Festo CPX Driver Block Library for SIMATIC PCS 7

Table of Contents

1 Introduction/Designated Use

The Festo CPX driver block library described in this documentation is designed for seamlessly integrating Festo CPX remote I/O terminals with or without attached valve terminals to a Siemens SIMATIC PCS 7 process control system. It fully supports the "Generate Module Drivers" function of PCS 7 and allows for convenient and automated set-up and module and channel diagnosis.

Please check the System Requirements about supported modules and required configuration settings.

This library is not to be used simultaneously with other manufacturer's libraries for Festo CPX, including the one by Siemens. Such libraries need to be removed from the Engineering System before installing the Festo CPX driver block library.

It is absolutely necessary to follow the safety instructions as well as the designated use of the relevant hardware referred to in this documentation. Please check the appropriate hardware manuals for further information.

If additional components such as sensors and actuators are connected, the specified limits for pressures, temperatures, electrical data, torques, etc. must not be exceeded.

Please do also check the Siemens specifications regarding the use of SIMATIC PCS 7.

The Festo CPX driver block library can be used free of charge.

2 Important User Information

2.1 Danger Categories

This document contains information on possible dangers that can occur if the product is not used as designated. This information is marked with a signal word (warning, caution, etc.), placed on a grey background and additionally marked with a pictogram. A distinction is made between the following danger warnings:

Warning

... means that serious injury to people and damage to property can occur if this warning is not observed.

Caution

... means that injury to people and damage to property can occur if this warning is not observed.

Note

... means that damage to property can occur if this warning is not observed.

You can find information on how special information is represented at "Marking Special Information".

2.2 Limited Warranty

Please note that it is not possible with the present state of technology to create programmed software which functions without problems and is compatible with all applications and combinations intended by the user. As a rule the software must therefore be used in the designated manner as specified in the program description and in the user instructions.

At the moment when the software is transferred or made available, it is in a state in which it will function under normal operating and application conditions. Festo does not guarantee that the software will suffice for all applications and purposes intended by the user, or that it will function without problems when used with other programs, or that it is compatible with these programs. The responsibility for the correct selection and the consequences of using the software within the scope of use defined by the user, as well as for the intended and achieved results therefore lies with the user. The same applies to the written material supplied with the software.

The use of the programmed software does not exempt you as customer from your duties and responsibility for observing and adhering to technical machine and safety regulations as well as for a comprehensive functional check.

3 System Requirements

The Festo CPX driver library for PCS 7 has been tested with **SIMATIC PCS 7 Version 9.1 SP2**. For Profinet support, at least **Basis Library V9.1 SP 2** is required. If you intend to install the CPX terminal behind a Siemens Y-Link, at least **Basis Library V9.1 SP 2** is required. It might also work with older versions of PCS 7, but no guarantee is given. A PCS 7 compliant CPU with ALARM_8P support is to be used.

For CPX terminals connected via **Profibus DP**, the "**DP-Slave [DPV1]**" GSD file entry is to be used for the CPX-FB13 bus node module, and the latter is to be installed in the terminal's first slot. If **Profinet IO** is used, the suitable "**FB3x PNIO Modul[STI]**" entry is to be used for configuring the bus node in slot 1, depending on whether you use FB33, 34 or 35.

For each module driver block, the "Area Of Application" section provides information about which modules are supported. A comprehensive list of all supported modules can be found in an own section, too. Some modules have officially been released for usage with the driver library, as can be seen from the "Release" column there. Modules not officially released are technologically supported, but have not been tested to the same extend as the released ones.

If a module is installed and configured in HW Config that is not supported by the Festo CPX driver block library, two scenarios exist:

1. At least one of the module's digital or analog channels is accessed from within the PCS 7 program, using an appropriate channel driver block. In this case, the "Generate Module Drivers" function will raise an error, stating that for the channel in question, no supported hardware is found.

2. None of the module's channels is accessed via a channel driver block. In this case, no error is raised by the "Generate Module Drivers" function, and the system will operate normally. However, as soon as an error occurs on the respective module, recurring error messages are triggered by the terminal driver block ("Fatal system error"). This is since the driver block recognizes the presence of a module error, but there is no suitable module driver block to provide detailed diagnostics.

Always make sure only to use modules that are supported by the Festo CPX driver block library. Please consult the "Module Driver Blocks" chapters for finding out which modules are supported.

If you use **Profibus DP**, the fieldbus node **CPX-FB13 must be Rev. 22 or higher**. Please check the label on the fieldbus node, valid entries are either "SW 17.04.07", "R23", "R24" or any higher revision number. Please note that changing the byte order for analog values **(Intel/Motorola) requires revision 23** or newer.

The CPX System supports **S2 Redundancy**.

For **Profinet IO**, the **CPX-FB3x bus node must be Rev. 19 or higher**. Please be aware that the Fast Startup mechanism is not supported.

In HW Config, the **"DP Interrupt Mode"** setting of the CPX terminal is to be kept at the default **"DPV0"** setting (HW Config/DP Slave Properties/Parameter Assignment/Station Parameters/DP Interrupt Mode). If DPV1 is selected here, certain system error messages might not be triggered correctly under rare circumstances.

It is furthermore advised to set the **"Filter alarm Vout/Vven"** station parameter to **"active"** instead of the default "inactive" value. This prevents the fieldbus nodes from flooding the bus system with alarm messages in case of undervoltage, and the default "inactive" setting might lead to sporadic "Device x/y/z: Multiple alarm (OB82)" alarm messages. The reason is that for each module, a separate diagnostic message is sent via Profibus, triggering OB 82. PCS 7 monitors the number of OB calls, and raises an alarm as soon as more than 5 instances are called.

Suppressing such messages via the "Filter alarm Vout/Vven" setting does not affect the Festo CPX driver library; i. e. even with the filter activated, undervoltage alarms are displayed correctly. The Festo CPX driver library does not use OB 82 for that purpose but does a monitoring of the status byte sent by the fieldbus node instead.

For direct access to this **online help system** from SIMATIC Manager (via the F1 key), PCS 7 Version 7.1 or newer must be installed.

This library supports both the **PCS 7 Standard Library** and the **Advanced Process Library (APL)** standards by Siemens.

4 What's New In This Version?

The following table provides an overview of the versions of the Festo CPX driver block library for PCS 7:

Version	Date	Description
1.0.0.0	20.04.2011	Initial version; support for PCS 7 Standard Library (no APL support)
1.0.1.0	04.07.2011	Maintenance release: - RACK_CPX: Prevented periphery access error in case FB13 Profibus node was not configured correctly; added QCFG_ERR to indicate an incorrectly configured CPX-FB13 Profibus node - MOD_VTX: Resolved OMODE8 not being updated correctly
1.1.0.0	25.07.2011	Minor release: - Added support for CPX-L-16DI and 8DI-8DO modules
1.1.1.0	28.03.2012	Maintenance release: - Corrected formal error in AL_CH_FE_CPX059E.xml
2.0.0.0	04.04.2013	Major release: - Added support for APL standard (new channel driver blocks CPXAnIn, CPXAnOu, CPXDiIn, CPXDiOu) - Added support for CPX-P-8DE-N... modules - Added support for VMPAL-EPL-CPX pneumatic interface - Added support for Profinet IO Additional changes: - RACK_CPX: Collective fault messages read from CPX status byte are suppressed by default (new parameter EN_STMSG) - CHAI_CPX: High and low range values are set up automatically for pressure sensor channels and forwarded to respective output parameters to allow for proper processing on succeeding blocks; added immediate broken wire detection via analog input value for temperature sensors - CHAO_CPX: Added consistency check for scaling limits
2.0.1.0	12.07.2013	Maintenance release: - Changed PNDRVSWT_333_Festo.xml file name to PNDRVSWTVENDOR_333_Festo.xml due to PCS 7 V8.0 SP1 Basis Library Update 2 requirements
2.1.1.0	19.05.2015	Minor release: - Added support for I-port CPV via CPX-CTEL I-Port master module - RACK_CPX: Changed periphery access for status interface to PIW/PQW for better support of partial process images; resolved channel diagnosis being shifted erroneously in rare cases - CPXAnIn + CH_AI: Activated substitution value also in case of a NAMUR limit error for current signals
2.1.2.0	07.07.2015	Maintenance release: - CH_AI, CPXAnIn: Corrected error in signal current limit check (block ver. 1.2/lib ver 2.1.1.0 regression)
2.1.3.0	14.11.2017	Maintenance release: - Added support for PROFINET node rev. 30
2.2.0.3	12.02.2024	Minor release: - Added support for CPX-CTEL module with 16/24/32 digital output bytes
2.2.0.4	25.03.2025	Minor release. - Documentation adapted.

Note

There might be updates added from time to time. Please check our Support Portal from time to time for newer versions of the Festo CPX driver block library for PCS 7.

5 Supported Modules

Some modules have officially been released for usage with the driver library, as can be seen from the "Release" column in the table below. Modules not officially released are technologically supported, but have not been tested to the same extend as the released ones.

The following hardware modules are supported by the Festo CPX driver library for PCS 7:

Type	GSD/GSDML File Entries	Description	Release
CPX-4DE	CPX-4DE [4DI]x2, *CPX-4DE [4DI]x0	4 channel digital input module	No
CPX-8DE	CPX-8DE [8DI]	8 channel digital input module	Yes
CPX-8DE-D	CPX-8DE-D [8DI-D]	8 channel digital input module with channel diagnosis	Yes
CPX-8NDE	CPX-8NDE [8NDI]	8 channel digital input module, negative logic	No
CPX-16DE	CPX-16DE [16DI]	16 channel digital input module	Yes
CPX-L-16DE-CC	CPX-L-16DE-CC [L-16DI-CC]	16 channel digital input module, IP20	Yes
CPX-M-16DE-D	CPX-M-16DE-D [M-16DI-D]	16 channel digital input module with channel diagnosis, metal housing	No
CPX-4DA	CPX-4DA [4DO]x2, *CPX-4DA [4DO]x0	4 channel digital output module	No
CPX-8DA	CPX-8DA [8DO]	8 channel digital output module	Yes
CPX-8DA-H	CPX-8DA-H [8DO-H]	8 channel digital output module, high current	No
CPX-L-8DE-8DA-CC	CPX-L-8DE-8DA-CC [L-8DI/8DO-CC]	8/8 channel digital input/output multi I/O module, IP20	Yes
CPX-8DE-8DA	CPX-8DE-8DA [8DI/8DO]	8/8 channel digital input/output multi I/O module	Yes
CPX-2AE-U-I	CPX-2AE-U/I [2AI]	2 channel analog input module, voltage or current, 12 bit resolution	Yes
CPX-4AE-I	CPX-4AE-I [4AI-I]	4 channel analog input module, current only, 12 bit resolution	Yes
CPX-4AE-U-I	CPX-4AE-U-I [4AI-U-I]	4 channel analog input module, voltage or current, 15 bit resolution	Yes
CPX-2AE-T	CPX-2AE-T [2AI-T]	2 channel analog input module, temperature sensors	Yes
CPX-4AE-T	CPX-4AE-T [4AI-T]	4 channel analog input module, temperature sensors	Yes
CPX-4AE-TC	CPX-4AE-TC [4AI-TC]	4 channel analog input module, thermocouple sensors	No
CPX-2AA-U-I	CPX-2AA-U/I [2AO]	2 channel analog output module, voltage or current, 12 bit resolution	Yes
CPX-4AE-P-B2	CPX-4AE-P-B2 [4AI-P]	4 channel pressure sensor module, -1..+1 bar	Yes
CPX-4AE-P-D10	CPX-4AE-P-D10 [4AI-P]	4 channel pressure sensor module, 0..10 bar	Yes
CPX-CP-4-FB	CPI: 0 Byte I/0 Byte O CPI: 0 Byte I/4 Byte O	CP interface (master module)	No

	CPI: 0 Byte I/8 Byte O CPI: 0 Byte I/12 Byte O CPI: 0 Byte I/16 Byte O CPI: 4 Byte I/0 Byte O CPI: 4 Byte I/4 Byte O CPI: 4 Byte I/8 Byte O CPI: 4 Byte I/12 Byte O CPI: 4 Byte I/16 Byte O CPI: 8 Byte I/0 Byte O CPI: 8 Byte I/4 Byte O CPI: 8 Byte I/8 Byte O CPI: 8 Byte I/12 Byte O CPI: 8 Byte I/16 Byte O CPI: 12 Byte I/0 Byte O CPI: 12 Byte I/4 Byte O CPI: 12 Byte I/8 Byte O CPI: 12 Byte I/12 Byte O CPI: 12 Byte I/16 Byte O CPI: 16 Byte I/0 Byte O CPI: 16 Byte I/4 Byte O CPI: 16 Byte I/8 Byte O CPI: 16 Byte I/12 Byte O CPI: 16 Byte I/16 Byte O		
CPX-P8DE-N	CPX-P-8DE-N [P8DI-N] CPX-P-8DE-N X [P8DI-N X]	8 channel digital NAMUR input module, for non EX signals	Yes
CPX-P8DE-N-IS	CPX-P-8DE-N-IS [P8DI-N-IS] CPX-P-8DE-N-IS X [P8DI-N-IS X]	8 channel digital NAMUR input module, for intrinsically safe signals	Yes
VMPA1-FB-EMS-8	MPA1S: VMPA1-FB-EMS-8 [8DO]	size 1, 8 valve coils, without electrical isolation	Yes
VMPA1-FB-EMS-D2-8	MPA1SD: VMPA1-FB-EMS-D2-8 [8DO]	size 1, 8 valve coils, without electrical isolation, diagnostics	Yes
VMPA1-FB-EMG-8	MPA1G: VMPA1-FB-EMG-8 [8DO]	size 1, 8 valve coils, with electrical isolation	Yes
VMPA1-FB-EMG-D2-8	MPA1GD: VMPA1-FB-EMG-D2-8 [8DO]	size 1, 8 valve coils, with electrical isolation, diagnostics	Yes
VMPA2-FB-EMS-4	MPA2S: VMPA2-FB-EMS-4 [4DO]x2 *MPA2S: VMPA2-FB-EMS-4 [4DO]x0	size 2, 4 valve coils, without electrical isolation	Yes
VMPA2-FB-EMS-D2-4	MPA2SD: VMPA2-FB-EMS-D2-4[4DO]x2 *MPA2SD:VMPA2-FB-EMS-D2-4[4DO]x0	size 2, 4 valve coils, without electrical isolation, diagnostics	Yes
VMPA2-FB-EMG-4	MPA2G: VMPA2-FB-EMG-4 [4DO]x2 *MPA2G: VMPA2-FB-EMG-4 [4DO]x0	size 2, 4 valve coils, with electrical isolation	Yes
VMPA2-FB-EMG-D2-4	MPA2GD:VMPA2-FB-EMG-D2-4 [4DO]x2 *MPA2GD:VMPA2-FB-EMG-D2-4[4DO]x0	size 2, 4 valve coils, with electrical isolation, diagnostics	Yes
VMPA-FB-PS-1	VMPA-FB-PS [1AI-P]	MPA pressure sensor, monitoring pneum. channel 1	Yes
VMPA-FB-PS-3/5	VMPA-FB-PS [1AI-P]	MPA pressure sensor, monitoring pneum. channels 3 and 5	Yes
VMPA-FB-PS-P1	VMPA-FB-PS [1AI-P]	MPA pressure sensor, monitoring externally connected pressure	Yes
VMPAF-FB-	VMPAF-FB-EPL-PS [1AI-P]	MPAF pneumatic interface with	Yes

EPL-PS		pressure sensor	
VPPM-6TA-L-1-F-0L2H	VPPM 2 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..2 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L2H-S1	VPPM 2 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..2 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L2H-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..2 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L2H-S1-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..2 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H	VPPM 6 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..6 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H-S1	VPPM 6 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..6 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..6 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H-S1-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..6 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L10H	VPPM 10 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..10 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L10H-S1	VPPM 10 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..10 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L10H-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..10 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L10H-S1-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..10 bar, 1 % accuracy	Yes
VPPM-8TA-L-1-F-0L2H-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..2 bar, 2 % accuracy	Yes
VPPM-8TA-L-1-F-0L2H-S1-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..2 bar, 1 % accuracy	Yes
VPPM-8TA-L-1-F-0L6H-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..6 bar, 2 % accuracy	Yes
VPPM-8TA-L-1-F-0L6H-S1-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..6 bar, 1 % accuracy	Yes
VPPM-8TA-L-1-F-0L10H-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..10 bar, 2 % accuracy	Yes
VPPM-8TA-L-1-F-0L10H-S1-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..10 bar, 1 % accuracy	Yes

VABA-1056-X1	**Until 08/2012:** ISO Plug-In DIL1 [8DO] ISO Plug-In DIL2 [16DO] ISO Plug-In DIL3 [24DO]	CPX pneumatic interface for VTSA pneumatics (ISO, type 44)	Yes
	ISO Plug-In DIL4 [32DO]#def **Since 09/2012:** VTSA Type44/45 DIL1 [8DO] VTSA Type44/45 DIL2 [16DO] VTSA Type44/45 DIL3 [24DO] VTSA Type44/45 DIL4 [32DO]#def		
VABA-1056-X2	**Until 08/2012:** ISO Plug-In DIL1 [8DO] ISO Plug-In DIL2 [16DO] ISO Plug-In DIL3 [24DO] ISO Plug-In DIL4 [32DO]#def **Since 09/2012:** VTSA Type44/45 DIL1 [8DO] VTSA Type44/45 DIL2 [16DO] VTSA Type44/45 DIL3 [24DO] VTSA Type44/45 DIL4 [32DO]#def	CPX pneumatic interface for VTSA pneumatics (ISO, type 44), metal version	Yes
VMPAL-EPL-CPX	VMPAL-EPL-CPX DIL1 [8DO] VMPAL-EPL-CPX DIL2 [16DO] VMPAL-EPL-CPX DIL3 [24DO] VMPAL-EPL-CPX DIL4 [32DO]#def	CPX pneumatics interface for MPA-L pneumatics	Yes
CPX-GP-03-4.0	Midi/Maxi DIL1 [8DO] Midi/Maxi DIL2 [16DO] Midi/Maxi DIL3 [24DO] Midi/Maxi DIL4 [32DO]#def	CPX pneumatic interface for Midi/Maxi (Type 03)	No
CPX-GP-CPA-10	CPA10/CPA14 DIL1 [8DO] CPA10/CPA14 DIL2 [16DO] CPA10/CPA14 DIL3 [24DO]#def	CPX pneumatic interface for CPA10	No
CPX-GP-CPA-14	CPA10/CPA14 DIL1 [8DO] CPA10/CPA14 DIL2 [16DO] CPA10/CPA14 DIL3 [24DO]#def	CPX pneumatic interface for CPA14	No
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/8 Byte O	CTEL interface (master module)	V2.2.0.3
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/16 Byte O	CTEL interface (master module)	V2.2.0.3
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/24 Byte O	CTEL interface (master module)	V2.2.0.3
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/32 Byte O	CTEL interface (master module)	V2.2.0.3

Note

There might be updates added from time to time. Please check our Support Portal from time to time for newer versions of the Festo CPX driver library for PCS 7.

6 Installation Notes

Installation of the Festo CPX driver block library for PCS 7 is done by executing the setup file "CPX_PCS7_v[Version Name].exe" that can be downloaded form the Festo Support Portal. It is then automatically installed in the standard folder for PCS 7 libraries, e. g. C:\Program Files\SIEMENS\STEP7\S7LIBS. The exact path depends on the installation folder you chose when installing PCS 7.

The library is furthermore registered automatically with SIMATIC Manager, such that it becomes available in the "Libraries" tab in CFC editor.

An context-sensitive online help document can be accessed by selecting a driver block in SIMATIC Manager or CFC editor and pressing F1.

Note

Please make sure that all editors of PCS 7 are closed during installation.

Note

There might be updates added from time to time. Please check our Support Portal from time to time for newer versions of the Festo CPX driver block library for PCS 7.

7 Rack Driver Blocks

7.1 RACK_CPX: Festo CPX Terminal Monitoring

7.1.1 RACK_CPX: Festo CPX Terminal Monitoring

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

7.1.2 Description of RACK_CPX

Object Name (Type + Number)

FB 800

Area of Application

RACK_CPX monitors the status of a Festo CPX Terminal with electrical I/O and/or valve terminal modules connected via Profibus DP or Profinet IO. The "DP-Slave [DPV1]" GSD file entry is to be used for the CPX-FB13 bus node module, and the latter is to be installed in the terminal's first slot. For Profinet IO, choose the "FB3x PNIO Modul [STI]" entry for configuring the bus node.

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 82	Diagnostic Interrupt Organization Block
OB 86	Rack Failure Organization Block
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX Terminal found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- OB_DIAG1 is installed in the run sequence after the corresponding SUBNET block (for monitoring the Profibus DP slave/Profinet IO device upon failure and recovers; see Siemens Help for PCS 7 Library for more details).
- RACK_CPX is installed in the run sequence after the previously installed OB_DIAG1 block.
- The RACK_NO, SUBN_ID, and EN_MSG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The EN input of OB_DIAG1 is interconnected with the output of an AND block whose inputs are interconnected to the EN_SUBx output of the OB_BEGIN block, and to the EN_Rx output of the SUBNET block.
- The CPU_DIAG and CPU_OB5x OUT structures of the OB_BEGIN block and SUB_DIAG of the SUBNET block are interconnected with the IN_OUT structures of the same name of the OB_DIAG1 block.
- The RACKF, SUBN1ERR and SUBN2ERR I/Os of the RACK_CPX block are interconnected with the QRACKF, SUBN1ERR and SUBN2ERR I/Os of the OB_DIAG1 block.
- The DIAG_IW, DIAG_QW, MOD_CNT and CHNCNTx I/Os are configured by the subordinate module drivers MOD_CPXx.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initizalize if

changes are applied online.

