNCSTATE	111
CANOPEN_NMTSERVICES	90
CANOPEN_READOBJECTDICT	94
CANOPEN_SDOREAD	98
CANOPEN_SDOREADBLOCK	100
CANOPEN_SDOREADMULTI	102
CANOPEN_SDOWRITE	104
CANOPEN_SDOWRITEBLOCK	106
CANOPEN_SDOWRITEMULTI	108
CANOPEN_SENDEMCMYMESSAGE	122
CANOPEN_SETSTATE	86
CANOPEN_SETSYNCSTATE	113
CANOPEN_WRITEOBJECTDICT	95
CAN-Schnittstellen	17
CAN-Schnittstellen deklarieren (z.B. CR1080)	41
CAN-Stack	190
CiA	190
CiA DS 304	190
CiA DS 401	190
CiA DS 402	190
CiA DS 403	190
CiA DS 404	190
CiA DS 405	190
CiA DS 406	190
CiA DS 407	190
COB-ID	191
CODESYS	191

CODESYS-Programmierhandbuch	5
CODESYS-Visualisierungs-Elemente	31
Copyright	4
CSV-Datei	191

D

Datentyp	191
DC	192
Diagnose	184, 192
Diagnose und Fehlerbehandlung	184
Dither	192
DLC	192
DRAM	192
DTC	192

E

ECU	192
EDS-Datei	193
Embedded Software	193
EMCY	193
EMCY-Codes	
CANx	188
System	188
EMV	193
ERROR-Zustand	26
Ethernet	193
EUC	193

F

Farb-Display des CR0451	14
FATAL-ERROR-Zustand	27
FB, FUN, PRG in CODESYS	22
Fehlanwendung	193
Fehler	184
CAN / CANopen	188
Fehlermerker	188
Fehler-Tabellen	188
FiFo	194
FLASH_INFO	157
FLASH_READ	158
Flash-Speicher	194
FLASH-Speicher	13
FRAM	13, 194
Funktionskonfiguration, allgemein	43
Funktionsrückmeldungen des Geräts	15
Für Netzwerkvariablen erforderliche Bibliotheken	46

G

Gerätekonfiguration	35
GET_APP_INFO	159
GET_HW_INFO	160
GET_IDENTITY	161
GET_SW_INFO	162
GET_SW_VERSION	163
GET_TEXT_FROM_FLASH	175
Grenzen für CAN in diesem Gerät	34
Grenzen für CAN J1939 in diesem Gerät	34
Grenzen für CANopen in diesem Gerät	34

H

Hardware-Aufbau	12
Hardware-Beschreibung	12
Heartbeat	194
Historie der Anleitung (CR0451)	8
HMI	194

I

ID – Identifier	194
IEC 61131	195
IEC-User-Zyklus	195
ifm weltweit • ifm worldwide • ifm à l'échelle internationale	211
ifm-Bausteine für das Gerät CR0451	51
ifm-Bibliotheken für das Gerät CR0451	46
ifm-Funktionselemente	46
ifm-Maintenance-Tool nutzen	24
INIT-Zustand (Reset)	25
Installation verifizieren	37
IP-Adresse	195
ISO 11898	195
ISO 11992	195
ISO 16845	195

J

J1939	195
J1939_DM1RX	149
J1939_DM1TX	151
J1939_DM1TX_CFG	154
J1939_DM3TX	155
J1939_ENABLE	125
J1939_GETDABYNAME	127
J1939_NAME	129
J1939_RX	136
J1939_RX_FIFO	137
J1939_RX_MULTI	139
J1939_SPEC_REQ	133
J1939_SPEC_REQ_MULTI	134
J1939_STATUS	131
J1939_TX	141
J1939_TX_ENH	142
J1939_TX_ENH_CYCLIC	144
J1939_TX_ENH_MULTI	146

K

Klemme 15	195
Konfigurationen	35

L

Laufzeitsystem	19, 196
Laufzeitsystem aktualisieren	37
Laufzeitsystem einrichten	35
Laufzeitsystem neu installieren	36
LED	196
LED im Anwendungsprogramm steuern	16
Leistungsgrenzen des Geräts	28
Link	196
LSB	196

M

MAC-ID	196
Master	196
MEM_ERROR	164
MEMCPY	165
Mindestens erforderliche Bibliotheken	46
MMI	196
MRAM	196
MSB	196

N

Netzwerkvariablen	45
NMT	197
Node	197
Node Guarding	197
NORM_DINT	177
NORM_REAL	178
Notizen • Notes • Notes	207

O

Obj / Objekt	197
Objektverzeichnis	197
OBV	197
OHC	167
OPC	197
operational	197

P

PC-Karte	197
PCMCIA-Karte	198
PDM	198
PDM_PAGECONTROL	182
PDO	198
PDU	198
PES	198
PGN	198
PID-Regler	198
Piktogramm	198
Piktogramme	6
Pre-Op	199
Programmierhinweise für CODESYS-Projekte	21
Programmiersystem einrichten	38
Programmiersystem manuell einrichten	38
Programmiersystem über Templates einrichten	42
Prozessabbild	199
PWM	199

R

rationetrisch	199
RAW-CAN	199
Reaktion auf Fehlermeldungen	185
remanent	199
Reset	25
Retain-Variablen	44
ro	199
RTC	200
Rücklesen von Retain-Variablen	45