Functional Description

The main task of RACK_CPX is to coordinate the access to the CPX Terminal's diagnostic interface. Since the subordinate MOD_CPXx driver blocks of the same terminal need to request data via the I/O interface, but the interface can only be accessed by one single block, RACK_CPX works as router between the module driver blocks and the hardware.

During normal operation, the terminal's status byte send by the bus node is just checked upon errors. As soon as a collective fault is signalled, RACK_CPX starts requesting module specific diagnostic data to find out which of the installed modules are affected. When all modules have been checked, RACK_CPX relays the diagnostic interface (both input and output word) to the installed modules one after the other. This is done by forwarding the information received via the respective DIAGxQW connector to the output word defined by DIAG_QW. The response by the CPX Terminal as read from the input word defined by DIAG_IW is consequently forwarded to the respective DIAGxIW connector.

As soon as the module driver block indicates that a communication cycle has been completed (i. e. all channels have been checked upon errors), RACK_CPX shifts to the next faulty module. When the last module has finished its diagnostic cycle, RACK_CPX starts the whole thing over, first checking the status byte upon collective faults and - if a fault is still present - checking the module errors. If no collective fault is coded in the status byte, all DIAGxIW connectors are set to 0, signalling the module driver blocks that no errors are present anymore.

If the OB_DIAG1 block superior to RACK_CPX detects a slave failure (complete DP slave missing or not working), this is signalled via the RACKF connector to RACK_CPX which in turn forwards the information to the module driver blocks via its QRACKF connector.

If the OB_DIAG1 block signals a subnet/redundancy error via the SUBN1ERR/SUBN2ERR connectors, the error information is forwarded to the module driver blocks by setting QSUBN_ERR = TRUE.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the RACK_CPX driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link. For Profinet IO, redundancy is not supported.

Error Handling

There are different levels of error handling. One covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

The second level of error handling covers the routing of the diagnostic interface. If a module driver doesn't release the diagnostic interface within the stretch of time configured via the TIMEOUT parameter, RACK_CPX generates an error message via ALARM_8P ("fatal system error") and continues with the next module.

The third level ensures that the diagnostic interface's addresses (DIAG_IW/DIAG_QW) are configured correctly, i. e. that they can be found in the hardware configuration for the first slot of the respective CPX Terminal. A "Configuration Error" message is triggered, and QCFG_ERR is set to TRUE if this is not the case. The same message is also raised if DPV1 is not activated on the Profibus master's side. Please make sure that the "DP-Slave [DPV1]" GSD file entry is used for configuring the CPX-FB13 Profibus node or "FB3x PNIO Modul [STI]" for Profinet IO devices (HW Config, first slot of CPX Terminal) if a configuration error is indicated.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, RACK_CPX initializes its internal error memory and resets the diagnostic information sent to the module driver blocks. It also calculates the CHNIDXx channel index values which tell the MOD_CPXx blocks the number of their first respective channel, based on the values set up at the CHNCNTx connectors. Last not least, the addresses configured via DIAG_IW and DIAG_QW are checked upon correctness (see above).

After startup, messages are suppressed and the status byte is ignored for the time specified by DELAY in order to allow for proper bus protocol initialization.

Messaging Functionality

RACK_CPX only reports events that apply to the monitored CPX Terminal. The block provides messages for the collective faults sent by the terminal via the status byte (e. g. "valve module error"). Additional messages are raised in case of a time out of the diagnostic interface or configuration errors. ALARM_8P is used for that purpose.

Messages are suppressed if a rack failure notification is received from OB_DIAG1 via the RACKF input connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

CPX status byte messages are suppressed by default to prevent message flooding; they can be enabled by setting EN_STMSG = TRUE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

7.1.3 I/Os of RACK_CPX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF	X	V
SUBN1_ID	ID of primary DP master system (when behind a Y-Link, Profibus only)	BYTE	16#FF	X	V
SUBN2_ID	ID of redundant DP master system (when behind a Y-Link, Profibus only)	BYTE	16#FF	X	V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
NODETYPE	Type of field bus node (FB13/33/34/35)	INT	0	X	V
DELAY	Startup alarm delay (sec)	INT	4		
TIMEOUT	Timeout for reply by CPX on diagnostic request	INT	2		
EV_ID	Message ID	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
EN_STMSG	Enable collective fault messages read from status byte	BOOL	FALSE		
RACKF	From OB_DIAG1: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN1ERR	From OB_DIAG1: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
SUBN2ERR	From OB_DIAG1: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
DIAG_IW	Address of input word coming from FB13/FB3x diagnostic interface (-1=not set); the address of the interface is supplied directly as Integer, not via connecting to the respective input word.	INT	-1	X	V
DIAG_QW	Address of output word to FB13/FB3x diagnostic interface (-1=not set); the address of the interface is supplied	INT	-1	X	V

	directly as Integer, not via connecting to the respective output word.				
DIAGxQW (x=1..47)	Output word of FB13/FB3x diagnostic interface (from module x); forwarded to DIAG_QW as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
CHNCNTx (x=1..47)	Number of I/O channels of module x; used for calculating the CHNIDXx values sent to the module driver blocks	BYTE	16#0	X	X
MOD_CNT	Number of modules installed	INT	0	X	X
MS	Maintenance state	WORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	C
QSUBNERR	1=DP subnet system failure (redundancy error, Profibus only)	BOOL	FALSE	N/A	C
QCFG_ERR	1=Configuration error (CPX-FB13 Profibus node not configured as "DP-Slave [DPV1]" or CPX-FB3x Profinet node not configured as "FB3x PNIO Modul [STI]")	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
MSG_ACK	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STAT	Error information of ALARM_8P	WORD	16#0	N/A	
DIAGxIW (x=1..47)	Input word of FB13/FB3x diagnostic interface (for module x); forwarded from DIAG_IW as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0	N/A	C
CHNIDXx (x=1..47)	First channel index of module x; calculated by adding the number of channels of all previous modules; needed by the module driver block for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

7.1.4 Message Texts And Associated Values of RACK_CPX

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID	SIG1	CPX Terminal @1%d@/@2%d@: Valve module error	S
EV_ID	SIG2	CPX Terminal @1%d@/@2%d@: Digital output module error	S
EV_ID	SIG3	CPX Terminal @1%d@/@2%d@: Digital input module error	S
EV_ID	SIG4	CPX Terminal @1%d@/@2%d@: Analog/function module error	S
EV_ID	SIG5	CPX Terminal @1%d@/@2%d@: Fatal system error	S
EV_ID	SIG6	CPX Terminal @1%d@/@2%d@: Configuration error	S
EV_ID	SIG7	(not used)	N/A
EV_ID	SIG8	(not used)	N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8 Module Driver Blocks

8.1 MOD_CPX1: Festo CPX Digital Input Module

8.1.1 MOD_CPX1: Festo CPX Digital Input Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.1.2 Description of MOD_CPX1

Object Name (Type + Number)

FB 801

Area of Application

MOD_CPX1 monitors up to 16 channels of a Festo CPX digital input module, with or without channel diagnosis.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-4DE	CPX-4DE [4DI]x2, *CPX-4DE [4DI]x0	4 channel digital input module	No
CPX-8DE	CPX-8DE [8DI]	8 channel digital input module	Yes
CPX-8DE-D	CPX-8DE-D [8DI-D]	8 channel digital input module with channel diagnosis	Yes
CPX-8NDE	CPX-8NDE [8NDI]	8 channel digital input module, negative logic	No
CPX-16DE	CPX-16DE [16DI]	16 channel digital input module	Yes
CPX-L-16DE-CC	CPX-L-16DE-CC [L-16DI-CC]	16 channel digital input module, IP20	Yes
CPX-M-16DE-D	CPX-M-16DE-D [M-16DI-D]	16 channel digital input module with channel diagnosis, metal housing	No

Address stacking is supported, so the *CPX-4DE module in the hardware catalogue can be used after CPX-4DE in order to fill up the binary addresses x.4 to x.7.

The following restrictions apply:

- Address stacking is only supported on Profibus DP. For CPX Terminals connected via Profinet IO, bits 4..7 of a byte are left unused when configuring four channel digital modules.
- For the "Generate Module Drivers" to recognize packed addresses correctly, please make sure to (in chronological order) first place the non-asterisk module for the lower four bits and then the asterisk module for the higher four bits. If you encounter errors or warnings when executing the "Generate Module Drivers" function referring to one or several modules using packed addresses, try to do the following:
  - Delete the respective (asterisk-)module in HWConfig and save the configuration.
  - Re-insert the module again from the hardware catalogue (thus inserting the asterisk module chronologically **after** the non-asterisk module)
  - Save the configuration and restart the "Generate Module Drivers" function.

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description

- Message Texts And Associated Values

8.1.3 I/Os of MOD_CPX1

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.1.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module	WORD	16#0		C

	Drivers" function will prevent channel and module diagnostics from working!				
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.1.5 Message Texts And Associated Values of MOD_CPX1

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class

EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Short circuit/overload	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Short circuit/overload	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Short circuit/overload	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Short circuit/overload	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Short circuit/overload	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 5: Short circuit/overload	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 6: Short circuit/overload	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 7: Short circuit/overload	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 8: Short circuit/overload	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 9: Short circuit/overload	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 10: Short circuit/overload	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 11: Short circuit/overload	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 12: Short circuit/overload	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 13: Short circuit/overload	S
EV_ID2	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 14: Short circuit/overload	S
EV_ID2	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 15: Short circuit/overload	S
EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Short circuit/overload	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: Module failure	S
EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: Unknown module error no. @4%d@	S
EV_ID3	SIG4		N/A
EV_ID3	SIG5		N/A
EV_ID3	SIG6		N/A
EV_ID3	SIG7		N/A
EV_ID3	SIG8		N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..3	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to

EV_ID1..3	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..3	3	SLOT_NO	Slot number
EV_ID3	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.2 MOD_CPX2: Festo CPX Digital Output Module

8.2.1 MOD_CPX2: Festo CPX Digital Output Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.2.2 Description of MOD_CPX2

Object Name (Type + Number)

FB 802

Area of Application

MOD_CPX2 monitors up to eight channels of a Festo CPX digital output module.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-4DA	CPX-4DA [4DO]x2, *CPX-4DA [4DO]x0	4 channel digital output module	No
CPX-8DA	CPX-8DA [8DO]	8 channel digital output module	Yes
CPX-8DA-H	CPX-8DA-H [8DO-H]	8 channel digital output module, high current	No

Address stacking is supported, so the *CPX-4DA module in the hardware catalogue can be used after CPX-4DA in order to fill up the binary addresses x.4 to x.7.

The following restrictions apply:

- Address stacking is only supported on Profibus DP. For CPX Terminals connected via Profinet IO, bits 4..7 of a byte are left unused when configuring four channel digital modules.
- For the "Generate Module Drivers" to recognize packed addresses correctly, please make sure to (in chronological order) first place the non-asterisk module for the lower four bits and then the asterisk module for the higher four bits. If you encounter errors or warnings when executing the "Generate Module Drivers" function referring to one or several modules using packed addresses, try to do the following:
  - Delete the respective (asterisk-)module in HWConfig and save the configuration.
  - Re-insert the module again from the hardware catalogue (thus inserting the asterisk module chronologically **after** the non-asterisk module)
  - Save the configuration and restart the "Generate Module Drivers" function.

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.2.3 I/Os of MOD_CPX2

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.2.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal	WORD	16#0		C

	is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!				
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.2.5 Message Texts And Associated Values of MOD_CPX2

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 0: Short circuit/overload	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 1: Short circuit/overload	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 2: Short circuit/overload	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 3: Short circuit/overload	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 4: Short circuit/overload	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 5: Short circuit/overload	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 6: Short circuit/overload	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@,module @3%d@, ch. 7: Short circuit/overload	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Undervoltage in power supply	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID2	SIG4		S
EV_ID2	SIG5		S
EV_ID2	SIG6		S
EV_ID2	SIG7		S
EV_ID2	SIG8		S

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..2	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..2	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..2	3	SLOT_NO	Slot number
EV_ID2	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.3 MOD_CPX3: Festo CPX Digital Input/Output Module

8.3.1 MOD_CPX3: Festo CPX Digital Input/Output Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.3.2 Description of MOD_CPX3

Object Name (Type + Number)

FB 803

Area of Application

MOD_CPX3 monitors the 16 channels of a Festo CPX digital input/output module (eight channels each).

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-L-8DE-8DA-CC	CPX-L-8DE-8DA-CC [L-8DI/8DO-CC]	8/8 channel digital input/output multi I/O module, IP20	Yes
CPX-8DE-8DA	CPX-8DE-8DA [8DI/8DO]	8/8 channel digital input/output multi I/O module	Yes

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start

channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.3.3 I/Os of MOD_CPX3

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.3.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		

EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the	WORD	16#0	N/A	C

	diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!				

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.3.5 Message Texts And Associated Values of MOD_CPX3

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, input ch. 0..7: Short circuit/overload	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 0..7: Undervoltage in power supply	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID1	SIG5	(not used)	N/A
EV_ID1	SIG6	(not used)	N/A
EV_ID1	SIG7	(not used)	N/A
EV_ID1	SIG8	(not used)	N/A
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 0: Short circuit/overload	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 1: Short circuit/overload	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 2: Short circuit/overload	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 3: Short circuit/overload	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 4: Short circuit/overload	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 5: Short circuit/overload	S

EV_ID2	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 6: Short circuit/overload	S
EV_ID3	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, output ch. 7: Short circuit/overload	S

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..2	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..2	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..2	3	SLOT_NO	Slot number
EV_ID1	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.4 MOD_CPX4: Festo CPX Analog Input Module, Incl. T/TC Temperature Modules

8.4.1 MOD_CPX4: Festo CPX Analog Input Module, Incl. T/TC Temperature Modules

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.4.2 Description of MOD_CPX4

Object Name (Type + Number)

FB 804

Area of Application

MOD_CPX4 monitors up to four channels of a Festo CPX analog input or output module, including T/TC temperature modules.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-2AE-U-I	CPX-2AE-U/I [2AI]	2 channel analog input module, voltage or current, 12 bit resolution	Yes
CPX-4AE-I	CPX-4AE-I [4AI-I]	4 channel analog input module, current only, 12 bit resolution	Yes
CPX-4AE-U-I	CPX-4AE-U-I [4AI-U-I]	4 channel analog input module, voltage or current, 15 bit resolution	Yes

CPX-2AE-T	CPX-2AE-T [2AI-T]	2 channel analog input module, temperature sensors	Yes
CPX-4AE-T	CPX-4AE-T [4AI-T]	4 channel analog input module, temperature sensors	Yes
CPX-4AE-TC	CPX-4AE-TC [4AI-TC]	4 channel analog input module, thermocouple sensors	No
CPX-2AA-U-I	CPX-2AA-U/I [2AO]	2 channel analog output module, voltage or current, 12 bit resolution	Yes

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.4.3 I/Os of MOD_CPX4

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.4.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the	INT	0	X	C

	terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!				
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver	BOOL	FALSE		V

	generator, re-initialize				

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.4.5 Message Texts And Associated Values of MOD_CPX4

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Short circuit/overload	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Wire fracture/idling at current input/output	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Lower limit exceeded	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Upper limit exceeded	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Parameter fault	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Input overload	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Signal underflow/overflow	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Short circuit/overload	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Wire fracture/idling at current input/output	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Lower limit exceeded	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Upper limit exceeded	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Fault in actuator supply	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Parameter fault	S
EV_ID2	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Input overload	S
EV_ID2	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Signal underflow/overflow	S
EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Short circuit/overload	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Wire fracture/idling at current input/output	S
EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Lower limit exceeded	S
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Upper limit exceeded	S

EV_ID3	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID3	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Parameter fault	S
EV_ID3	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Input overload	S
EV_ID3	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Signal underflow/overflow	S
EV_ID4	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Short circuit/overload	S
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Wire fracture/idling at current input/output	S
EV_ID4	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Lower limit exceeded	S
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Upper limit exceeded	S
EV_ID4	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Short circuit/overload	S
EV_ID4	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Parameter fault	S
EV_ID4	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Input overload	S
EV_ID4	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Signal underflow/overflow	S

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..4	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..4	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..4	3	SLOT_NO	Slot number
EV_ID3	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.5 MOD_CPX5: Festo CPX Pressure Sensor Module

8.5.1 MOD_CPX5: Festo CPX Pressure Sensor Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.5.2 Description of MOD_CPX5

Object Name (Type + Number)

FB 805

Area of Application

MOD_CPX5 monitors up to four channels of a Festo CPX pressure sensor module.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-4AE-P-B2	CPX-4AE-P-B2 [4AI-P]	4 channel pressure sensor module, -1..+1 bar	Yes
CPX-4AE-P-D10	CPX-4AE-P-D10 [4AI-P]	4 channel pressure sensor module, 0..10 bar	Yes

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initizalize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is

connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.5.3 I/Os of MOD_CPX5

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.5.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output);	WORD	16#0	X	V

	needed for requesting channel related information from the terminal				
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.5.5 Message Texts And Associated Values of MOD_CPX5

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Lower limit exceeded	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Upper limit exceeded	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Wrong device type mounted	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Parameter fault	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Sensor limit exceeded	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Invalid process value	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, MOD. @3%d@: Unknown module error number @4%@	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Lower limit exceeded	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Upper limit exceeded	S
EV_ID2	SIG3	(not used)	N/A
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Parameter fault	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Sensor limit exceeded	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Invalid process value	S
EV_ID2	SIG7	(not used)	N/A
EV_ID2	SIG8	(not used)	N/A
EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Lower limit exceeded	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Upper limit exceeded	S

EV_ID3	SIG3	(not used)	N/A
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Parameter fault	S
EV_ID3	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Sensor limit exceeded	S
EV_ID3	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Invalid process value	S
EV_ID3	SIG7	(not used)	N/A
EV_ID3	SIG8	(not used)	N/A
EV_ID4	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Lower limit exceeded	S
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Upper limit exceeded	S
EV_ID4	SIG3	(not used)	N/A
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Parameter fault	S
EV_ID4	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Sensor limit exceeded	S
EV_ID4	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Invalid process value	S
EV_ID4	SIG7	(not used)	N/A
EV_ID4	SIG8	(not used)	N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..4	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..4	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..4	3	SLOT_NO	Slot number
EV_ID1	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.6 MOD_CPX6: Festo CPX CP Master Module

8.6.1 MOD_CPX6: Festo CPX CP Master Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.6.2 Description of MOD_CPX6

Object Name (Type + Number)

FB 806

Area of Application

MOD_CPX6 monitors a Festo CPX CP master module.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-CP-4-FB	CPI: 0 Byte I/0 Byte O CPI: 0 Byte I/4 Byte O CPI: 0 Byte I/8 Byte O CPI: 0 Byte I/12 Byte O CPI: 0 Byte I/16 Byte O CPI: 4 Byte I/0 Byte O CPI: 4 Byte I/4 Byte O CPI: 4 Byte I/8 Byte O CPI: 4 Byte I/12 Byte O CPI: 4 Byte I/16 Byte O CPI: 8 Byte I/0 Byte O CPI: 8 Byte I/4 Byte O CPI: 8 Byte I/8 Byte O CPI: 8 Byte I/12 Byte O CPI: 8 Byte I/16 Byte O CPI: 12 Byte I/0 Byte O CPI: 12 Byte I/4 Byte O CPI: 12 Byte I/8 Byte O CPI: 12 Byte I/12 Byte O CPI: 12 Byte I/16 Byte O CPI: 16 Byte I/0 Byte O CPI: 16 Byte I/4 Byte O CPI: 16 Byte I/8 Byte O CPI: 16 Byte I/12 Byte O CPI: 16 Byte I/16 Byte O	CP interface (master module)	No

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.

- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

CP modules are monitored line-wise. Up to four CP lines can be connected to the CP master module, each of which can consist of up to four modules. If an error is raised by at least one module of a line, the respective sub-collective fault is triggered by MOD_CPX6. If more than one CP module shows an error, several messages are triggered for the respective line at the same time.

For technical reasons, it is not possible to provide a (CP-) module or even channel specific diagnosis.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.6.3 I/Os of MOD_CPX6

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module

Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.6.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see	BOOL	FALSE	N/A	

	MSG_STAT)				
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.6.5 Message Texts And Associated Values of MOD_CPX6

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 1: Short circuit/overload	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 1: Undervoltage	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 1: CP module lost/fault	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 1: CP configuration failure	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 1: Short circuit CP line	S

EV_ID1	SIG6	(not used)	N/A
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 2: Short circuit/overload	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 2: Undervoltage	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 2: CP module lost/fault	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 2: CP configuration failure	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 2: Short circuit CP line	S
EV_ID2	SIG6	(not used)	N/A
EV_ID2	SIG7	(not used)	N/A
EV_ID2	SIG8	(not used)	N/A
EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 3: Short circuit/overload	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 3: Undervoltage	S
EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 3: CP module lost/fault	S
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 3: CP configuration failure	S
EV_ID3	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 3: Short circuit CP line	S
EV_ID3	SIG6	(not used)	N/A
EV_ID3	SIG7	(not used)	N/A
EV_ID3	SIG8	(not used)	N/A
EV_ID4	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 4: Short circuit/overload	S
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 4: Undervoltage	S
EV_ID4	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 4: CP module lost/fault	S
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 4: CP configuration failure	S
EV_ID4	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, CP line 4: Short circuit CP line	S
EV_ID4	SIG6	(not used)	N/A
EV_ID4	SIG7	(not used)	N/A
EV_ID4	SIG8	(not used)	N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description

EV_ID1..4	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..4	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..4	3	SLOT_NO	Slot number
EV_ID1	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.7 MOD_CPX7: Festo CPX Digital NAMUR Input Module

8.7.1 MOD_CPX7: Festo CPX Digital NAMUR Input Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.7.2 Description of MOD_CPX7

Object Name (Type + Number)

FB 807

- Interface Description

Area of Application

MOD_CPX7 monitors a Festo CPX digital NAMUR input module.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-P8DE-N	CPX-P-8DE-N [P8DI-N] CPX-P-8DE-N X [P8DI-N X]	8 channel digital NAMUR input module, for non EX signals	Yes
CPX-P8DE-N-IS	CPX-P-8DE-N-IS [P8DI-N-IS] CPX-P-8DE-N-IS X [P8DI-N-IS X]	8 channel digital NAMUR input module, for intrinsically safe signals	Yes

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.

- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initizalize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.7.3 I/Os of MOD_CPX7

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.7.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator

QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.7.5 Message Texts And Associated Values of MOD_CPX7

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 0: Short circuit/overload	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 0: Wire fracture/idling	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Lower limit exceeded	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Upper limit exceeded	S

EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 0: Parameter fault	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 0: Invalid process value (counting range exceeded)	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 1: Short circuit/overload	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 1: Wire fracture/idling	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 1: Lower limit exceeded	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 1: Upper limit exceeded	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 1: Parameter fault	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 1: Invalid process value (counting range exceeded)	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: ch. 2: Short circuit/overload	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 2: Wire fracture/idling	S
EV_ID2	SIG7	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 2: Lower limit exceeded	S
EV_ID2	SIG8	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 2: Upper limit exceeded	S
EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 2: Parameter fault	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 2: Invalid process value (counting range exceeded)	S
EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: ch. 3: Short circuit/overload	S
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 3: Wire fracture/idling	S
EV_ID3	SIG5	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 3: Lower limit exceeded	S
EV_ID3	SIG6	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 3: Upper limit exceeded	S
EV_ID3	SIG7	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 3: Parameter fault	S
EV_ID3	SIG8	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 3: Invalid process value (counting range exceeded)	S
EV_ID4	SIG1	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 4: Short circuit/overload	S
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 4: Wire fracture/idling	S
EV_ID4	SIG3	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 4: Parameter fault	S
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 5: Short circuit/overload	S
EV_ID4	SIG5	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: ch. 5: Wire fracture/idling	S
EV_ID4	SIG6	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 5: Parameter fault	S

EV_ID4	SIG7	(not used)	N/A
EV_ID4	SIG8	(not used)	N/A
EV_ID5	SIG1	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 6: Short circuit/overload	S
EV_ID5	SIG2	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 6: Wire fracture/idling	S
EV_ID5	SIG3	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 6: Parameter fault	S
EV_ID5	SIG4	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 7: Short circuit/overload	S
EV_ID5	SIG5	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: ch. 7: Wire fracture/idling	S
EV_ID5	SIG6	CPX Terminal @1%d@/@2%d@, Mod. @3%d@, ch. 7: Parameter fault	S
EV_ID5	SIG7	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: Module failure	S
EV_ID5	SIG8	CPX Terminal @1%d@/@2%d@, Mod. @3%d@: Unknown module error number @4%@	S

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..8	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..8	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..8	3	SLOT_NO	Slot number
EV_ID8	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.8 MOD_CPX8: Festo CPX CTEL Master Module

8.8.1 MOD_CPX8: Festo CPX CTEL Master Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.8.2 Description of MOD_CPX8

Object Name (Type + Number)

FB 808

Area of Application

MOD_CPX8 monitors a Festo CPX CTEL master module for I-port CPV and VTUG valve terminals.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/8 Byte O	CTEL interface (master module)	V2.2.0.3
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/16 Byte O	CTEL interface (master module)	V2.2.0.3
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/24 Byte O	CTEL interface (master module)	V2.2.0.3
CPX-CTEL-4M12-5POL	CTEL: 0 Byte I/32 Byte O	CTEL interface (master module)	V2.2.0.3

Note

The only supported configuration is for up to 4 I-port digital output devices with up to 32 channels each (e. g. CPV and VTUG valve terminals).

Please use the "Operation as pure output module" operating mode with 8 bytes I/Os (32 Channels per I- port). The DIL switches at the CPX-CTEL module need to be set accordingly, see figures below. Please refer to the manual "System description CTEL - Installation and commissioning of CTEL systems" (part no. 574601), ch. 1.5.1 for further details.

Please keep in mind to switch off the power supply before disassembling the CPX-CTEL module for setting the DIL switches! The module might be destroyed when removed under voltage!

1.5.1 DIL switch settings

1 DIL switch group 1

2 DIL switch group 2

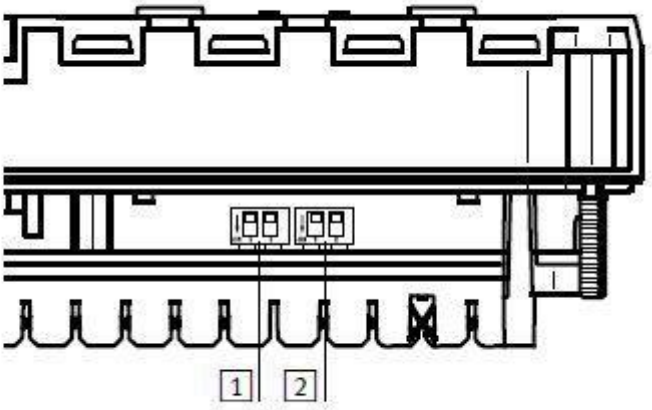

Fig. 1/7: Position of the DIL switches

DIL switch 1	S1.1	S1.2	Function
	OFF[1]	OFF[1]	Automatic I/O configuration[2]
	OFF	ON	Operation as pure output module
	ON	OFF	Operation as pure input module
	ON	ON	Mixed operation (Inputs and outputs)

[1] Standard setting
[2] In this operating mode, the DIL switch group 2 has no function

Tab. 1/5: Functions of DIL switch 1 (I/O mode)

DIL switch 2	S2.1	S2.2	Function
	OFF[1)]	OFF[1)]	8 bytes I/Os (2 bytes per I-port)
	OFF	ON	16 bytes I/Os (4 bytes per I-port)
	ON	OFF	24 bytes I/Os (6 bytes per I-port)
	ON	ON	32 bytes I/Os (8 bytes per I-port)
1) Standard setting			

Tab. 1/6: Functions of DIL switch 2 (I/O length)

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

CP modules are monitored line-wise. Up to four CP lines can be connected to the CP master module, each of which can consist of up to four modules. If an error is raised by at least one module of a line, the respective sub-collective fault is triggered by MOD_CPX6. If more than one CP module shows an error, several messages are triggered for the respective line at the same time.

For technical reasons, it is not possible to provide a (CP-) module or even channel specific diagnosis.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.8.3 I/Os of MOD_CPX8

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.8.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master	BOOL	FALSE		C

	system/Profinet IO system failure (1=error)				
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module	WORD	16#0	N/A	C

	Drivers" function will prevent channel and module diagnostics from working for this module!				

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.8.5 Message Texts And Associated Values of MOD_CPX8

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: General error	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: Short circuit	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: Wire fracture	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: Undervoltage in power supply	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: Device missing/failed	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: Configuration error	S
EV_ID1	SIG7	(not used)	N/A
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 1: Unknown error number @4%@	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: General error	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: Short circuit	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: Wire fracture	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: Undervoltage in power supply	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: Device missing/failed	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: Configuration error	S
EV_ID2	SIG7	(not used)	N/A
EV_ID2	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 2: Unknown error number @4%@	S

EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: General error	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: Short circuit	S
EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: Wire fracture	S
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: Undervoltage in power supply	S
EV_ID3	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: Device missing/failed	S
EV_ID3	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: Configuration error	S
EV_ID3	SIG7	(not used)	N/A
EV_ID3	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 3: Unknown error number @4%@	S
EV_ID4	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: General error	S
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: Short circuit	S
EV_ID4	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: Wire fracture	S
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: Undervoltage in power supply	S
EV_ID4	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: Device missing/failed	S
EV_ID4	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: Configuration error	S
EV_ID4	SIG7	(not used)	N/A
EV_ID4	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, I-port 4: Unknown error number @4%@	S
EV_ID5	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Undervoltage in power supply	S
EV_ID5	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Incorrect I/O length	S
EV_ID5	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Parameter error	S
EV_ID5	SIG4	(not used)	S
EV_ID5	SIG5	(not used)	S
EV_ID5	SIG6	(not used)	N/A
EV_ID5	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	N/A
EV_ID5	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..5	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to

EV_ID1..5	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..5	3	SLOT_NO	Slot number
EV_ID1..4	4	N/A	Channel (I-port specific) error code for port 1..4
EV_ID5	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.9 MOD_MPA: Festo MPA Valve Terminal Module

8.9.1 MOD_MPA: Festo MPA Valve Terminal Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.9.2 Description of MOD_MPA

Object Name (Type + Number)

FB 810

Area of Application

MOD_MPA monitors a Festo MPA valve terminal module installed in a CPX system.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
VMPA1-FB-EMS-8	MPA1S: VMPA1-FB-EMS-8 [8DO]	size 1, 8 valve coils, without electrical isolation	Yes
VMPA1-FB-EMS-D2-8	MPA1SD: VMPA1-FB-EMS-D2-8 [8DO]	size 1, 8 valve coils, without electrical isolation, diagnostics	Yes
VMPA1-FB-EMG-8	MPA1G: VMPA1-FB-EMG-8 [8DO]	size 1, 8 valve coils, with electrical isolation	Yes
VMPA1-FB-EMG-D2-8	MPA1GD: VMPA1-FB-EMG-D2-8 [8DO]	size 1, 8 valve coils, with electrical isolation, diagnostics	Yes
VMPA2-FB-EMS-4	MPA2S: VMPA2-FB-EMS-4 [4DO]x2 *MPA2S: VMPA2-FB-EMS-4 [4DO]x0	size 2, 4 valve coils, without electrical isolation	Yes
VMPA2-FB-EMS-D2-4	MPA2SD: VMPA2-FB-EMS-D2-4[4DO]x2 *MPA2SD:VMPA2-FB-EMS-D2-4[4DO]x0	size 2, 4 valve coils, without electrical isolation, diagnostics	Yes
VMPA2-FB-EMG-4	MPA2G: VMPA2-FB-EMG-4 [4DO]x2 *MPA2G: VMPA2-FB-EMG-4 [4DO]x0	size 2, 4 valve coils, with electrical isolation	Yes
VMPA2-FB-EMG-D2-4	MPA2GD:VMPA2-FB-EMG-D2-4 [4DO]x2 *MPA2GD:VMPA2-FB-EMG-D2-4[4DO]x0	size 2, 4 valve coils, with electrical isolation, diagnostics	Yes

Address stacking is supported, so the MPA size 2 modules marked with an asterisk (*) in the hardware catalogue can be used after their non-asterisk equivalents in order to fill up the binary addresses x.4 to x.7.

The following restrictions apply:

- Address stacking is only supported on Profibus DP. For CPX Terminals connected via Profinet IO, bits 4..7 of a byte are left unused when configuring four channel digital modules.
- For the "Generate Module Drivers" to recognize packed addresses correctly, please make sure to (in chronological order) first place the non-asterisk module for the lower four bits and then the asterisk module for the higher four bits. If you encounter errors or warnings when executing the "Generate Module Drivers" function referring to one or several modules using packed addresses, try to do the following:
  - Delete the respective (asterisk-)module in HWKonfig and save the configuration.
  - Re-insert the module again from the hardware catalogue (thus inserting the asterisk module chronologically **after** the non-asterisk module)
  - Save the configuration and restart the "Generate Module Drivers" function.

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initizalize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.9.3 I/Os of MOD_MPA

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.9.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related	WORD	16#0	X	V

	information from the terminal				
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.9.5 Message Texts And Associated Values of MOD_MPA

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Undervoltage in power supply	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Short circuit at valve	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Wire fracture (open load)	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Condition counter exceeded	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Short circuit at valve	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Wire fracture (open load)	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Condition counter exceeded	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Short circuit at valve	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Wire fracture (open load)	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Condition counter exceeded	S
EV_ID2	SIG5		N/A
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Short circuit at valve	S
EV_ID2	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Wire fracture (open load)	S
EV_ID2	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Condition counter exceeded	S
EV_ID3	SIG1		N/A
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Short circuit at valve	S

EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Wire fracture (open load)	S
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Condition counter exceeded	S
EV_ID3	SIG5		N/A
EV_ID3	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 5: Short circuit at valve	S
EV_ID3	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 5: Wire fracture (open load)	S
EV_ID3	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 5: Condition counter exceeded	S
EV_ID4	SIG1		N/A
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 6: Short circuit at valve	S
EV_ID4	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 6: Wire fracture (open load)	S
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 6: Condition counter exceeded	S
EV_ID4	SIG5		N/A
EV_ID4	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 7: Short circuit at valve	S
EV_ID4	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 7: Wire fracture (open load)	S
EV_ID4	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 7: Condition counter exceeded	S

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..4	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..4	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..4	3	SLOT_NO	Slot number
EV_ID2	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.10 MOD_VMPS: Festo VMPA-FB-PS Pressure Sensor Module

8.10.1 MOD_VMPS: Festo VMPA-FB-PS Pressure Sensor Module

This chapter contains the following information:

- Block Description

- Interface Description
- Message Texts And Associated Values

8.10.2 Description of MOD_VMPS

Object Name (Type + Number)

FB 811

Area of Application

MOD_VMPS monitors a Festo VMPA-FB-PS pressure sensor module mounted in an MPA type valve terminal.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
VMPA-FB-PS-1	VMPA-FB-PS [1AI-P]	MPA pressure sensor, monitoring pneum. channel 1	Yes
VMPA-FB-PS-3/5	VMPA-FB-PS [1AI-P]	MPA pressure sensor, monitoring pneum. channels 3 and 5	Yes
VMPA-FB-PS-P1	VMPA-FB-PS [1AI-P]	MPA pressure sensor, monitoring externally connected pressure	Yes
VMPAF-FB-EPL-PS	VMPAF-FB-EPL-PS [1AI-P]	MPAF pneumatic interface with pressure sensor	Yes

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a

possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.10.3 I/Os of MOD_VMPS

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.10.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V

RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to	WORD	16#0	N/A	C

	access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!				

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.10.5 Message Texts And Associated Values of MOD_VMPS

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Lower limit exceeded	S
EV_ID	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Upper limit exceeded	S
EV_ID	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Parameter fault	S
EV_ID	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID	SIG6	(not used)	N/A
EV_ID	SIG7	(not used)	N/A
EV_ID	SIG8	(not used)	N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1	3	SLOT_NO	Slot number
EV_ID1	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.11 MOD_VPPM: Festo VPPM Proportional Pressure Regulator Module

8.11.1 MOD_VPPM: Festo VPPM Proportional Pressure Regulator Module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.11.2 Description of MOD_VPPM

Object Name (Type + Number)

FB 812

Area of Application

MOD_VPPM monitors a Festo VPPM proportional pressure regulator module mounted in an MPA type valve terminal.

The proportional pressure regulator VPPM-6TA-... (VPPM/MPA) has been designed to regulate a pressure proportional to a specified setpoint value. An integrated pressure sensor records the pressure at the work connection and compares this with the setpoint value. If the actual value differs from the setpoint value, the pressure regulator is actuated further until the setpoint value is reached.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
VPPM-6TA-L-1-F-0L2H	VPPM 2 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..2 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L2H-S1	VPPM 2 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..2 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L2H-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..2 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L2H-S1-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..2 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H	VPPM 6 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..6 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H-S1	VPPM 6 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..6 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..6 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L6H-S1-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..6 bar, 1 % accuracy	Yes
VPPM-6TA-L-1-F-0L10H	VPPM 10 bar [1AI/1AO]	Pressure regulator module, size 6, output pressure 0..10 bar, 2 % accuracy	Yes
VPPM-6TA-L-	VPPM 10 bar [1AI/1AO]	Pressure regulator module, size 6,	Yes

1-F-0L10H-S1		output pressure 0..10 bar, 1 % accuracy	
VPPM-6TA-L-1-F-0L10H-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..10 bar, 2 % accuracy	Yes
VPPM-6TA-L-1-F-0L10H-S1-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 6, output pressure 0..10 bar, 1 % accuracy	Yes
VPPM-8TA-L-1-F-0L2H-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..2 bar, 2 % accuracy	Yes
VPPM-8TA-L-1-F-0L2H-S1-C1	VPPM Display 2 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..2 bar, 1 % accuracy	Yes
VPPM-8TA-L-1-F-0L6H-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..6 bar, 2 % accuracy	Yes
VPPM-8TA-L-1-F-0L6H-S1-C1	VPPM Display 6 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..6 bar, 1 % accuracy	Yes
VPPM-8TA-L-1-F-0L10H-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..10 bar, 2 % accuracy	Yes
VPPM-8TA-L-1-F-0L10H-S1-C1	VPPM Display 10 bar [1AI/1AO]	Pressure regulator module with display, size 8, output pressure 0..10 bar, 1 % accuracy	Yes

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initialize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start

channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.11.3 I/Os of MOD_VPPM

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.11.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V
TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		

EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	
DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the	WORD	16#0	N/A	C

	diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!				

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.11.5 Message Texts And Associated Values of MOD_VPPM

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: General fault	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Undervoltage in power supply	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Lower limit exceeded	S
EV_ID1	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Upper limit exceeded	S
EV_ID1	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Condition counter exceeded	S
EV_ID1	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Fault in parametrizing lower limit	S
EV_ID1	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Fault in parametrizing upper limit	S
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Wrong device type mounted	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Parameter fault	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Missing valve	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID2	SIG5	(not used)	N/A
EV_ID2	SIG6	(not used)	N/A

EV_ID2	SIG7	(not used)	N/A
EV_ID2	SIG8	(not used)	N/A

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..2	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..2	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..2	3	SLOT_NO	Slot number
EV_ID2	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

8.12 MOD_VTX: Festo non-MPA valve terminal module

8.12.1 MOD_VTX: Festo non-MPA valve terminal module

This chapter contains the following information:

- Block Description
- Interface Description
- Message Texts And Associated Values

8.12.2 Description of MOD_VTX

8.12.3 Object Name (Type + Number)

FB 819

Area of Application

MOD_VTX monitors a Festo non-MPA valve terminal module installed in a CPX system.

The following modules are supported:

Type	GSD/GSDML File Entries	Description	Release
VABA-1056-X1	**Until 08/2012:** ISO Plug-In DIL1 [8DO] ISO Plug-In DIL2 [16DO] ISO Plug-In DIL3 [24DO] ISO Plug-In DIL4 [32DO]#def **Since 09/2012:** VTSA Type44/45 DIL1 [8DO] VTSA Type44/45 DIL2 [16DO]	CPX pneumatic interface for VTSA pneumatics (ISO, type 44)	Yes

	VTSA Type44/45 DIL3 [24DO] VTSA Type44/45 DIL4 [32DO]#def		
VABA-1056-X2	**Until 08/2012:** ISO Plug-In DIL1 [8DO] ISO Plug-In DIL2 [16DO] ISO Plug-In DIL3 [24DO] ISO Plug-In DIL4 [32DO]#def **Since 09/2012:** VTSA Type44/45 DIL1 [8DO] VTSA Type44/45 DIL2 [16DO] VTSA Type44/45 DIL3 [24DO] VTSA Type44/45 DIL4 [32DO]#def	CPX pneumatic interface for VTSA pneumatics (ISO, type 44), metal version	Yes
VMPAL-EPL-CPX	VMPAL-EPL-CPX DIL1 [8DO] VMPAL-EPL-CPX DIL2 [16DO] VMPAL-EPL-CPX DIL3 [24DO] VMPAL-EPL-CPX DIL4 [32DO]#def	CPX pneumatics interface for MPA-L pneumatics	Yes
CPX-GP-03-4.0	Midi/Maxi DIL1 [8DO] Midi/Maxi DIL2 [16DO] Midi/Maxi DIL3 [24DO] Midi/Maxi DIL4 [32DO]#def	CPX pneumatic interface for Midi/Maxi (Type 03)	No
CPX-GP-CPA-10	CPA10/CPA14 DIL1 [8DO] CPA10/CPA14 DIL2 [16DO] CPA10/CPA14 DIL3 [24DO]#def	CPX pneumatic interface for CPA10	No
CPX-GP-CPA-14	CPA10/CPA14 DIL1 [8DO] CPA10/CPA14 DIL2 [16DO] CPA10/CPA14 DIL3 [24DO]#def	CPX pneumatic interface for CPA14	No

Calling OBs

The block is installed in the run sequence in the following OBs:

Type	Description
OB 1	Cyclic program
OB 100	Warm restart

Automatic Configuration via "Generate Module Drivers"

For each CPX module found in the hardware configuration, the following actions are executed automatically by the "**Generate Module Drivers**" function:

- The module driver block is installed in the run sequence after the superseding RACK_CPX block.
- The MOD_CNT value of the superseding RACK_CPX block is incremented.
- The CHNCNTx (x=1..47) I/O of the superseding RACK_CPX block is configured.
- The RACK_NO, SLOT_NO, SUBN_ID, EN_MSG and CHN_CFG I/Os are configured.
- The ACC_ID parameter is set to TRUE.
- The DIAG_IW, DIAG_QW, CHN_IDX and RACKF I/Os are interconnected with the superseding RACK_CPX block.