RUN-Zustand	26
rw	200

S

SAE J1939	124, 200
Schnittstellen	14
Schnittstellen-Beschreibung	17
SD-Card	200
SDO	200
Selbsttest	200
SET_IDENTITY	169
SET_LED	170
SET_PASSWORD	172
Sicherheitshinweise	9
Sichern von Retain-Variablen	45
Slave	201
Software	18
Software-Module für das Gerät	18
Software-Steuerungskonfiguration	39
Speicher, verfügbar	13
SRAM	13
Status-LED	16
Steuerungskonfiguration	39, 201
Steuerungskonfiguration aktivieren	40
stopped	201
STOP-Zustand	25
Symbole	201
Systembeschreibung	11
Systemmarker	186
Systemvariable	201
Systemvariablen	43
Systemvoraussetzungen	11

T

Target	201
Target einrichten	39
Tasten-LEDs dimmbar	15
TCP	201
Template	201
Texte	32
TIMER_READ_US	173
TOGGLE	179

U

Über diese Anleitung	4
Übersicht	
Dokumentations-Module für ecomatmobile-Geräte	5
UDP	202

V

Variablen	44
Verfügbarer Speicher	13
Verhalten des Watchdog	28
Verteilen des Anwendungsprogramms	24
Verwendung, bestimmungsgemäß	202
Visualisierungsgrenzen	29
Vorkenntnisse	10

W

Was bedeuten die Symbole und Formatierungen?	6
Watchdog	28, 202
Welche Vorkenntnisse sind notwendig?	10
Wie ist diese Dokumentation aufgebaut?	7
wo	202

Z

Zeichenbereich	32
Zubehör	11
Zykluszeit	202
Zykluszeit beachten!	22





11 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Stand: 2015-03-06

8310

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	ifm electronic gmbh Vertrieb Deutschland Niederlassung Nord • 31135 Hildesheim • Tel. 0 51 21 / 76 67-0 Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0 Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0 Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0 Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0 Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0 Niederlassung Ost • 07639 Tautenhain • Tel. 0 36 601 / 771-0 ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen
A	ifm electronic gmbh • 1120 Wien • Tel. +43 16 17 45 00
AUS	ifm efector Pty Ltd. • Mulgrave Vic 3170 • Tel. +61 3 00 365 088
B, L	ifm electronic N.V. • 1731 Zellik • Tel. +32 2 / 4 81 02 20
BR	ifm electronic Ltda. • 03337-000, Sao Paulo SP • Tel. +55 11 / 2672-1730
CH	ifm electronic ag • 4 624 Härkingen • Tel. +41 62 / 388 80 30
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • Tel. +86 21 / 3813 4800
CND	ifm efector Canada inc. • Oakville, Ontario L6K 3V3 • Tel. +1 800-441-8246
CZ	ifm electronic spol. s.r.o. • 25243 Průhonice • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 BROENDBY • Tel. +45 70 20 11 08
E	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
F	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
FIN	ifm electronic oy • 00440 Helsinki • Tel. +358 75 329 5000
GB, IRL	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 208 / 213-0000
GR	ifm electronic Monoprosopi E.P.E. • 15125 Amaroussio • Tel. +30 210 / 6180090
H	ifm electronic kft. • 9028 Győr • Tel. +36 96 / 518-397
I	ifm electronic s.a. • 20041 Agrate-Brianza (MI) • Tel. +39 039 / 68.99.982
IL	Astragal Ltd. • Azur 58001 • Tel. +972 3 -559 1660
IND	ifm electronic India Branch Office • Kolhapur, 416234 • Tel. +91 231-267 27 70
J	efector co., Ltd. • Chiba-shi, Chiba 261-7118 • Tel. +81 043-299-2070
MAL	ifm electronic Pte. Ltd. • 47100 Puchong Selangor • Tel. +603 8063 9522
MEX	ifm efector S. de R. L. de C. V. • Monterrey, N. L. 64630 • Tel. +52 81 8040-3535
N	Sivilingeniør J. F. Knudtzen A/S • 1396 Billingstad • Tel. +47 66 / 98 33 50
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341 / 438 438
P	ifm electronic s.a. • 4410-136 São Félix da Marinha • Tel. +351 223 / 71 71 08
PL	ifm electronic Sp. z o.o. • 40-106 Katowice • Tel. +48 32-608 74 54
RA, ROU	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 / 5353 3436
ROK	ifm electronic Ltd. • 140-884 Seoul • Tel. +82 2 / 790 5610
RP	Gram Industrial, Inc. • 1770 Mantilupa City • Tel. +63 2 / 850 22 18
RUS	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
S	ifm electronic a b • 41250 Göteborg • Tel. +46 31 / 750 23 00
SGP	ifm electronic Pte. Ltd. • Singapore 609 916 • Tel. +65 6562 8661/2/3
SK	ifm electronic s.r.o. • 835 54 Bratislava • Tel. +421 2 / 44 87 23 29
THA	SCM Alliances Co., Ltd. • Bangkok 10 400 • Tel. +66 02 615 4888
TR	ifm electronic Ltd. Sti. • 34381 Sisli/Istanbul • Tel. +90 212 / 210 50 80
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501 8543
USA	ifm efector inc. • Exton, PA 19341 • Tel. +1 610 / 5 24-2000
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 345 44 49

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.