Manual Configuration

If the terminal's module configuration is changed manually, it needs to be taken care that all the I/Os specified above are set up correctly. This especially applies to MOD_CNT and CHNCNTx I/Os of the superseding RACK_CPX block, which need to be adjusted to the new channel configuration. Furthermore, ACC_ID needs to be set to TRUE to tell RACK_CPX to re-initizalize if changes are applied online.

Functional Description

The module driver block monitors the modules specified above and raises messages via ALARM_8P in case of

errors. It also generates channel-specific quality codes (using the OMODEx output parameters) that are passed to the signal processing blocks.

Monitoring is done as follows: As soon one or more module errors are detected by RACK_CPX, it sends a "start channel diag" flag to the DIAG_IW connector of the affected module driver block. The latter then subsequently requests error codes for all available channels, trigger ALARM_8P messages if applicable and sends a "finished channel diag" flag to RACK_CPX via the DIAG_QW connector. Depending on each channel's error status and a possible rack failure received from RACK_CPX, the module driver block sets the OMODEx output parameter accordingly.

Rack failure information is forwarded to QRACKF.

Redundancy

The CPX Terminal does not natively support redundant field bus connections. Consequently, neither does the module driver block. It may however be used behind a Y-Link interface. In this case, the subnet/DP address information displayed in error messages refer to the values for the DP master system the CPX terminal is connected to, i. e. the one behind the Y-Link.

Error Handling

Error handling only covers the proper execution of the ALARM_8P block used for sending messages to the visualization system. If ALARM_8P encounters errors during execution, this is signalled via the QERR connector. Detailed information can then be found via the MSG_STAT connector. Please check the ALARM_8P block documentation for details.

Startup Characteristics

Upon CPU startup (warm restart, via OB 100) or a TRUE signal at ACC_ID, the module driver block resets its internal sequencer for channel diagnosis and recalculates the channel configuration according to CHN_CFG.

Messaging Functionality

The module driver block only reports events that apply to the monitored CPX module. The block provides messages with channel specific error information. Several ALARM_8P instances are used for that purpose.

Messages are suppressed if a rack failure notification is received from RACK_CPX via the RACKF connector.

Message generation can also be suppressed by setting EN_MSG = FALSE.

Additional Information

For further information, please check the following sections:

- Interface Description
- Message Texts And Associated Values

8.12.4 I/Os of MOD_VTX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

8.12.5 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
SUBN_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to	WORD	16#FFFF		V
RACK_NO	Rack number (Profibus address) or device number (Profinet IO)	BYTE	16#0	X	V
SLOT_NO	Slot number	BYTE	16#0	X	V

TIMEOUT	Timeout for reply by CPX on diagnostic request (sec)	INT	2		
EV_IDx	Message ID for ALARM_8P instance x	DWORD	16#0	X	
EN_MSG	Enable messages	BOOL	TRUE		V
RACKF	From RACK_CPX: DP slave/IO device broken (1=error)	BOOL	FALSE		C
SUBN_ERR	From RACK_CPX: DP master system/Profinet IO system failure (1=error)	BOOL	FALSE		C
CHN_CFG	Number of input/output channels (high BYTE = input, low BYTE = output); needed for requesting channel related information from the terminal	WORD	16#0	X	V
CHN_IDX	Index of first input/output channel within terminal; calculated by RACK_CPX by adding the number of channels of all previous modules; needed for requesting channel related information from the terminal. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	INT	0	X	C
REF_NO	Reference number for identifying CPX module	INT	0	X	V
DIAG_IW	Diagnostic input word (from RACK_CPX block); the input word read from the diagnostic interface of the CPX terminal is forwarded by RACK_CPX as soon as the module driver block in question is allowed to access the interface. **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working!	WORD	16#0		C
MS	Maintenance state	DWORD	16#0		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QERR	1=program error (ALARM_8P failed, see MSG_STAT)	BOOL	FALSE	N/A	
QRACKF	1=DP slave/IO device broken	BOOL	FALSE	N/A	
O_MS	Maintenance state	DWORD	16#0	N/A	
CH_EXIST	Existing channels	DWORD	16#0	N/A	
CH_ACTIVE	Active channels	DWORD	16#0	N/A	
CH_OK	Channels without errors	DWORD	16#0	N/A	
EXT_STAT	Maintenance release - extended status	DWORD	16#0	N/A	
OMODEx	Status + MODE ch. x	DWORD	16#0	N/A	C
DXCHGx	Bidirectional data exchange ch. x	DWORD	16#0	N/A	C
MSG_ACKx	Acknowledge status for messages	WORD	16#0	N/A	
MSG_STATx	Error information of ALARM_8P	WORD	16#0	N/A	

DIAG_QW	Diagnostic output word (to RACK_CPX block); this output word is written to the diagnostic interface of the CPX terminal by RACK_CPX as soon as the module driver block in question is allowed to access the interface (decision by RACK_CPX). **CAUTION:** Removing the CFC connection created by the "Generate Module Drivers" function will prevent channel and module diagnostics from working for this module!	WORD	16#0	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
ACC_ID	1=parameters changed by driver generator, re-initialize	BOOL	FALSE		V

Additional Information

For further information, please check the following sections:

- Block Description
- Message Texts And Associated Values

8.12.6 Message Texts And Associated Values of MOD_VTX

Message Texts and Classes

ALARM_8P Instance	Message No.	Default Message Text	Message Class
EV_ID1	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Undervoltage in power supply	S
EV_ID1	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Module failure	S
EV_ID1	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@: Unknown module error number @4%@	S
EV_ID1	SIG4	(not used)	N/A
EV_ID1	SIG5	(not used)	N/A
EV_ID1	SIG6	(not used)	N/A
EV_ID1	SIG7	(not used)	N/A
EV_ID1	SIG8	(not used)	N/A
EV_ID2	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Short circuit at valve	S
EV_ID2	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 0: Wire fracture (open load)	S
EV_ID2	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Short circuit at valve	S
EV_ID2	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 1: Wire fracture (open load)	S
EV_ID2	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Short circuit at valve	S
EV_ID2	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 2: Wire fracture (open load)	S

EV_ID2	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Short circuit at valve	S
EV_ID2	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 3: Wire fracture (open load)	S
EV_ID3	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Short circuit at valve	S
EV_ID3	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 4: Wire fracture (open load)	S
EV_ID3	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 5: Short circuit at valve	S
EV_ID3	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 5: Wire fracture (open load)	S
EV_ID3	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 6: Short circuit at valve	S
EV_ID3	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 6: Wire fracture (open load)	S
EV_ID3	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 7: Short circuit at valve	S
EV_ID3	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 7: Wire fracture (open load)	S
EV_ID4	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 8: Short circuit at valve	S
EV_ID4	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 8: Wire fracture (open load)	S
EV_ID4	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 9: Short circuit at valve	S
EV_ID4	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 9: Wire fracture (open load)	S
EV_ID4	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 10: Short circuit at valve	S
EV_ID4	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 10: Wire fracture (open load)	S
EV_ID4	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 11: Short circuit at valve	S
EV_ID4	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 11: Wire fracture (open load)	S
EV_ID5	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 12: Short circuit at valve	S
EV_ID5	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 12: Wire fracture (open load)	S
EV_ID5	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 13: Short circuit at valve	S
EV_ID5	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 13: Wire fracture (open load)	S
EV_ID5	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 14: Short circuit at valve	S
EV_ID5	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 14: Wire fracture (open load)	S
EV_ID5	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 15: Short circuit at valve	S
EV_ID5	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 15: Wire fracture (open load)	S

EV_ID6	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 16: Short circuit at valve	S
EV_ID6	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 16: Wire fracture (open load)	S
EV_ID6	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 17: Short circuit at valve	S
EV_ID6	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 17: Wire fracture (open load)	S
EV_ID6	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 18: Short circuit at valve	S
EV_ID6	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 18: Wire fracture (open load)	S
EV_ID6	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 19: Short circuit at valve	S
EV_ID6	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 19: Wire fracture (open load)	S
EV_ID7	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 20: Short circuit at valve	S
EV_ID7	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 20: Wire fracture (open load)	S
EV_ID7	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 21: Short circuit at valve	S
EV_ID7	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 21: Wire fracture (open load)	S
EV_ID7	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 22: Short circuit at valve	S
EV_ID7	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 22: Wire fracture (open load)	S
EV_ID7	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 23: Short circuit at valve	S
EV_ID7	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 23: Wire fracture (open load)	S
EV_ID8	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 24: Short circuit at valve	S
EV_ID8	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 24: Wire fracture (open load)	S
EV_ID8	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 25: Short circuit at valve	S
EV_ID8	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 25: Wire fracture (open load)	S
EV_ID8	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 26: Short circuit at valve	S
EV_ID8	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 26: Wire fracture (open load)	S
EV_ID8	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 27: Short circuit at valve	S
EV_ID8	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 27: Wire fracture (open load)	S
EV_ID9	SIG1	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 28: Short circuit at valve	S
EV_ID9	SIG2	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 28: Wire fracture (open load)	S

EV_ID9	SIG3	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 29: Short circuit at valve	S
EV_ID9	SIG4	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 29: Wire fracture (open load)	S
EV_ID9	SIG5	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 30: Short circuit at valve	S
EV_ID9	SIG6	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 30: Wire fracture (open load)	S
EV_ID9	SIG7	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 31: Short circuit at valve	S
EV_ID9	SIG8	CPX Terminal @1%d@/@2%d@, mod. @3%d@, ch. 31: Wire fracture (open load)	S

Please check the Message Classes section for details on the different possible classes.

Associated Values

ALARM_8P Instance	Value No.	Block Parameter	Description
EV_ID1..9	1	SUBN1_ID	ID of DP master/sub system or Profinet IO system the CPX terminal is connected to
EV_ID1..9	2	RACK_NO	Rack number (Profibus address) or device number (Profinet IO)
EV_ID1..9	3	SLOT_NO	Slot number
EV_ID1	4	N/A	Module error code

Additional Information

For further information, please check the following sections:

- Block Description
- Interface Description

9 Channel Driver Blocks (Non APL)

9.1 CHAI_CPX: Festo CPX Analogue Input Channel

9.1.1 CHAI_CPX: Festo CPX Analogue Input Channel (Non APL)

This chapter contains the following information:

- Block Description
- Interface Description

9.1.2 Description of CHAI_CPX

Object Name (Type + Number)

FC 532

Area of Application

CHAI_CPX is used for any analogue input signal coming from a Festo CPX terminal. This includes CPX analogue input modules as well as pressure sensors or the feedback signals of proportional valves mounted in a valve terminal.

This driver block has been designed for usage in non APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The MODE parameter is interconnected with the appropriate OMODEx output of the respective module driver block.
- The CHLRANGE and CHHRANGE parameters are set to either the values defined in HWConfig ("Scale lowest value"/"Scale highest value") or predefined values, depending on the installed module and which input format has been chosen in HWConfig.
- The FORMAT parameter is set to the input format chosen in HWConfig or predefined values for certain modules.
- The INPTTYPE parameter is set to either the signal area chosen in HWConfig (e. g. "4..20 mA") or predefined values for certain modules.
- The BYTEORDR parameter is set to the value defined for the CPX terminal in HWConfig (right click the CPX terminal's icon in HWConfig -> Object Properties -> Parameter Assignment -> Device-specific parameters -> Byte order of analogue values).
- The VLRANGE and VHRANGE parameters are set up for pressure sensor signals to reflect the measurement range correctly. As for temperature sensor signals, the input word provided by the module then directly contains the actual physical value (e. g. pressure in mbar), in some cases with additional places. Scaling is then only applied by CPXAnIn to convert it to its base unit.

After a run of the "Generate Module Drivers" function, CHAI_CPX is fully adjusted to the settings made in HWConfig. Thus, the internal scaling algorithm fully adjusts itself to the settings made by the user. Please be aware that after changes are applied in HWConfig, the "Generate Module Drivers" function must be run in order to transfer the changed settings to the CHAI_CPX instances.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's MODE input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the QBAD output signal.

Additionally, the parameters that are normally set up by the "Generate Module Drivers" function automatically need to be configured according to the module used and the settings made in HWConfig (see above for details).

Finding the correct settings for the block's parameters manually is a complex task, and the risk of misconfigurations is high. Additionally, some block parameters might be protected against manual

modifications. It is therefore strongly suggested to use the "Generate Module Drivers" function and not to configure the channel driver block manually.

Functional Description

CHAI_CPX processes an analogue signal interconnected by the user from the process image (input word of the respective channel) to the VALUE parameter cyclically (depending on the insert point chosen by the user).

The block reads the raw data value and converts it to the corresponding physical value by scaling it between the lower and upper physical limits defined by the VLRANGE and VHRANGE parameters. Scaling is only applied if the analogue signal is provided as either voltage or current (e. g. 4..20 mA). For temperature and pressure values provided by the appropriate modules, sensors or proportional valves, scaling is not necessary, since those modules provide the actual physical value directly. In these cases, the VLRANGE and VHRANGE parameters are ignored.

The processed result is supplied at the V output parameter.

If the "Generate Module Drivers" function is used, the block is configured automatically to adjust the internal algorithms to the settings made in HWConfig (see above for details).

If the input format "comparator bit" is used for VPPM proportional valves, the process value will be displayed as 0.0 if the comparator bit is not set by the VPPM module and 1.0 if the comparator bit is set.

If the input format "process value + comparator bit" is used, the comparator bit is ignored by the channel driver block, and the current value is used as process value. This would provide the same result as using the "process value" setting.

Quality Code

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the MODE input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block.

The internal part is based on the setting of the SIM_ON, SUBS_ON and LAST_ON parameters (see table below).

The following quality codes are generated by the channel driver block:

Quality Code	Description	Condition
16#60	Simulation active	• SIM_ON = TRUE
16#80	Valid value	• SIM_ON = FALSE and • High byte of MODE input = 16#80
16#48	Substituted value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = TRUE
16#44	Last valid value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = FALSE and • LAST_ON = TRUE
16#00	Invalid value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = FALSE and • LAST_ON = FALSE

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states, while for the "Invalid value" state simulation, value substitution and keeping the last valid value must all be deactivated via their respective input parameters.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "IW 512") of the signal that is to be processed by the channel driver block must be interconnected to the VALUE parameter by the user.

Raw Value Check

Depending on the measurement type and range of the analog input module, the nominal range sets the range for converting analog signals into digital values (raw values). This includes an overshoot/undershoot range within which an analog signal can still be converted. Values outside this range constitute an overflow or underflow. The block indicates whether the raw value lies within the nominal range of the module.

Depending on its hardware revision, a CPX analogue input module might not provide overshoot and undershoot ranges. A 4..20 mA signal, for example, might be limited to either the lower physical limit (4 mA), the upper limit (20 mA), or both.

Additionally, undershoot and overshoot are not supported by temperature and pressure sensor modules, which provide the physical value directly and do not require additional scaling by CHAI_CPX.

QCHF_LL = TRUE indicates that the raw value is below the lower physical limit (undershoot), QCHF_HL = TRUE that it is above the upper physical limit (overshoot).

QBAD = TRUE (channel error) is also set if overflow or underflow is detected.

NAMUR Limit Check

By NAMUR definition, the current of a 4..20 mA loop is valid as long as 3.6 mA <= current <= 21 mA; if those limits are exceeded, the device is considered dead, and QBAD = TRUE is set. Alternatively, customized current limits may be defined via CH_F_HL and CH_F_LL and activated by setting CH_F_ON = TRUE.

NAMUR limit checks only apply to 4..20 mA signals. If CHAI_CPX is used for processing other signal formats, the check is omitted. Depending on the module type and its hardware revision, undershoot and/or overshoot might not be supported. In this case, the raw data value is limited to the nominal signal range (4 and/or 20 mA) by the module itself, rendering the NAMUR limit check ineffective.

NAMUR limit checks are processed in the AS and do not correlate with the analogue input module's internal limit monitoring. It is advised to deactivate the latter by leaving the "Channel x: Monitor lower (resp. upper) limit" parameters to their default disabled setting.

Normal Operation

During normal operation (valid value), the raw data value is converted to its physical value by scaling it between the limits defined by the VLRANGE and VHRANGE parameters (for voltage and current signals) or by forwarding the physical value received from the hardware (for temperature and pressure signals). The type of the signal connected to CHAI_CPX is defined by the INPTTYPE parameter, which is set by the "Generate Module Drivers" function automatically, based on the configuration done in HWConfig.

In detail, the following conversions are applied for the different types of input signals:

INPTTYPE	Type of Input Signal	Default Data Conversion	VLRANGE, VHRANGE
16#0	unknown/by DIL switch	VALUE is scaled between VLRANGE and VHRANGE	Used
16#1	0..10 V	VALUE is scaled between VLRANGE and VHRANGE	Used
16#2	+/- 10 V	VALUE is scaled between VLRANGE and VHRANGE	Used
16#3	+/- 5 V	VALUE is scaled between VLRANGE and VHRANGE	Used
16#4	1..5 V	VALUE is scaled between VLRANGE and VHRANGE	Used
16#5	0..20 mA	VALUE is scaled between VLRANGE and VHRANGE	Used
16#6	4..20 mA	VALUE is scaled between VLRANGE and VHRANGE	Used
16#7	+/-20 mA	VALUE is scaled between VLRANGE and VHRANGE	Used
16#8	0..10 V (w/o undershoot)	VALUE is scaled between VLRANGE and VHRANGE	Used
16#9	0..20 mA (w/o undershoot)	VALUE is scaled between VLRANGE and VHRANGE	Used
16#A	4..20 mA (w/o undershoot)	VALUE is scaled between VLRANGE and VHRANGE	Used
16#14	Pressure in mbar	VALUE is converted to bar (by dividing by 1000), VLRANGE and VHRANGE are calculated	Not used

		automatically	
16#15	Pressure in kPa	VALUE is used without scaling, VLRANGE and VHRANGE are calculated automatically	Not used
16#16	Pressure in 0.1 psi	VALUE is converted to psi (by dividing by 10), VLRANGE and VHRANGE are calculated automatically	Not used
16#1E	Temperature in 0.1 °C	VALUE is converted to °C (by dividing by 10), VLRANGE and VHRANGE are ignored	Not used
16#1F	Temperature in 0.1 °F	VALUE is converted to °F (by dividing by 10), VLRANGE and VHRANGE are ignored	Not used

As shown above, linear scaling is always applied for voltage and current input signals, with the scaling limits being defined by VLRANGE and VHRANGE. If, for example, VLRANGE is set to 0.0 and VHRANGE to 100.0, the input value is converted to a percentage. Or, for processing the value delivered by a 0..150 °C temperature sensor via 4..20 mA, VLRANGE needs to be set to 0.0 and VHRANGE to 150.0.

Pressure and temperature values are scaled to their base units (bar, kPa, psi, °C, °F). For pressure values, VHRANGE and VLRANGE are set up according to the channel's measurement range. However, they do not affect the internal calculation of CHAI_CPX in this case.

In order to adjust the internal scaling algorithm to the settings made in HWConfig, the data format (e. g. 12 bit right bound), byte order (Intel/Motorola) and low and high range limits of the raw data value are to be set up at the appropriate parameters. The "Generate Module Drivers" function fully takes care of that matter, such that the CHAI_CPX block is fully adjusted to the hardware configuration automatically. For that reason, it does not make a difference to the driver block which data format is applied. However, it is advised to stay with the default settings and just change the byte order to "Motorola" in the CPX terminal's system parameters in HWConfig for easier debugging (due to the swapped bytes, the default "Intel" setting does not allow for convenient monitoring of the raw value in CFC test mode).

The lower and upper limit parameters (VLRANGE/VHRANGE) are copied to the outputs OVLRANGE and OVHRANGE to allow the interconnection of the settings to other block I/Os.

During normal operation (valid value), QUALITY is set to 16#80 and QBAD to FALSE.

For more information about the parameters mentioned above, please also check the Interface Description for CHAI_CPX.

Simulation Mode

Simulation can be activated by setting the SIM_ON = TRUE. The value provided at SIM_V is then written to the V output parameter instead of processing the raw value VALUE. QUALITY is set to 16#60, QBAD is set to FALSE, while QSIM = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SIM_ON is set to TRUE, any other value is ignored. This also affects the substitute and hold last value values, which are not effective when simulation is active.

Substitute Value

If SUBS_ON = TRUE and the value read from the process image is invalid, QBAD = TRUE is set and the value defined at or interconnected to SUBS_V is written to the V output parameter. QUALITY is set to 16#48, QSUBS = TRUE states that the value currently provided at V is substituted.

Hold Last Value

If LAST_ON = TRUE, SUBS_ON = FALSE and the value read from the process image is invalid (MODE high byte <> 16#80), QBAD = TRUE is set and the last valid value is held at the V output (i. e. the value that was effective before it was detected that the process data has become invalid). QUALITY is set to 16#44. The last valid value is additionally provided at the V_LAST IN_OUT parameter, while the QLAST = TRUE states that the value currently provided at V is the (held) last valid value.

Invalid Value

If both SUBS_ON and LAST_ON are set to FALSE and the value read from the process image is indicated as invalid (MODE high byte <> 16#80), the result of the scaling is still written to the V output parameter.

Exception: If CHAI_CPX itself detects an error during the scaling (broken wire or input overload), the result of the scaling is ignored, and the process value's low limit value (VLRANGE) is written to V. This only applies if a 2AE-U-I or 4AE-I module with data format "12bit left bound + diagnosis" or a CPX-4AE-U-I with "sign+15 bit" data format (FORMAT = 3 or 4) is used. Other data formats and/or modules do not provide broken wire information within the raw data value.

In both cases, QBAD = TRUE indicates the invalid value, and QUALITY is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at MODE, QMOD_ERR = TRUE is set.

Error Handling

The plausibility of input parameters is not checked.

Startup Characteristics

N/A

Time Response

N/A

Message Response

N/A

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

9.1.3 I/Os of CHAI_CPX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

9.1.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
CHHRANGE	High range of raw value: "Scale highest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
CHLRANGE	Low range of raw value: "Scale lowest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
BYTEORDR	Byte order (0=Intel, 1=Motorola): As set up in HWConfig in the terminal's (not module's!) properties, "Byte order of analogue values"	BOOL	FALSE	X	V
FORMAT	Input format ("Input format" or "Input signal type" in HWConfig, depending on the module type). Valid values: • 16#0=15bit linear scaled • 16#1=12 bit right bound	BYTE	0	X	V

	• 16#2=15bit • 16#3=12bit left bound + diagnosis • 16#4=sign+15 bit (ET200 compatible) • 16#A=process value (no scaling required) • 16#B=comparator bit • 16#C=process value + comparator bit				
INPTTYPE	Input signal type ("Signal area", "Measuring adjustment", "Dimension unit" or "Input unit" in HWConfig, depending on the module type). Valid values: • 16#0=no sensor connected/range set via DIL switch/linear scaled (only 4AE-P) • 16#1=0..10V • 16#2=+/- 10V • 16#3=+/- 5V • 16#4=1..5V • 16#5=0..20mA • 16#6=4..20mA • 16#7=+/-20mA • 16#8=0..10V (without undershoot range) • 16#9=0..20mA (without undershoot range) • 16#A=4..20mA (without undershoot range) • 16#14=pressure in mbar • 16#15=pressure in kPa • 16#16=pressure in 0,1 psi • 16#1E=temperature in 0.1 deg C • 16#1F=temperature in 0.1 deg F	BYTE	0	X	V

If the input format "comparator bit" is used for VPPM proportional valves, the process value will be displayed as 0.0 if the comparator bit is not set by the VPPM module and 1.0 if the comparator bit is set.

If the input format "process value + comparator bit" is used, the comparator bit is ignored by the channel driver block, and the current value is used as process value. This would provide the same result as using the "process value" setting.

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QBAD	1=bad process value: (Quality information in MODE high byte <> 16#80 or broken wire detected or signal out of NAMUR or defined range) and simulation not active	BOOL	FALSE	N/A	

QCHF_HL	1=process value high limit failure: VALUE < CHHRANGE	BOOL	FALSE	N/A	
QCHF_LL	1=process value low limit failure: VALUE < CHLRANGE	BOOL	FALSE	N/A	
QSIM	1=simulation active: SIM_ON = TRUE	BOOL	FALSE	N/A	
QSUBS	1=failure substitution active: Invalid value and SUBS_ON = TRUE	BOOL	FALSE	N/A	
QLAST	1=failure last value active: Invalid value, SUBS_ON = FALSE and LAST_ON = TRUE	BOOL	FALSE	N/A	
QMOD_ERR	1=higher level error: Quality information in MODE high byte = 16#40	BOOL	FALSE	N/A	
OVHRANGE	High range of process value (copy)	REAL	0.0	N/A	
OVLRANGE	Low range of process value (copy)	REAL	0.0	N/A	
V	Process value	REAL	0.0	N/A	
QUALITY	Quality code	BYTE	16#0	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
VALUE	Input/periphery value	WORD	16#0		
VHRANGE	High limit of scaled process value	REAL	100.0		(V)*
VLRANGE	Low limit of scaled process value	REAL	0.0		(V)*
V_LAST	Last valid process value	REAL	0.0		
CH_F_ON	1=limit failure active	BOOL	FALSE		
CH_F_HL	High limit of input value (in mA)	REAL	21.5		
CH_F_LL	Low limit of input value (in mA)	REAL	3.3		
SIM_ON	1=activate simulation	BOOL	FALSE		
SIM_V	Simulation value	REAL	0.0		
LAST_ON	1=enable last valid value	BOOL	FALSE		
SUBS_ON	1=enable failure substitution	BOOL	FALSE		
SUBS_V	Substitution value	REAL	0.0		

* If the analogue signal processed by the channel driver block is a pressure value, the VLRANGE and VHRANGE parameters are set up to reflect the measurement range of the module in question. In both cases as well as for temperature values, the input word provided by the input module directly contains the actual physical value (e. g. in °C), in some cases with additional decimal places. Scaling is then only applied by CHAI_CPX to convert it to its base unit.

Additional Information

For further information, please check the following sections:

- Block Description

9.2 CHAO_CPX: Festo CPX Analogue Output Channel

9.2.1 CHAO_CPX: Festo CPX Analogue Output Channel (Non APL)

This chapter contains the following information:

- Block Description
- Interface Description

9.2.2 Description of CHAO_CPX

Object Name (Type + Number)

FC 533

Area of Application

CHAO_CPX is used for any analogue input signal written to a Festo CPX terminal. This includes CPX analogue output modules as well as set point values for proportional valves mounted in a valve terminal.

This driver block has been designed for usage in non APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The MODE parameter is interconnected with the appropriate OMODEx output of the respective module driver block.
- The CHLRANGE and CHHRANGE parameters are set to either the values defined in HWConfig ("Scale lowest value"/"Scale highest value") or predefined values, depending on the installed module and which input format has been chosen in HWConfig.
- The BYTEORDR parameter is set to the value defined for the CPX terminal in HWConfig (right click the CPX terminal's icon in HWConfig -> Object Properties -> Parameter Assignment -> Device-specific parameters -> Byte order of analogue values).

After a run of the "Generate Module Drivers" function, CHAO_CPX is fully adjusted to the settings made in HWConfig. Thus, the internal scaling algorithm fully adjusts itself to the settings made by the user. Please be aware that after changes are applied in HWConfig, the "Generate Module Drivers" function must be run in order to transfer the changed settings to the CHAO_CPX instances.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's MODE input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the QBAD output signal.

Additionally, the parameters that are normally set up by the "Generate Module Drivers" function automatically need to be configured according to the module used and the settings made in HWConfig (see above for details).

Finding the correct settings for the block's parameters manually is a complex task, and the risk of misconfigurations is high. Additionally, some block parameters might be protected against manual modifications. It is therefore strongly suggested to use the "Generate Module Drivers" function and not to configure the channel driver block manually.

Functional Description

CHAO_CPX processes an analogue set point value interconnected to the U parameter cyclically (depending on the insert point chosen by the user) and provides a suitable raw value at the VALUE output parameter, which is to be interconnected to the process image (output word of the respective channel) by the user. Linear scaling is applied in order to do the conversion from process to periphery format.

If the "Generate Module Drivers" function is used, the block is configured automatically to adjust the internal algorithms to the settings made in HWConfig (see above for details).

Quality Code

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the MODE input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block.

The internal part is based on the setting of the SIM_ON, SUBS_ON and LAST_ON parameters (see table below).

The following quality codes are generated by the channel driver block:

Quality Code	Description	Condition
16#60	Simulation active	• SIM_ON = TRUE
16#56	High limit exceeded	• SIM_ON = FALSE and • U > UHRANGE
16#55	Low limit exceeded	• SIM_ON = FALSE and • U < ULRANGE
16#80	Valid value	• SIM_ON = FALSE and • High byte of MODE input = 16#80 and • ULRANGE <= U <= UHRANGE
16#00	Invalid value	• SIM_ON = FALSE and • invalid value

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "QW 512") of the signal that is to be processed by the channel driver block must be interconnected to the VALUE parameter by the user.

Normal Operation

During normal operation (valid value), the process value is converted to the raw value by scaling it between the limits between the process value limits ULRANGE/UHRANGE and the raw data limits CHLRANGE/CHHRANGE.

In contrast to the analogue input driver block CHAI_CPX, scaling is always applied by CHAO_CPX, and the ULRANGE and UHRANGE parameters are set up by default to interpret the value provided at U as percentage. If the set point value for a VPPM proportional value is to be interpreted as physical value, ULRANGE and UHRANGE need to be set to the appropriate physical limits of the hardware (e. g. 0.0/6000.0 or 0.0/6.0 for a 0..6 bar VPPM valve, depending on whether the value is provided as millibars or bars by the PCS7 application).

In order to adjust the internal scaling algorithm to the settings made in HWConfig, the byte order (Intel/Motorola) and low and high range limits of the raw data value are to be set up at the appropriate parameters. The "Generate Module Drivers" function fully takes care of that matter, such that the CHAO_CPX block is fully adjusted to the hardware configuration automatically. For that reason, it does not make a difference to the driver block which byte order is applied. However, it is advised to change the byte order to "Motorola" in the CPX terminal's system parameters in HWConfig for easier debugging (due to the swapped bytes, the default "Intel" setting does not allow for convenient monitoring of the raw value in CFC test mode).

The lower and upper limit parameters (ULRANGE/UHRANGE) are copied to the outputs OVLRANGE and OVHRANGE to allow the interconnection to other block I/Os.

During normal operation (valid value), QUALITY is set to 16#80 and QBAD to FALSE.

For more information about the parameters mentioned above, please also check the Interface Description for CHAO_CPX.

Limiting Of The Process Value

The CPX/MPA hardware does NOT support an under- or overshoot for the periphery signal; the set point value is therefore always limited to the values given at the ULRANGE/UHRANGE parameters, before scaling is applied. The QCHF_LL and QCHF_HL output parameters indicate whether the process value has been limited. If, for example, a CHAO_CPX block is configured with ULRANGE=0.0 and UHRANGE=6.0, a process value U=6.5 would be treated as 6.0 internally, and QCHF_HL = TRUE would be set.

Additional limits can be applied if LIMIT_ON = TRUE is set. Now, V_LL and V_HL are effective in addition to ULRANGE and UHRANGE. But in contrast to the latter, the former do not take influence on the scaling algorithm itself, but only apply the following limits to the process value U:

- U > V_HL: U is limited to V_HL
- U < V_LL: U is limited to LL_V (not V_LL!)

Activation of those limits is not indicated at the block's interface.

An example might be a VPPM proportional valve with physical limits of 0..6 bars. So, ULRANGE = 0.0 and

UHRANGE = 6.0 would allow for providing the process value U in bars from the PCS7 application. If now the actual pressure at the valve's outlet connector would need to be limited to 5 bars, LIMIT_ON = TRUE and V_HL = 5.0 would allow for exactly that. If UHRANGE was used instead, the scaling would be affected such that U = 5.0 would then make CHAO_CPX deliver a raw value telling the proportional valve to go for the maximum, i. e. 6 bars (instead of the desired 5 bars).

It is important to keep in mind the difference between the two parameter sets that form limits for the process value U. ULRANGE/UHRANGE limit, indicate active limits via QCHF_LL/QCHF_HL and affect the internal scaling algorithm. V_LL/V_HL need to be activated via LIMIT_ON and - once activated - do not affect the scaling itself, but only provide "silent" limiting without indication at the block's interface.

Simulation Mode

Simulation can be activated by setting the SIM_ON = TRUE. The value provided at SIM_U is then processed instead of the process U, and the result is written as raw data value to the VALUE output parameter. QUALITY is set to 16#60, QBAD is set to FALSE, while QSIM = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SIM_ON is set to TRUE, any other value is ignored. This also affects the substitute and hold last value values, which are not effective when simulation is active.

I/O Fault

Even if a module error is signalled via the MODE parameter's high byte, the value provided at the U input parameter is always written to the VALUE output parameter.

QBAD = TRUE indicates the invalid value, and QUALITY is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at MODE, QMOD_ERR = TRUE is set.

Error Handling

The plausibility of input parameters is not checked.

Startup Characteristics

N/A

Time Response

N/A

Message Response

N/A

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

9.2.3 I/Os of CHAO_CPX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

9.2.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
CHHRANGE	High range of raw value: "Scale highest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
CHLRANGE	Low range of raw value: "Scale lowest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
BYTEORDR	Byte order (0=Intel, 1=Motorola): As set up in HWConfig in the terminal's (not module's!) properties, "Byte order of analogue values"	BOOL	FALSE	X	V

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QCHF_HL	1=process value high limit failure: U > UHRANGE	BOOL	FALSE	N/A	
QCHF_LL	1=process value low limit failure: U < ULRANGE	BOOL	FALSE	N/A	
QBAD	1=bad process value: Quality information in MODE high byte <> 16#80 and simulation not active	BOOL	FALSE	N/A	
QSIM	1=simulation active: SIM_ON = TRUE	BOOL	FALSE	N/A	
QMOD_ERR	1=higher level error: Quality information in MODE high byte = 16#40	BOOL	FALSE	N/A	
OVHRANGE	High range of process value (copy)	REAL	0.0	N/A	
OVLRANGE	Low range of process value (copy)	REAL	0.0	N/A	
VALUE	Periphery output (raw) value	WORD	16#0	N/A	
QUALITY	Quality code	BYTE	16#0	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
U	Setpoint value	REAL	0.0		
UHRANGE	High range of process value	REAL	100.0		
ULRANGE	Low range of process value	REAL	0.0		
LIMIT_ON	1=enable limiting of the process value: V_HL/V_LL limit monitoring	BOOL	FALSE		
V_HL	High limit	REAL	100.0		
V_LL	Low Limit	REAL	0.0		
LL_V	Process value, if U < V_LL	REAL	0.0		
SIM_ON	1=activate simulation	BOOL	FALSE		

SIM_U	Simulation value	REAL	0.0		

Additional Information

For further information, please check the following sections:

- Block Description

9.3 CHCN1CPX: Festo CPX-P Counter/Frequency Channel

9.3.1 CHCN1CPX: Festo CPX counter type 1 channel driver block (Non APL)

This chapter contains the following information:

- Block Description
- Interface Description

9.3.2 Description of CHCN1CPX

Object Name (Type + Number)

FC 534

Area of Application

CHCN1CPX is used for reading counter or frequency measurement values from a CPX-P-8DE-N... module. It furthermore controls the function of the respective channel via triggering its corresponding output signals.

Please note that if a channel is not configured as counter or frequency measurement but as NAMUR digital input instead, the CHDI_CPX channel driver block needs to be used instead.

This driver block has been designed for usage in non APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The MODE parameter is interconnected with the appropriate OMODEx output of the respective module driver block.
- The INPTTYPE parameter is set to the signal type chosen in HWConfig (e. g. counter value).
- The BYTEORDR parameter is set to the value defined for the CPX terminal in HWConfig (right click the CPX terminal's icon in HWConfig -> Object Properties -> Parameter Assignment -> Device-specific parameters -> Byte order of analogue values).
- The EDGE_POL parameter is set according to the edge polarity (positive/negative) set up in HWConfig.
- The CNT_DIR parameter is set according to the counting direction (up/down) set up in HWConfig.
- The STATE parameter is interconnected to the "State" digital input signal (limit exceeded yes/no) for the respective channel.
- The DIAG parameter is interconnected to the "Diag" digital input signal (channel error yes/no) for the respective channel.
- The QSTART parameter is interconnected to the "Start/Stop" digital output signal (start counting/frequency measurement yes/no) for the respective channel.
- The QRESET parameter is interconnected to the "Reset" digital output signal (reset counter yes/no) for the respective channel.

After a run of the "Generate Module Drivers" function, CHCN1CPX is fully adjusted to the settings made in HWConfig. Thus, the internal algorithm fully adjusts itself to the settings made by the user. Please be aware that after changes are applied in HWConfig, the "Generate Module Drivers" function must be run in order to transfer the changed settings to the CHCN1CPX instances.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's MODE input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the QBAD output signal.

Additionally, the parameters that are normally set up or interconnected by the "Generate Module Drivers" function automatically need to be configured according to the module used and the settings made in HWConfig (see above for details).

Finding the correct settings for the block's parameters manually is a complex task, and the risk of misconfigurations is high. Additionally, some block parameters might be protected against manual modifications. It is therefore strongly suggested to use the "Generate Module Drivers" function and not to configure the channel driver block manually.

Functional Description

CHCN1CPX processes a counter or frequency measurement signal interconnected by the user from the process image (input word of the respective channel) to the VALUE parameter cyclically (depending on the insert point chosen by the user).

When used for a channel configured as counter, the raw data (counter value) is directly forwarded to the V output parameter. If the block is used for a frequency measurement signal, the raw data value is scaled using the F_FACT parameter. Thus, for example, the frequency value supplied in Hz by the hardware can conveniently be converted into rounds per minute; the default setting for F_FACT is 1. The processed result is supplied at the V output parameter.

If the "Generate Module Drivers" function is used, the block is configured automatically to adjust the internal algorithms to the settings made in HWConfig (see above for details).

Quality Code

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the MODE input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block. Additionally, the status of the DIAG parameter (interconnected to the corresponding input signal by the "Generate Module Drivers" function, see above for details) is considered.

The internal part is based on the setting of the SIM_ON, SUBS_ON and LAST_ON parameters (see table below).

The following quality codes are generated by the channel driver block:

Quality Code	Description	Condition
16#60	Simulation active	• SIM_ON = TRUE
16#80	Valid value	• SIM_ON = FALSE and • High byte of MODE input = 16#80
16#48	Substituted value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = TRUE
16#44	Last valid value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = FALSE and • LAST_ON = TRUE
16#00	Invalid value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = FALSE and • LAST_ON = FALSE

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states, while for the "Invalid value" state simulation, value substitution and keeping the last valid value must all be deactivated via their respective input parameters.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "IW 512") of the counter or frequency measurement value that is to be processed by the channel driver block must be interconnected to the VALUE parameter by the user.

Limit Check

Both counter and frequency measurement values are monitored upon limits by the hardware. The digital input signal interconnected to the STATE parameter by the "Generate Module Drivers" function provides information about whether the respective value is in range or not. It is forwarded to QSTATE output parameter and thus can be interconnected to other block I/Os. If QSTATE = TRUE, either the lower or upper limit set up in HWConfig has been exceeded.

Suitable messages are generated by the MOD_CPX7 module driver block in both cases (provided that the "Generate Module Drivers" function has been executed before).

QBAD = TRUE (channel error) is set in case of broken wire or shortcut.

Normal Operation

During normal operation (valid value), the raw data value received from the hardware is either scaled (for frequency measurement values) or forwarded directly (for counter values). The type of signal connected to CHCN1CPX is defined by the INPTTYPE parameter, which is set by the "Generate Module Drivers" function automatically, based on the configuration done in HWConfig.

In detail, the following conversions are applied for the different types of input signals:

INPTTYPE	Type of Input Signal	Default Data Conversion
16#0	NAMUR digital input	N/A, use CHDI_CPX digital input driver block instead
16#28	Counter value	None, VALUE is used without conversion
16#29	Frequency measurement	VALUE is scaled using F_FACT

As shown above, scaling is applied for frequency measurement values only. Thus, for example, the speed of an agitator can be easily calculated in rounds per minute. Counter values are always transferred without any data conversion.

In order to adjust the internal scaling algorithm to the settings made in HWConfig, the byte order (Intel/Motorola) and input type setting are to be set up at the appropriate parameters. The "Generate Module Drivers" function fully takes care of that matter, such that the CHCN1CPX block is fully adjusted to the hardware configuration automatically. For that reason, it does not make a difference to the driver block which data format is applied. However, it is advised to change the byte order to "Motorola"in the CPX terminal's system parameters in HWConfig for easier debugging (due to the swapped bytes, the default "Intel" setting does not allow for convenient monitoring of the raw value in CFC test mode).

The CNT_DIR and EDGE_POL parameters are copied to the outputs QCNT_DIR and QEDGEPOL to allow for the interconnection of the settings to other block I/Os.

The START and RESET parameters are used for controlling the counter or frequency measurement. Both parameters are forwarded to the QSTART and QRESET output parameters, which are interconnected to the appropriate hardware output signals by the "Generate Module Drivers" function and thus control the behaviour of the respective channel.

If START = TRUE, counting or frequency measurement is activated, and the appropriate value is supplied by the hardware at the input word interconnected to the VALUE input parameter. START = FALSE freezes the counter value or sets the frequency measurement value to 0.

As long as RESET = TRUE, the counter value is reset to either the lower limit set up in HWConfig (when counting up, CNT_DIR = FALSE) or to the upper limit (when counting down, CNT_DIR = TRUE). The RESET value does have no effect if frequency measurement is used. Please see the module's hardware documentation for further details.

During normal operation (valid value), QUALITY is set to 16#80 and QBAD to FALSE.

For more information about the parameters mentioned above, please also check the Interface Description for CHCN1CPX.

Simulation Mode

Simulation can be activated by setting the SIM_ON = TRUE. The value provided at SIM_V is then (directly, i. e. without using the F_FACT parameter for frequency measurement) written to the V output parameter instead of processing the raw value VALUE. QUALITY is set to 16#60, QBAD is set to FALSE, while QSIM = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SIM_ON is set to TRUE, any other value is ignored. This also affects the substitute and hold last value values, which are not effective when simulation is active.

Substitute Value

If SUBS_ON = TRUE and the value read from the process image is invalid, QBAD = TRUE is set and the value defined at or interconnected to SUBS_V is written to the V output parameter. QUALITY is set to 16#48, QSUBS = TRUE states that the value currently provided at V is substituted.

Hold Last Value

If LAST_ON = TRUE, SUBS_ON = FALSE and the value read from the process image is invalid (MODE high byte <> 16#80 or DIAG = TRUE), QBAD = TRUE is set and the last valid value is held at the V output (i. e. the value that was effective before it was detected that the process data has become invalid). QUALITY is set to 16#44. The last valid value is additionally provided at the V_LAST IN_OUT parameter, while the QLAST = TRUE states that the value currently provided at V is the (held) last valid value.

Invalid Value

If both SUBS_ON and LAST_ON are set to FALSE and the value read from the process image is indicated as invalid (MODE high byte <> 16#80 or DIAG = TRUE), the same value as during normal operation is still written to the V output parameter.

QBAD = TRUE indicates the invalid value, and QUALITY is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at MODE, QMOD_ERR = TRUE is set.

Error Handling

The plausibility of input parameters is not checked.

Startup Characteristics

N/A

Time Response

N/A

Message Response

N/A

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

9.3.3 I/Os of CHCN1CPX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

9.3.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator

MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
BYTEORDR	Byte order (0=Intel, 1=Motorola): As set up in HWConfig in the terminal's (not module's!) properties, "Byte order of analogue values"	BOOL	FALSE	X	V
INPTTYPE	Input signal type ("Signal area", "Measuring adjustment", "Dimension unit" or "Input unit" in HWConfig, depending on the module type). Valid values: • 16#0=digital input (NAMUR) • 16#28=counter value • 16#29=frequency in Hz	BYTE	0	X	V
CNT_DIR	Counting direction (0=up; 1=down)	BOOL	FALSE	X	V
EDGE_POL	Edge polarity (0=rising, 1=falling)	BOOL	FALSE	X	V

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QBAD	1=bad process value: (Quality information in MODE high byte <> 16#80 or VALUE_QC = FALSE and external quality information is activated via PQC = TRUE) and simulation not active	BOOL	FALSE	N/A	
QSTATE	1=limit exceeded	BOOL	FALSE	N/A	
QCNT_DIR	Counting direction (0=up; 1=down)	BOOL	FALSE	N/A	
QEDGEPOL	Edge polarity (0=rising, 1=falling)	BOOL	FALSE	N/A	
QSIM	1=simulation active: SIM_ON = TRUE	BOOL	FALSE	N/A	
QSUBS	1=failure substitution active: Invalid value and SUBS_ON = TRUE	BOOL	FALSE	N/A	
QLAST	1=failure last value active: Invalid value, SUBS_ON = FALSE and LAST_ON = TRUE	BOOL	FALSE	N/A	
QMOD_ERR	1=higher level error: Quality information in MODE high byte = 16#40	BOOL	FALSE	N/A	
QSTART	Start/stop command to hardware (0=stop, 1=start counter/measurement)	BOOL	FALSE	N/A	C
QRESET	Reset command to hardware (0=counter active, 1=reset counter)	BOOL	FALSE	N/A	C
V	Process value	BOOL	FALSE	N/A	
QUALITY	Quality code	BYTE	16#0	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
VALUE	Input/periphery value	WORD	16#0		
STATE	Channel state (0=limits not exceeded, 1=limit exceeded)	BOOL	FALSE		C
DIAG	Channel diagnostics (0=ok, 1=channel error)	BOOL	FALSE		C

START	Start/stop command (0=stop, 1=start counter/measurement)	BOOL	FALSE		
RESET	Reset command (0=counter active, 1=reset counter)	BOOL	FALSE		
F_FACT	Factor for scaling of frequency measurement (V = VALUE * F_FACT)	REAL	1.0		
V_LAST	Last valid process value	REAL	0.0		
SIM_ON	1=activate simulation	BOOL	FALSE		
SIM_V	Simulation value	REAL	0.0		
LAST_ON	1=enable last valid value	BOOL	FALSE		
SUBS_ON	1=enable failure substitution	BOOL	FALSE		
SUBS_V	Substitution value	REAL	0.0		

Additional Information

For further information, please check the following sections:

- Block Description

9.4 CHDI_CPX: Festo CPX Digital Input Channel

9.4.1 CHDI_CPX: Festo CPX Digital Input Channel (Non APL)

This chapter contains the following information:

- Block Description
- Interface Description

9.4.2 Description of CHDI_CPX

Object Name (Type + Number)

FC 530

Area of Application

CHDI_CPX is used for any digital input signal coming from a Festo CPX terminal. This includes CPX digital input modules as well as modules connected via CPI to a CPX terminal.

This driver block has been designed for usage in non APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions a executed automatically by the "**Generate Module Drivers**" function:

- The MODE parameter is interconnected with the appropriate OMODEx output of the respective module driver block.
- If the hardware provides quality information via a digital input signal, the PQC parameter is set to TRUE and VALUE_QC is interconnected with the hardware quality bit.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's MODE input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the QBAD output signal.

Functional Description

CHDI_CPX processes a digital signal interconnected by the user from the process image to the VALUE parameter cyclically (depending on the insert point chosen by the user). The processed result is supplied at the Q output parameter.

Quality Code

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the MODE input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block. An external binary (good/bad) quality signal can be interconnected to the VALUE_QC input and activated by setting PQC to TRUE. In this case, both information parts need to show a "good" value (logical AND) for the block to consider the value as such.

The internal part is based on the setting of the SIM_ON and SUBS_ON parameters (see table below).

The following quality codes are generated by the channel driver block:

Quality Code	Description	Condition
16#60	Simulation active	• SIM_ON = TRUE
16#80	Valid value	• SIM_ON = FALSE and • High byte of MODE input = 16#80 and • (VALUE_QC = FALSE or PQC = FALSE)
16#48	Substituted value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = TRUE
16#00	Invalid value	• SIM_ON = FALSE and • invalid value and • SUBS_ON = FALSE

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states, while for the "Invalid value" state simulation and value substitution must be deactivated via their respective input parameters.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "I 10.4") of the signal that is to be processed by the channel driver block must be interconnected to the VALUE parameter by the user. If an external quality signal is to be used, it is to be interconnected to the VALUE_QC input parameter, and PQC needs to be set to TRUE. For suitable CPX modules, this is done automatically by the "Generate Module Drivers" function.

Normal Operation

During normal operation (valid value), the input signal provided at VALUE is written to the Q output parameter. QUALITY is set to 16#80 and QBAD to FALSE.

Simulation Mode

Simulation can be activated by setting SIM_ON = TRUE. The value provided at SIM_I is then written to the Q output parameter, and QUALITY is set to 16#60. QBAD is set to FALSE, while QSIM = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SIM_ON is set to TRUE, any other value is ignored. Additionally, some block parameters might be protected against manual modifications. This also affects the substitute value, which is not effective when simulation is active.

Substitute Value

If SUBS_ON = TRUE and the value read from the process image is invalid, QBAD = TRUE is set and the value defined at or interconnected to SUBS_I is written to the Q output parameter. QUALITY is set to 16#48, QSUBS = TRUE states that the value currently provided at Q is substituted.

Invalid Value

If SUBS_ON is set to FALSE and the value read from the process image (via VALUE) is invalid, it is nevertheless

written to the Q output parameter. QBAD = TRUE indicates the invalid value, and QUALITY is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at MODE, QMOD_ERR = TRUE is set.

Error Handling

The plausibility of input parameters is not checked.

Startup Characteristics

N/A

Time Response

N/A

Message Response

N/A

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

9.4.3 I/Os of CHDI_CPX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

9.4.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
MODE	Quality and mode (from MOD_CPXx module driver block)	DWORD	16#0		C

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QBAD	1=bad process value (quality information in MODE high byte <> 16#80 or VALUE_QC = FALSE and external quality information is activated via PQC = TRUE	BOOL	FALSE	N/A	
QMOD_ERR	1=higher level error (quality information in MODE high byte = 16#40)	BOOL	FALSE	N/A	
QSIM	1=simulation active (SIM_ON = TRUE)	BOOL	FALSE	N/A	
QSUBS	1=failure substitution active (invalid value and SUBS_ON = TRUE)	BOOL	FALSE	N/A	

Q	Process value	BOOL	FALSE	N/A	
QUALITY	Quality code	BYTE	16#0	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
PQC	1=use process image quality bit	BOOL	FALSE		(V)*
VALUE	Input value	BOOL	FALSE		
VALUE_QC	Process image quality bit (only used if activated via PQC = TRUE)	BOOL	FALSE		(C)*
SIM_ON	1=activate simulation	BOOL	FALSE		
SIM_I	Simulation value	BOOL	FALSE		
SUBS_ON	1=enable failure substitution	BOOL	FALSE		
SUBS_I	Substitution value	BOOL	FALSE		

* Only for modules providing hardware quality information in the process image data.

Additional Information

For further information, please check the following sections:

- Block Description

9.5 CHDO_CPX: Festo CPX Digital Output Channel

9.5.1 CHDO_CPX: Festo CPX Digital Output Channel (Non APL)

This chapter contains the following information:

- Block Description
- Interface Description

9.5.2 Description of CHDO_CPX

Object Name (Type + Number)

FC 531

Area of Application

CHDO_CPX is used for any digital output signal sent to a Festo CPX terminal. This includes CPX digital output modules, I-Port devices, valve terminal and CPI modules connected to a CPX terminal.

This driver block has been designed for usage in non-APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The MODE parameter is interconnected with the appropriate OMODEx output of the respective module driver block.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's MODE input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the QBAD output

signal.

Functional Description

CHDO_CPX processes a digital signal interconnected by the user from the user program to the I input parameter cyclically (depending on the insert point chosen by the user). The processed result is supplied at the VALUE output parameter, which needs to be interconnected to the appropriate periphery signal by the user.

Quality Code

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the MODE input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block.

The internal part is based on the setting of the SIM_ON parameter (see table below).

The following quality codes are generated by the channel driver block:

Quality Code	Description	Condition
16#60	Simulation active	• SIM_ON = TRUE
16#80	Valid value	• SIM_ON = FALSE and • High byte of MODE input = 16#80
16#00	Invalid value	• SIM_ON = FALSE and • invalid value

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "Q 10.4") of the signal that is to be provided by the channel driver block must be interconnected to the VALUE output parameter by the user.

Normal Operation

During normal operation (valid value), the signal provided at I is written to the VALUE output parameter. QUALITY is set to 16#80 and QBAD to FALSE.

Simulation Mode

Simulation can be activated by setting the SIM_ON = TRUE. The value provided at SIM_I is then written to the VALUE output parameter, and QUALITY is set to 16#60. QBAD is set to FALSE, while QSIM = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SIM_ON is set to TRUE, any other value is ignored.

I/O Fault

Even if a module error is signalled via the MODE parameter's high byte, the value provided at the I input parameter is always written to the VALUE output parameter.

QBAD = TRUE indicates the invalid value, and QUALITY is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at MODE, QMOD_ERR = TRUE is set.

Error Handling

The plausibility of input parameters is not checked.

Startup Characteristics

The value provided at the I input parameter is always written to the VALUE output parameter. This also applies to the startup phase of the CPU.

Time Response

N/A

Message Response

N/A

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

9.5.3 I/Os of CHDO_CPX

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

9.5.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
QBAD	1=bad process value (quality information in MODE high byte <> 16#80 or VALUE_QC = FALSE and external quality information is activated via PQC = TRUE	BOOL	FALSE	N/A	
QMOD_ERR	1=higher level error (quality information in MODE high byte = 16#40)	BOOL	FALSE	N/A	
QSIM	1=simulation active (SIM_ON = TRUE)	BOOL	FALSE	N/A	
VALUE	Output value	BOOL	FALSE	N/A	
QUALITY	Quality code of the output value	BYTE	16#0	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
I	Process value	BOOL	FALSE		
SIM_ON	1=activate simulation	BOOL	FALSE		
SIM_I	Simulation value	BOOL	FALSE		

Additional Information

For further information, please check the following sections:

- Block Description

10 Channel Driver Blocks (APL)

10.1 CPXAnIn: Festo CPX Analogue Input Channel

10.1.1 CPXAnIn: Festo CPX Analogue Input Channel (APL)

This chapter contains the following information:

- Block Description
- Interface Description

10.1.2 Description of CPXAnIn

Object Name (Type + Number)

FB 532

Area of Application

CPXAnIn is used for any analogue input signal coming from a Festo CPX terminal. This includes CPX analogue input modules as well as pressure sensors or the feedback signals of proportional valves mounted in a valve terminal.

This driver block has been designed for usage in APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The `MS` parameter is interconnected with the `O_MS` output of the respective module driver block.
- The `Mode` parameter is interconnected with the appropriate `OMODEx` output of the module driver block.
- The `DataXchg` parameter is interconnected with the appropriate `DXCHG_xx` output of the module driver block.
- The `ChLRange` and `ChHRange` parameters are set to either the values defined in HWConfig ("Scale lowest value"/"Scale highest value") or predefined values, depending on the installed module and which input format has been chosen in HWConfig.
- The `Format` parameter is set to the input format chosen in HWConfig or predefined values for certain modules.
- The `InptType` parameter is set to either the signal area chosen in HWConfig (e. g. "4..20 mA") or predefined values for certain modules.
- The `ByteOrdr` parameter is set to the value defined for the CPX terminal in HWConfig (right click the CPX terminal's icon in HWConfig -> Object Properties -> Parameter Assignment -> Device-specific parameters -> Byte order of analogue values).
- The `Scale.Low` and `Scale.High` parameters are set up for pressure sensor signals to reflect the measurement range correctly. As for temperature sensor signals, the input word provided by the module then directly contains the actual physical value (e. g. pressure in mbar), in some cases with additional places. Scaling is then only applied by CPXAnIn to convert it to its base unit.
- The `PV_InUnit` parameter is set up according to the settings made in HWConfig for temperature and pressure signals. Please check the Siemens APL online help for details on how physical units are coded.

After a run of the "Generate Module Drivers" function, CPXAnIn is fully adjusted to the settings made in HWConfig. Thus, the internal scaling algorithm fully adjusts itself to the settings made by the user. Please be aware that after changes are applied in HWConfig, the "Generate Module Drivers" function must be run in order to transfer the changed settings to the CPXAnIn instances.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's `Mode` input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the `Bad` output signal.

Additionally, the parameters that are normally set up by the "Generate Module Drivers" function automatically need to be configured according to the module used and the settings made in HWConfig (see above for details).

Finding the correct settings for the block's parameters manually is a complex task, and the risk of misconfigurations is high. Additionally, some block parameters might be protected against manual modifications. It is therefore strongly suggested to use the "Generate Module Drivers" function and not to configure the channel driver block manually.

Functional Description

CPXAnIn processes an analogue signal interconnected by the user from the process image (input word of the respective channel) to the `PV_In` parameter cyclically (depending on the insert point chosen by the user).

The block reads the raw data value and converts it to the corresponding physical value by scaling it between the lower and upper physical limits defined by the `Scale.Low` and `Scale.High` parameters. Scaling is only applied if the analogue signal is provided as either voltage or current (e. g. 4..20 mA). For temperature and pressure values provided by the appropriate modules, sensors or proportional valves, scaling is only applied to convert the physical value to its base unit. In these cases, the `Scale.Low` and `Scale.High` parameters are ignored.

The processed result is supplied at the `PV_Out` output parameter.

If the "Generate Module Drivers" function is used, the block is configured automatically to adjust the internal algorithms to the settings made in HWConfig (see above for details).

If the input format "comparator bit" is used for VPPM proportional valves, the process value will be displayed as 0.0 if the comparator bit is not set by the VPPM module and 1.0 if the comparator bit is set.

If the input format "process value + comparator bit" is used, the comparator bit is ignored by the channel driver block, and the current value is used as process value. This would provide the same result as using the "process value" setting.

Signal Status

The status of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the `Mode` input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block.

The internal part is based on the setting of the `SimOn`, `Feature.Bit29` (bad value substitution) and `Feature.Bit30` (hold last value) parameters (see table below).

The following signal states are generated by the channel driver block:

Signal Status	Description	Condition
16#60	Simulation active, substituted value or last valid value	• `SimOn` = TRUE OR • `SimOn` = FALSE and • invalid value and • `Feature.Bit29` = TRUE OR • `SimOn` = FALSE and • invalid value and • `Feature.Bit29` = FALSE and • `Feature.Bit30` = TRUE
16#80	Valid value	• `SimOn` = FALSE and • High byte of `Mode` input = 16#80
16#00	Invalid value	• `SimOn` = FALSE and

		• invalid value and • Feature.Bit29 = FALSE and • Feature.Bit30 = FALSE

The table reflects the priority of the individual states. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states, while for the "Invalid value" state simulation, value substitution and keeping the last valid value must all be deactivated via their respective input parameters.

The signal status is transmitted at the PV_Out.ST parameter for further processing by APL blocks.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "IW 512") of the signal that is to be processed by the channel driver block must be interconnected to the PV_In parameter by the user.

Raw Value Check

Depending on the measurement type and range of the analog input module, the nominal range sets the range for converting analog signals into digital values (raw values). This includes an overshoot/undershoot range within which an analog signal can still be converted. Values outside this range constitute an overflow or underflow. The block indicates whether the raw value lies within the nominal range of the module.

Depending on its hardware revision, a CPX analogue input module might not provide overshoot and undershoot ranges. A 4..20 mA signal, for example, might be limited to either the lower physical limit (4 mA), the upper limit (20 mA), or both.

Additionally, undershoot and overshoot are not supported by temperature and pressure sensor modules, which provide the physical value directly and do not require additional scaling by the CPXAnIn.

PV_LoAct = TRUE indicates that the raw value is below the lower physical limit (undershoot), PV_HiAct = TRUE that it is above the upper physical limit (overshoot).

Bad = TRUE (channel error) is also set if overflow or underflow is detected.

NAMUR Limit Check

By NAMUR definition, the current of a 4..20 mA loop is valid as long as 3.6 mA <= current <= 21 mA; if those limits are exceeded, the device is considered dead, and Bad = TRUE is set. Alternatively, customized current limits may be defined via HighLimit and LowLimit and activated by setting NamurOff = TRUE.

NAMUR limit checks only apply to 4..20 mA signals. If CPXAnIn is used for processing other signal formats, the check is omitted. Depending on the module type and its hardware revision, undershoot and/or overshoot might not be supported. In this case, the raw data value is limited to the nominal signal range (4 and/or 20 mA) by the module itself, rendering the NAMUR limit check ineffective.

NAMUR limit checks are processed in the AS and do not correlate with the analogue input module's internal limit monitoring. It is advised to deactivate the latter by leaving the "Channel x: Monitor lower (resp. upper) limit" parameters to their default disabled setting.

Normal Operation

During normal operation (valid value), the raw data value is converted to its physical value by scaling it between the limits defined by the Scale.Low and Scale.High parameters (for voltage and current signals) or by forwarding the physical value received from the hardware (for temperature and pressure signals). The type of the signal connected to CPXAnIn is defined by the InptType parameter, which is set by the "Generate Module Drivers" function automatically, based on the configuration done in HWConfig.

In detail, the following conversions are applied for the different types of input signals:

INPTTYPE	Type of Input Signal	Default Data Conversion	Scale.Low, Scale.High
16#0	unknown/by DIL switch	VALUE is scaled between Scale.Low and Scale.High	Used
16#1	0..10 V	VALUE is scaled between Scale.Low and Scale.High	Used
16#2	+/- 10 V	VALUE is scaled between Scale.Low and Scale.High	Used

16#3	+/- 5 V	VALUE is scaled between Scale.Low and Scale.High	Used
16#4	1..5 V	VALUE is scaled between Scale.Low and Scale.High	Used
16#5	0..20 mA	VALUE is scaled between Scale.Low and Scale.High	Used
16#6	4..20 mA	VALUE is scaled between Scale.Low and Scale.High	Used
16#7	+/-20 mA	VALUE is scaled between Scale.Low and Scale.High	Used
16#8	0..10 V (w/o undershoot)	VALUE is scaled between Scale.Low and Scale.High	Used
16#9	0..20 mA (w/o undershoot)	VALUE is scaled between Scale.Low and Scale.High	Used
16#A	4..20 mA (w/o undershoot)	VALUE is scaled between Scale.Low and Scale.High	Used
16#14	Pressure in mbar	VALUE is converted to bar (by dividing by 1000), Scale.Low and Scale.High (and thus ScaleOut.Low and ScaleOut.High) are calculated automatically	Not used
16#15	Pressure in kPa	VALUE is used without scaling, Scale.Low and Scale.High (and thus ScaleOut.Low and ScaleOut.High) are calculated automatically	Not used
16#16	Pressure in 0.1 psi	VALUE is converted to psi (by dividing by 10), Scale.Low and Scale.High (and thus ScaleOut.Low and ScaleOut.High) are calculated automatically	Not used
16#1E	Temperature in 0.1 °C	VALUE is converted to °C (by dividing by 10), Scale.Low and Scale.High are ignored	Not used
16#1F	Temperature in 0.1 °F	VALUE is converted to °F (by dividing by 10), Scale.Low and Scale.High are ignored	Not used

As shown above, linear scaling is always applied for voltage and current input signals, with the scaling limits being defined by Scale.Low and Scale.High. If, for example, Scale.Low is set to 0.0 and Scale.High to 100.0, the input value is converted to a percentage. Or, for processing the value delivered by a 0..150 °C temperature sensor via 4..20 mA, Scale.Low needs to be set to 0.0 and Scale.High to 150.0.

Pressure and temperature values are scaled to their base units (bar, kPa, psi, °C, °F). For pressure values, Scale.Low and Scale.High are set up according to the channel's measurement range. However, they do not affect the internal calculation of CPXAnIn in this case.

In order to adjust the internal scaling algorithm to the settings made in HWConfig, the data format (e. g. 12 bit right bound), byte order (Intel/Motorola) and low and high range limits of the raw data value are to be set up at the appropriate parameters. The "Generate Module Drivers" function fully takes care of that matter, such that the CPXAnIn block is fully adjusted to the hardware configuration automatically. For that reason, it does not make a difference to the driver block which data format is applied. However, it is advised to stay with the default settings and just change the byte order to "Motorola" in the CPX terminal's system parameters in HWConfig for easier debugging (due to the swapped bytes, the default "Intel" setting does not allow for convenient monitoring of the raw value in CFC test mode).

The lower and upper limit parameters (Scale.Low/Scale.High) are copied to the outputs ScaleOut.Low and ScaleOut.High to allow the interconnection of the settings to other block I/Os.

During normal operation (valid value), PV_Out.ST is set to 16#80 and Bad to FALSE.

For more information about the parameters mentioned above, please also check the Interface Description for CPXAnIn.

Simulation Mode

Simulation can be activated by setting the SimOn = TRUE. The value provided at SimPV_In is then written to the PV_Out output parameter instead of processing the raw value PV_In.PV_Out.ST is set to 16#60, Bad is set to FALSE, while SimAct = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SimOn is set to TRUE, any other value is ignored. This also affects the substitute and hold last value values, which are not effective when simulation is active.

Substitute Value

If Feature.Bit29 = TRUE and the value read from the process image is invalid, Bad = TRUE is set and the value defined at or interconnected to SubsPV_In is written to the PV_Out output parameter. PV_Out.ST is set to 16#60. Feature.Bit28 needs to be set to FALSE for Bit29 to become effective.

Hold Last Value

If Feature.Bit30 = TRUE, Feature.Bit29 = FALSE and the value read from the process image is invalid (Mode high byte <> 16#80), Bad = TRUE is set and the last valid value is held at the PV_Out output (i. e. the value that was effective before it was detected that the process data has become invalid). PV_Out.ST is set to 16#60.

Invalid Value

If both Feature.Bit29 and Feature.Bit30 are set to FALSE and the value read from the process image is indicated as invalid (Mode high byte <> 16#80), the result of the scaling is still written to the PV_Out output parameter.

Exception: If CPXAnIn itself detects an error during the scaling (broken wire or input overload), the result of the scaling is ignored, and the process value's low limit value (Scale.Low) is written to PV_Out. This only applies if a 2AE-U-I or 4AE-I module with data format "12bit left bound + diagnosis" or a CPX-4AE-U-I with "sign+15 bit" data format (Format = 3 or 4) is used. Other data formats and/or modules do not provide broken wire information within the raw data value.

In both cases, Bad = TRUE indicates the invalid value, and PV_Out.ST is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at Mode, ModErr = TRUE is set.

Flutter Suppression

Flutter suppression is not supported by CPXAnIn. If you encounter fluttering messages due to limits being exceeded repeatingly, you might want to use the "Filter measured value" parameter in HWConfig to steady the measurement value.

Feature Parameter

CPXAnIn can be configured using the Feature parameter. Please check the Siemens APL online help for an overview of all reactions which are provided by the Feature parameter. The following functionality is supported by CPXAnIn:

Bit	Description
28	Use invalid value
29	Use substitute value if value is invalid
30	Use last valid value if value is invalid

Sign-Of-Life Monitoring

Sign-of-life monitoring is not supported by CPXAnIn.

Error Handling

The plausibility of input parameters is not checked.

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

10.1.3 I/Os of CPXAnIn

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

10.1.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
PV_In	Input/periphery value	WORD	16#0		
Scale	Range of process value	STRUCT • High: REAL • Low: REAL	- • High: 100.0 • Low: 0.0		(V)*
PV_InUnit	Unit of process value: Please check the Siemens APL online help for details on how physical units are coded.	INT	1001		(V)**
NamurOff	1=NAMUR limits off	BOOL	FALSE		
HighLimit	High limit of input value (in mA) used if NamurOff=1	REAL	21.5		
LowLimit	Low limit of input value (in mA) used if NamurOff=1	REAL	3.3		
SimOn	1=Activate simulation	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SimPV_In	Simulation value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80		
SubsPV_In	Substitution value	REAL	0.0		
ChHRange	High range of raw value: "Scale highest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
ChLRange	Low range of raw value: "Scale lowest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure	INT	0	X	V

	values				
ByteOrdr	Byte order (0=Intel, 1=Motorola): As set up in HWConfig in the terminal's (not module's!) properties, "Byte order of analogue values"	BOOL	FALSE	X	V
Format	Input format ("Input format" or "Input signal type" in HWConfig, depending on the module type). Valid values: • 16#0=15bit linear scaled • 16#1=12 bit right bound • 16#2=15bit • 16#3=12bit left bound + diagnosis • 16#4=sign+15 bit (ET200 compatible) • 16#A=process value (no scaling required) • 16#B=comparator bit • 16#C=process value + comparator bit	BYTE	0	X	V
InptType	Input signal type ("Signal area", "Measuring adjustment", "Dimension unit" or "Input unit" in HWConfig, depending on the module type). Valid values: • 16#0=no sensor connected/range set via DIL switch/linear scaled (only 4AE-P) • 16#1=0..10V • 16#2=+/- 10V • 16#3=+/- 5V • 16#4=1..5V • 16#5=0..20mA • 16#6=4..20mA • 16#7=+/-20mA • 16#8=0..10V (without undershoot range) • 16#9=0..20mA (without undershoot range) • 16#A=4..20mA (without undershoot range) • 16#14=pressure in mbar • 16#15=pressure in kPa	BYTE	0	X	V

	• 16#16=pressure in 0,1 psi • 16#1E=temperature in 0.1 deg C • 16#1F=temperature in 0.1 deg F				
MS_Release	Maintenance release	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
MS	Maintenance state	DWORD	16#0		C
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
Feature	Status of various features	STRUCT • Bit0: BOOL • … • Bit28: BOOL • … • Bit31: BOOL	- • FALSE • FALSE • TRUE • FALSE • FALSE		

If the input format "comparator bit" is used for VPPM proportional valves, the process value will be displayed as 0.0 if the comparator bit is not set by the VPPM module and 1.0 if the comparator bit is set.

If the input format "process value + comparator bit" is used, the comparator bit is ignored by the channel driver block, and the current value is used as process value. This would provide the same result as using the "process value" setting.

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
Bad	1=bad process value: (Quality information in `Mode` high byte <> 16#80 or broken wire	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

	detected or signal out of NAMUR or defined range) and simulation not active				
PV_Out	Process value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80	N/A	
PV_OutUnit	Unit of process value: Copied from `PV_InUnit` parameter. Please check the Siemens APL online help for details on how physical units are coded.	INT	0	N/A	
ScaleOut	Range of process value. Copied from `Scale` input parameter.	STRUCT • High: REAL • Low: REAL	- • High: 100.0 • Low: 0.0	N/A	
PV_HiAct	1=process value high limit failure: `PV_In` > `ChHRange`	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
PV_LoAct	1=process value low limit failure: `PV_In` < `ChLRange`	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
SimAct	1=simulation active: `SimOn` = TRUE	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
ModErr	1=higher level error: Quality information in `Mode` high byte = 16#40	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
OosAct	Field device out of service, maintenance in progress	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
DataXchg	Data exchange: • Bit 0: Maintenance release • Bit 1..31: Not	DWORD	16#0		C

	used				

* If the analogue signal processed by the channel driver block is a pressure value, the `Scale.Low` and `Scale.High` parameters are set up to reflect the measurement range of the module in question. In both cases as well as for temperature values, the input word provided by the input module directly contains the actual physical value (e. g. in °C), in some cases with additional decimal places. Scaling is then only applied by CPXAnIn to convert it to its base unit.

** The `PV_InUnit` parameter is set up according to the settings made in HWConfig for temperature and pressure signals.

Additional Information

For further information, please check the following sections:

- Block Description

10.2 CPXAnOu: Festo CPX Analogue Output Channel

10.2.1 CPXAnOu: Festo CPX Analogue Output Channel (APL)

This chapter contains the following information:

- Block Description
- Interface Description

10.2.2 Description of CPXAnOu

Object Name (Type + Number)

FB 533

Area of Application

CPXAnOu is used for any analogue input signal written to a Festo CPX terminal. This includes CPX analogue output modules as well as set point values for proportional valves mounted in a valve terminal.

This driver block has been designed for usage in APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The `MS` parameter is interconnected with the `O_MS` output of the respective module driver block.
- The `Mode` parameter is interconnected with the appropriate `OMODEx` output of the respective module driver block.
- The `DataXchg` parameter is interconnected with the appropriate `DXCHG_xx` output of the module driver block.
- The `ChLRange` and `ChHRange` parameters are set to either the values defined in HWConfig ("Scale lowest value"/"Scale highest value") or predefined values, depending on the installed module and which input format has been chosen in HWConfig.
- The `ByteOrdr` parameter is set to the value defined for the CPX terminal in HWConfig (right click the CPX terminal's icon in HWConfig -> Object Properties -> Parameter Assignment -> Device-specific parameters -> Byte order of analogue values).

After a run of the "Generate Module Drivers" function, CPXAnOu is fully adjusted to the settings made in HWConfig. Thus, the internal scaling algorithm fully adjusts itself to the settings made by the user. Please be aware that after changes are applied in HWConfig, the "Generate Module Drivers" function must be run in order to transfer the changed settings to the CPXAnOu instances.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's `Mode` input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the `Bad` output signal.

Additionally, the parameters that are normally set up by the "Generate Module Drivers" function automatically need to be configured according to the module used and the settings made in HWConfig (see above for details).

Finding the correct settings for the block's parameters manually is a complex task, and the risk of misconfigurations is high. Additionally, some block parameters might be protected against manual modifications. It is therefore strongly suggested to use the "Generate Module Drivers" function and not to configure the channel driver block manually.

Functional Description

CPXAnOu processes an analogue set point value interconnected to the `PV_In` parameter cyclically (depending on the insert point chosen by the user) and provides a suitable raw value at the `PV_Out` output parameter, which is to be interconnected to the process image (output word of the respective channel) by the user. Linear scaling is applied in order to do the conversion from process to periphery format.

If the "Generate Module Drivers" function is used, the block is configured automatically to adjust the internal algorithms to the settings made in HWConfig (see above for details).

Signal Status

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the `Mode` input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block.

The internal part is based on the setting of the `SimOn` parameter and a comparison of the set point value provided at `PV_In` against the limits defined by the `Range` parameter structure (see table below).

The following quality codes are generated by the channel driver block:

Signal Status	Description	Condition
16#60	Simulation active	• `SimOn` = TRUE
16#78	Low or high limit exceeded	• `SimOn` = FALSE AND(• `PV_In` > `Scale.High` or • `PV_In` < `Scale.Low`)
16#80	Valid value	• `SimOn` = FALSE and • High byte of `Mode` input = 16#80 and • `Scale.Low` <= `PV_In` <= `Scale.High`
16#00	Invalid value	• `SimOn` = FALSE and • invalid value

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states.

The signal status is transmitted at the `PV_ChnST.ST` parameter for further processing by APL blocks.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "QW 512") of the signal that is to be processed by the channel driver block must be interconnected to the VALUE parameter by the user.

Normal Operation

During normal operation (valid value), the process value is converted to the raw value by scaling it between the limits between the process value limits `Scale.Low`/`Scale.High` and the raw data limits `ChLRange`/`ChHRange`.

In contrast to the analogue input driver blocks, scaling is always applied by CPXAnOu, and the `Scale.Low` and `Scale.High` parameters are set up by default to interpret the value

provided at PV_In as percentage. If the set point value for a VPPM proportional value is to be interpreted as physical value, Scale.Low and Scale.High need to be set to the appropriate physical limits of the hardware (e. g. 0.0/6000.0 or 0.0/6.0 for a 0..6 bar VPPM valve, depending on whether the value is provided as millibars or bars by the PCS7 application).

In order to adjust the internal scaling algorithm to the settings made in HWConfig, the byte order (Intel/Motorola) and low and high range limits of the raw data value are to be set up at the appropriate parameters. The "Generate Module Drivers" function fully takes care of that matter, such that the CPXAnOu block is fully adjusted to the hardware configuration automatically. For that reason, it does not make a difference to the driver block which byte order is applied. However, it is advised to change the byte order to "Motorola" in the CPX terminal's system parameters in HWConfig for easier debugging (due to the swapped bytes, the default "Intel" setting does not allow for convenient monitoring of the raw value in CFC test mode).

The lower and upper limit parameters (Scale.Low/Scale.High) are copied to the outputs ScaleOut.Low and ScaleOut.High to allow the interconnection to other block I/Os.

During normal operation (valid value), PV_ChnST.ST is set to 16#80 and Bad to FALSE.

For more information about the parameters mentioned above, please also check the Interface Description for CPXAnOu.

Limiting Of The Process Value

The CPX/MPA hardware does NOT support an under- or overshoot for the periphery signal; the set point value is therefore always limited to the values given at the Scale.Low/Scale.High parameters, before scaling is applied. The PV_HiAct and PV_LoAct output parameters indicate whether the process value has been limited. If, for example, a CPXAnOu block is configured with Scale.Low =0.0 and Scale.High =6.0, a process value PV_In =6.5 would be treated as 6.0 internally, and PV_HiAct = TRUE would be set.

Additional limits can be applied if LimitsOn = TRUE is set. Now, Limits.Low and Limits.High are effective in addition to ScaleOut.Low and ScaleOut.High. But in contrast to the latter, the former do not take influence on the scaling algorithm itself, but only apply the following limits to the process value PV_In:

- PV_In > Limits.High: PV_In is limited to Limits.High
- PV_In < Limits.Low: PV_In is limited to Limits.Low

Activation of those limits is not indicated at the block's interface.

An example might be a VPPM proportional valve with physical limits of 0..6 bars. So, Scale.Low = 0.0 and Scale.High = 6.0 would allow for providing the process value PV_In in bars from the PCS7 application. If now the actual pressure at the valve's outlet connector would need to be limited to 5 bars, LimitsOn = TRUE and Limits.High = 5.0 would allow for exactly that. If Scale.High was used instead, the scaling would be affected such that PV_In = 5.0 would then make CPXAnOu deliver a raw value telling the proportional valve to go for the maximum, i. e. 6 bars (instead of the desired 5 bars).

It is important to keep in mind the difference between the two parameter sets that form limits for the process value PV_In.Scale.Low/Scale.High limit, indicate active limits via PV_HiAct/PV_LoAct and affect the internal scaling algorithm.
Limits.Low/Limits.High need to be activated via LimitsOn and - once activated - do not affect the scaling itself, but only provide "silent" limiting without indication at the block's interface.

Simulation Mode

Simulation can be activated by setting the SimOn = TRUE. The value provided at SimPV_In is then processed instead of the raw value PV_In, and the result is written to the PV_Out output parameter. PV_Out.ST is set to 16#60, Bad is set to FALSE, while SimAct = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SimOn is set to TRUE, any other value is ignored. This also affects the substitute and hold last value values, which are not effective when simulation is active.

If an external simulation is detected via a 16#60 value provided at PV_In.ST and Feature.Bit30 = FALSE, PV_Out is set to Scale.Low to provoke a powerless state for the analogue output and PV_ChnST.ST remains at 16#80. If Feature.Bit30 = TRUE (default), PV_Out is set according to PV_In in spite of the external simulation.

I/O Fault

Even if a module error is signalled via the Mode parameter's high byte, the value provided at the PV_In input parameter is always written to the PV_Out output parameter.

Bad = TRUE indicates the invalid value, and PV_Out.ST is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at Mode, ModErr = TRUE is set.

Feature Parameter

CPXAnOu can be configured using the Feature parameter. Please check the Siemens APL online help for an overview of all reactions which are provided by the Feature parameter. The following functionality is supported by CPXAnOu:

Bit	Description
30	1=Use de-energized value at a block-external simulation

Error Handling

The plausibility of input parameters is not checked.

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

10.2.3 I/Os of CPXAnOu

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

10.2.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
PV_In	Process value incl. ST	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80		
PV_InUnit	Unit of process value: Please check the Siemens APL online help for details on how physical units are coded.	INT	1342		
Scale	Range of process value	STRUCT • High: REAL	- • High: 100.0		

		• Low: REAL	• Low: 0.0		
LimitsOn	1=Use additional limits for process value	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
Limits	Additional limits for process value	STRUCT • High: REAL • Low: REAL	- • High: 100.0 • Low: 0.0		
SimOn	1=Activate simulation	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SimPV_In	Simulation value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80		
ChHRange	High range of raw value: "Scale highest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
ChLRange	Low range of raw value: "Scale lowest value" in HWConfig for voltage and current signals, fixed value for temperature and pressure values	INT	0	X	V
ByteOrdr	Byte order (0=Intel, 1=Motorola): As set up in HWConfig in the terminal's (not module's!) properties, "Byte order of analogue values"	BOOL	FALSE	X	V
MS_Release	Maintenance release	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
MS	Maintenance state	DWORD	16#0		C

MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
Feature	Status of various features	STRUCT • Bit0: BOOL • … • Bit30: BOOL • Bit31: BOOL	- • FALSE • FALSE • TRUE • FALSE		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
Bad	1=Bad process value: Quality information in MODE high byte <> 16#80 and simulation not active	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
PV_Out	Periphery output (raw) value	WORD	16#0	N/A	
PV_ChnST	Value and state of PV_Out	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80	N/A	
PV_OutUnit	Unit of process value: Copied from `PV_InUnit` input parameter. Please check the Siemens APL online help for details on how physical units are coded.	INT	0	N/A	
ScaleOut	Range of process value. Copied from `Scale` input parameter.	STRUCT • High: REAL • Low: REAL	- • High: 100.0 • Low: 0.0	N/A	
PV_HiAct	1=process value high limit failure: `PV_In > ChHRange`	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
PV_LoAct	1=process value low limit failure: `PV_In < ChLRange`	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

SimAct	1=simulation active: SimOn = TRUE	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
ModErr	1=higher level error: Quality information in Mode high byte = 16#40	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
OOSAct	Field device out of service, maintenance in progress	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
DataXchg	Data exchange: • Bit 0: Maintenance release • Bit 1..31: Not used	DWORD	16#0		C

Additional Information

For further information, please check the following sections:

- Block Description

10.3 CPXCnt1: Festo CPX-P Counter/Frequency Channel

10.3.1 CPXCnt1: Festo CPX counter type 1 channel driver block (APL)

This chapter contains the following information:

- Block Description
- Interface Description

10.3.2 Description of CPXCnt1

Object Name (Type + Number)

FB 534

Area of Application

CPXCnt1 is used for reading counter or frequency measurement values from a CPX-P-8DE-N... module. It furthermore controls the function of the respective channel via triggering its corresponding output signals.

Please note that if a channel is not configured as counter or frequency measurement but as NAMUR digital input instead, the CPXDiIn channel driver block needs to be used instead.

This driver block has been designed for usage in APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The MS parameter is interconnected with the O_MS output of the respective module driver block.
- The Mode parameter is interconnected with the appropriate OMODEx output of the module driver block.
- The DataXchg parameter is interconnected with the appropriate DXCHG_xx output of the module driver block.
- The InptType parameter is set to the signal type chosen in HWConfig (e. g. counter value).
- The ByteOrdr parameter is set to the value defined for the CPX terminal in HWConfig (right click the CPX terminal's icon in HWConfig -> Object Properties -> Parameter Assignment -> Device-specific parameters -> Byte order of analogue values).
- The EdgePol_Par parameter is set according to the edge polarity (positive/negative) set up in HWConfig.
- The CntDir_Par parameter is set according to the counting direction (up/down) set up in HWConfig.
- The State_In parameter is interconnected to the "State" digital input signal (limit exceeded yes/no) for the respective channel.
- The Diag_In parameter is interconnected to the "Diag" digital input signal (channel error yes/no) for the respective channel.
- The Start_Out parameter is interconnected to the "Start/Stop" digital output signal (start counting/frequency measurement yes/no) for the respective channel.
- The Reset_Out parameter is interconnected to the "Reset" digital output signal (reset counter yes/no) for the respective channel.

After a run of the "Generate Module Drivers" function, CPXCnt1 is fully adjusted to the settings made in HWConfig. Thus, the internal algorithm fully adjusts itself to the settings made by the user. Please be aware that after changes are applied in HWConfig, the "Generate Module Drivers" function must be run in order to transfer the changed settings to the CPXCnt1 instances.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's Mode input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the Bad output signal.

Additionally, the parameters that are normally set up or interconnected by the "Generate Module Drivers" function automatically need to be configured according to the module used and the settings made in HWConfig (see above for details).

Finding the correct settings for the block's parameters manually is a complex task, and the risk of misconfigurations is high. Additionally, some block parameters might be protected against manual modifications. It is therefore strongly suggested to use the "Generate Module Drivers" function and not to configure the channel driver block manually.

Functional Description

CPXCnt1 processes a counter or frequency measurement signal interconnected by the user from the process image (input word of the respective channel) to the PV_In parameter cyclically (depending on the insert point chosen by the user).

When used for a channel configured as counter, the raw data (counter value) is directly forwarded to the V output parameter. If the block is used for a frequency measurement signal, the raw data value is scaled using the Factor parameter. Thus, for example, the frequency value supplied in Hz by the hardware can conveniently be converted into rounds per minute; the default setting for Factor is 1. The processed result is supplied at the V output parameter.

If the "Generate Module Drivers" function is used, the block is configured automatically to adjust the internal algorithms to the settings made in HWConfig (see above for details).

Signal Status

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the Mode input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the

corresponding MOD_CPXx module driver block. Additionally, the status of the Diag_In parameter (interconnected to the corresponding input signal by the "Generate Module Drivers" function, see above for details) is considered.

The internal part is based on the setting of the SimOn, Feature.Bit29 (bad value substitution) and Feature.Bit30 (hold last value) parameters (see table below).

The following quality codes are generated by the channel driver block:

Signal Status	Description	Condition
16#60	Simulation active, substituted value or last valid value	• SimOn = TRUE OR • SimOn = FALSE and • invalid value and • Feature.Bit29 = TRUE OR • SimOn = FALSE and • invalid value and • Feature.Bit29 = FALSE and • Feature.Bit30 = TRUE
16#80	Valid value	• SimOn = FALSE and • High byte of Mode input = 16#80
16#00	Invalid value	• SimOn = FALSE and • invalid value and • Feature.Bit29 = FALSE and • Feature.Bit30 = FALSE

The table reflects the priority of the individual states. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states, while for the "Invalid value" state simulation, value substitution and keeping the last valid value must all be deactivated via their respective input parameters.

The signal status is transmitted at the PV_Out.ST parameter for further processing by APL blocks.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "IW 512") of the counter or frequency measurement value that is to be processed by the channel driver block must be interconnected to the PV_In parameter by the user.

Limit Check

Both counter and frequency measurement values are monitored upon limits by the hardware. The digital input signal interconnected to the State_In parameter by the "Generate Module Drivers" function provides information about whether the respective value is in range or not. It is forwarded to State output parameter and thus can be interconnected to other block I/Os. If State = TRUE, either the lower or upper limit set up in HWConfig has been exceeded.

Suitable messages are generated by the MOD_CPX7 module driver block in both cases (provided that the "Generate Module Drivers" function has been executed before).

Bad = TRUE (channel error) is set in case of broken wire or shortcut.

Normal Operation

During normal operation (valid value), the raw data value received from the hardware is either scaled (for frequency measurement values) or forwarded directly (for counter values). The type of signal connected to CPXCnt1 is defined by the InptType parameter, which is set by the "Generate Module Drivers" function automatically, based on the configuration done in HWConfig.

In detail, the following conversions are applied for the different types of input signals:

InptType	Type of Input Signal	Default Data Conversion

16#0	NAMUR digital input	N/A, use CPXDiIn digital input driver block instead
16#28	Counter value	None, PV_In is used without conversion
16#29	Frequency measurement	PV_In is scaled using Factor

As shown above, scaling is applied for frequency measurement values only. Thus, for example, the speed of an agitator can be easily calculated in rounds per minute. Counter values are always transferred without any data conversion.

In order to adjust the internal scaling algorithm to the settings made in HWConfig, the byte order (Intel/Motorola) and input type setting are to be set up at the appropriate parameters. The "Generate Module Drivers" function fully takes care of that matter, such that the CPXCnt1 block is fully adjusted to the hardware configuration automatically. For that reason, it does not make a difference to the driver block which data format is applied. However, it is advised to change the byte order to "Motorola"in the CPX terminal's system parameters in HWConfig for easier debugging (due to the swapped bytes, the default "Intel" setting does not allow for convenient monitoring of the raw value in CFC test mode).

The CntDir_Par and EdgePol_Par parameters are copied to the outputs CntDir and EdgePol to allow for the interconnection of the settings to other block I/Os.

The Start and Reset parameters are used for controlling the counter or frequency measurement. Both parameters are forwarded to the Start_Out and Reset_Out output parameters, which are interconnected to the appropriate hardware output signals by the "Generate Module Drivers" function and thus control the behaviour of the respective channel.

If Start = TRUE, counting or frequency measurement is activated, and the appropriate value is supplied by the hardware at the input word interconnected to the PV_In input parameter. Start = FALSE freezes the counter value or sets the frequency measurement value to 0.

As long as Reset = TRUE, the counter value is reset to either the lower limit set up in HWConfig (when counting up, CntDir_Par = FALSE) or to the upper limit (when counting down, CntDir_Par = TRUE). The Reset value does have no effect if frequency measurement is used. Please see the module's hardware documentation for further details.

During normal operation (valid value), PV_Out.ST is set to 16#80 and Bad to FALSE.

For more information about the parameters mentioned above, please also check the Interface Description for CPXCnt1.

Simulation Mode

Simulation can be activated by setting the SimOn = TRUE. The value provided at SimPV_In is then written to the PV_Out output parameter (directly, i. e. without using the Factor parameter for frequency measurement) instead of processing the raw value PV_In. PV_Out.ST is set to 16#60, Bad is set to FALSE, while SimAct = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SimOn is set to TRUE, any other value is ignored. This also affects the substitute and hold last value values, which are not effective when simulation is active.

Substitute Value

If Feature.Bit29 = TRUE and the value read from the process image is invalid, Bad = TRUE is set and the value defined at or interconnected to SubsPV_In is written to the PV_Out output parameter. PV_Out.ST is set to 16#60.

Hold Last Value

If Feature.Bit30 = TRUE, Feature.Bit29 = FALSE and the value read from the process image is invalid (Mode high byte <> 16#80 or Diag_In = TRUE), Bad = TRUE is set and the last valid value is held at the PV_Out output (i. e. the value that was effective before it was detected that the process data has become invalid). PV_Out.ST is set to 16#60.

Invalid Value

If both Feature.Bit29 and Feature.Bit30 are set to FALSE and the value read from the process image is indicated as invalid (Mode high byte <> 16#80 or Diag_In = TRUE), the result of the scaling is still written to the PV_Out output parameter.

Bad = TRUE indicates the invalid value, and PV_Out.ST is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at Mode, ModErr =

TRUE is set.

Feature Parameter

CPXCnt1 can be configured using the `Feature` parameter. Please check the Siemens APL online help for an overview of all reactions which are provided by the `Feature` parameter. The following functionality is supported by CPXCnt1:

Bit	Description
28	Use invalid value
29	Use substitute value if value is invalid
30	Use last valid value if value is invalid

Sign-Of-Life Monitoring

Sign-of-life monitoring is not supported by CPXCnt1.

Error Handling

The plausibility of input parameters is not checked.

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

10.3.3 I/Os of CPXCnt1

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

10.3.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
PV_In	Input/periphery value	WORD	16#0		
PV_InUnit	Unit of process value: Please check the Siemens APL online help for details on how physical units are coded.	INT	1998		
Factor	Factor for scaling of frequency measurement (PV_Out = PV_In * Factor)	REAL	0.0		
State_In	Channel state (0=limits not exceeded, 1=limit exceeded)	BOOL	FALSE		C
Diag_In	Channel diagnostics	BOOL	FALSE		C

	(0=ok, 1=channel error)				
Start	Start/stop command (0=stop, 1=start counter/measurement)	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
Reset	Reset command (0=counter active, 1=reset counter)	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SimOn	1=Activate simulation	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SimPV_In	Simulation value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80		
SubsPV_In	Substitution value	REAL	0.0		
ByteOrdr	Byte order (0=Intel, 1=Motorola): As set up in HWConfig in the terminal's (not module's!) properties, "Byte order of analogue values"	BOOL	FALSE	X	V
InptType	Input signal type ("Signal area", "Measuring adjustment", "Dimension unit" or "Input unit" in HWConfig, depending on the module type). Valid values: • 16#0=digital input (NAMUR) • 16#28=counter value • 16#29=frequency in Hz	BYTE	0	X	V
CntDir_Par	Counting direction (0=up, 1=down)	BOOL	FALSE	X	V
EdgePol_Par	Edge polarity (0=rising, 1=falling)	BOOL	FALSE	X	V
MS_Release	Maintenance release	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
MS	Maintenance state	DWORD	16#0		C
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
Feature	Status of various	STRUCT	-		

	features	• Bit0: BOOL • … • Bit28: BOOL • … • Bit31: BOOL	• FALSE • FALSE • TRUE • FALSE • FALSE		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
Bad	1=bad process value: (Quality information in `Mode` high byte <> 16#80 or broken wire detected or signal out of NAMUR or defined range) and simulation not active	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
PV_Out	Process value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80	N/A	
PV_OutUnit	Unit of process value: Copied from `PV_InUnit` parameter. Please check the Siemens APL online help for details on how physical units are coded.	INT	0	N/A	
State	1=limit exceeded	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
CntDir	Counting direction (0=up; 1=down)	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
EdgePol	Edge polarity (0=rising, 1=falling)	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
SimAct	1=simulation active: `SimOn` = TRUE	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
ModErr	1=higher level error: Quality information in `Mode` high byte = 16#40	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
OosAct	Field device out of service, maintenance	STRUCT	-	N/A	

	in progress	• Value: BOOL • ST: BYTE	• FALSE • 16#80		
Start_Out	Start/stop command to hardware (0=stop, 1=start counter/measurement)	BOOL	FALSE	N/A	C
Reset_Out	Reset command to hardware (0=counter active, 1=reset counter)	BOOL	FALSE	N/A	C

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
DataXchg	Data exchange: • Bit 0: Maintenance release • Bit 1..31: Not used	DWORD	16#0		C

Additional Information

For further information, please check the following sections:

- Block Description

10.4 CPXDiIn: Festo Digital Input Channel

10.4.1 CPXDiIn: Festo CPX Digital Input Channel (APL)

This chapter contains the following information:

- Block Description
- Interface Description

10.4.2 Description of CPXDiIn

Object Name (Type + Number)

FB 530

Area of Application

CPXDiIn is used for any digital input signal coming from a Festo CPX terminal. This includes CPX digital input modules as well as modules connected via CPI to a CPX terminal.

This driver block has been designed for usage in APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions a executed automatically by the "**Generate Module Drivers**" function:

- The `MS` parameter is interconnected with the `O_MS` output of the respective module driver block.
- The `Mode` parameter is interconnected with the appropriate `OMODEx` output of the module driver block.

- The DataXchg parameter is interconnected with the appropriate DXCHG_xx output of the module driver block.
- If the hardware provides quality information via a digital input signal, the SelQB parameter is set to TRUE and ProImQB is interconnected with the hardware quality bit.

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's Mode input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the Bad output signal.

Functional Description

CPXDiIn processes a digital signal interconnected by the user from the process image to the PV_In parameter cyclically (depending on the insert point chosen by the user). The processed result is supplied at the PV_Out output parameter.

Signal Status

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the Mode input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block. An external binary (good/bad) quality signal can be interconnected to the ProImQB input and activated by setting SelQB to TRUE. In this case, both information parts need to show a "good" value (logical AND) for the block to consider the value as such.

The internal part is based on the setting of the SimOn, and Feature.Bit29 (bad value substitution) parameters (see table below).

The following quality codes are generated by the channel driver block:

Signal Status	Description	Condition
16#60	Simulation active or substituted value	• SimOn = TRUE OR • SimOn = FALSE and • invalid value and • Feature.Bit29 = TRUE
16#80	Valid value	• SimOn = FALSE and • High byte of MODE input = 16#80 and • (ProImQB = FALSE or SelQB = FALSE)
16#00	Invalid value	• SimOn = FALSE and • invalid value and • Feature.Bit29 = FALSE

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states, while for the "Invalid value" state simulation and value substitution must be deactivated via their respective input parameters.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "I 10.4") of the signal that is to be processed by the channel driver block must be interconnected to the PV_In parameter by the user. If an external quality signal is to be used, it is to be interconnected to the ProImQB input parameter, and SelQB needs to be set to TRUE. For suitable CPX modules, this is done automatically by the "Generate Module Drivers" function.

Normal Operation

During normal operation (valid value), the input signal provided at PV_In is written to the PV_Out output parameter. PV_Out.ST is set to 16#80 and Bad to FALSE.

Simulation Mode

Simulation can be activated by setting SimOn = TRUE. The value provided at SimPV_In is then written to the PV_Out output parameter, and PV_Out.ST is set to 16#60. Bad is set to FALSE, while SimAct = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SimOn is set to TRUE, any other value is ignored. This also affects the substitute value, which is not effective when simulation is active.

Substitute Value

If Feature.Bit29 = TRUE and the value read from the process image is invalid, Bad = TRUE is set and the value defined at or interconnected to SubsPV_In is written to the PV_Out output parameter. PV_Out.ST is set to 16#60. Feature.Bit28 needs to be set to FALSE for Bit29 to become effective.

Invalid Value

If Feature.Bit29 (bad value substitution) is set to FALSE and the value read from the process image (via PV_In) is invalid, it is nevertheless written to the PV_Out output parameter. Bad = TRUE indicates the invalid value, and PV_Out.ST is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at Mode,ModErr = TRUE is set.

Feature Parameter

CPXDiIn can be configured using the Feature parameter. Please check the Siemens APL online help for an overview of all reactions which are provided by the Feature parameter. The following functionality is supported by CPXDiIn:

Bit	Description
28	Use invalid value
29	Use substitute value if value is invalid

Error Handling

The plausibility of input parameters is not checked.

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

10.4.3 I/Os of CPXDiIn

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

10.4.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
PV_In	Input/periphery value	BOOL	FALSE		

SimOn	1=Activate simulation	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SimPV_In	Simulation value	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SubsPV_In	Substitution value	BOOL	FALSE		
ProImQB	Process image quality bit (only used if activated via `SelQB` = TRUE; visible if supported by the CPX module)	BOOL	FALSE		(C)*
SelQB	1=use process image quality bit	BOOL	FALSE		(V)*
MS_Release	Maintenance release	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
MS	Maintenance state	DWORD	16#0		C
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
Feature	Status of various features	STRUCT • Bit0: BOOL • … • Bit28: BOOL • … • Bit31: BOOL	- • FALSE • FALSE • TRUE • FALSE • FALSE		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
Bad	1=bad process value: Quality information in `Mode` high byte <> 16#80 and simulation not active	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
PV_Out	Process value	STRUCT	-	N/A	

		• Value: BOOL • ST: BYTE	• FALSE • 16#80		
SimAct	1=simulation active: SimOn = TRUE	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
ModErr	1=higher level error: Quality information in Mode high byte = 16#40	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
OosAct	Field device out of service, maintenance in progress	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
DataXchg	Data exchange: • Bit 0: Maintenance release • Bit 1..31: Not used	DWORD	16#0		C

* Only for modules providing hardware quality information in the process image data.

Additional Information

For further information, please check the following sections:

- Block Description

10.5 CPXDiOu: Festo CPX Digital Output Channel

10.5.1 CPXDiOu: Festo CPX Digital Output Channel (APL)

This chapter contains the following information:

- Block Description
- Interface Description

10.5.2 Description of CPXDiOu

Object Name (Type + Number)

FB 531

Area of Application

CPXDiOu is used for any digital output signal sent to a Festo CPX terminal. This includes CPX digital output modules, I-Port devices, valve terminal and CPI modules connected to a CPX terminal.

This driver block has been designed for usage in APL (Advanced Process Library) projects.

Calling OBs

The block is installed manually in the appropriate CFC chart. Thus, the calling OB depends on the insert point chosen by the user. In most cases, a cyclic interrupt like e. g. OB35 would be suitable.

Automatic Configuration via "Generate Module Drivers"

The following actions are executed automatically by the "**Generate Module Drivers**" function:

- The MS parameter is interconnected with the O_MS output of the respective module driver block.
- The Mode parameter is interconnected with the appropriate OMODEx output of the module driver block.
- The DataXchg parameter is interconnected with the appropriate DXCHG_xx output of the module driver block

Manual Configuration

If, for any reason, the "Generate Module Drivers" function is not used, the block's Mode input parameter is to be set to 16#80000000 manually in order to prevent the channel driver block from setting the QBAD output signal.

Functional Description

CPXDiOu processes a digital signal interconnected by the user from the user program to the PV_In input parameter cyclically (depending on the insert point chosen by the user). The processed result is supplied at the PV_Out output parameter, which needs to be interconnected to the appropriate periphery signal by the user.

Signal Status

The quality of the signal processed is defined by information provided both from externally and internally.

For the external part, the high byte of the value provided at the Mode input parameter is evaluated. If the "Generate module Drivers" function is used, this parameter is interconnected automatically with the corresponding MOD_CPXx module driver block.

The internal part is based on the setting of the SimOn parameter (see table below).

The following quality codes are generated by the channel driver block:

Quality Code	Description	Condition
16#60	Simulation active	• SimOn = TRUE
16#80	Valid value	• SimOn = FALSE and • High byte of Mode input = 16#80
16#00	Invalid value	• SimOn = FALSE and • invalid value

The table reflects the priority of the individual quality codes. As can also be seen from the "Condition" column, "Simulation active" overrides all other quality states.

The signal status is transmitted at the PV_ChnST.ST parameter for further processing by APL blocks.

Addressing

Either the symbol defined in the symbol table or the absolute operand (e. g. "Q 10.4") of the signal that is to be provided by the channel driver block must be interconnected to the PV_Out output parameter by the user.

Normal Operation

During normal operation (valid value), the signal provided at PV_In is written to the PV_Out output parameter. PV_ChnST.ST is set to 16#80 and Bad to FALSE.

Simulation Mode

Simulation can be activated by setting the SimOn = TRUE. The value provided at SimPV_In is then written to the PV_Out output parameter, and PV_ChnST.ST is set to 16#60. Bad is set to FALSE, while SimAct = TRUE indicates the activated simulation mode at the block's interface.

Simulation has the highest priority of all the operating modes. That means that as long as SimOn is set to TRUE, any other value is ignored.

If an external simulation is detected via a 16#60 value provided at PV_In.ST and Feature.Bit30 = FALSE, PV_Out is set to FALSE to provoke a powerless state for the digital output and PV_ChnST.ST remains at 16#80. If Feature.Bit30 = TRUE (default), PV_Out is set according to PV_In in spite of the external simulation.

I/O Fault

Even if a module error is signalled via the Mode parameter's high byte, the value provided at the PV_In input parameter is always written to the PV_Out output parameter.

Bad = TRUE indicates the invalid value, and PV_ChnST.ST is set to 16#00.

Module Error

If a higher order module error is detected via a 16#40 high byte in the value provided at Mode,ModErr = TRUE is set.

Feature Parameter

CPXDiOu can be configured using the Feature parameter. Please check the Siemens APL online help for an overview of all reactions which are provided by the Feature parameter. The following functionality is supported by CPXDiOu:

Bit	Description
30	0=Use lowest/de-energized value at a block-external simulation

Error Handling

The plausibility of input parameters is not checked.

Startup Characteristics

The value provided at the PV_In input parameter is always written to the PV_Out output parameter. This also applies to the startup phase of the CPU.

Operating And Monitoring

The block does not have a faceplate.

Additional Information

For further information, please check the following sections:

- Interface Description

10.5.3 I/Os of CPXDiOu

The default setting for a parameter's visibility can be seen in the "I/O Name" column: I/O name **bold** = I/O visible, I/O name normal = I/O not visible.

The "Locked" column indicates whether a parameter is locked against changes by the user in order to provide configuration errors. If an "X" is shown in this column, the respective parameter cannot be changed manually. "N/A" is stated for OUT parameters, since those cannot be edited due to their type.

The "Set by Driver Generators" defines whether a parameter is set up automatically when the "Generate Module Drivers" function is run. "V" states that a value is written to the parameter upon each run, "C" indicates a CFC connection automatically created. Please be aware that any manual changes applied to parameters defined as either "V" or "C" in this column will be overridden on each run of the "Generate Module Drivers" function.

10.5.4 IN Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
PV_In	Process value incl. ST	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		

SimOn	1=Activate simulation	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
SimPV_In	Simulation value	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
MS_Release	Maintenance release	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80		
MS	Maintenance state	DWORD	16#0		C
MODE	Quality and mode: Interconnected from MOD_CPXx module driver block	DWORD	16#0		C
Feature	Status of various features	STRUCT • Bit0: BOOL • ... • Bit30: BOOL • Bit31: BOOL	- • FALSE • FALSE • TRUE • FALSE		

OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
Bad	1=Bad process value: Quality information in MODE high byte <> 16#80 and simulation not active	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
PV_Out	Periphery output value	BOOL	FALSE	N/A	
PV_ChnST	Value and state of PV_Out	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80	N/A	
SimAct	1=simulation active: SimOn = TRUE	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	
ModErr	1=higher level error: Quality information in Mode high byte = 16#40	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

OOSAct	Field device out of service, maintenance in progress	STRUCT • Value: BOOL • ST: BYTE	- • FALSE • 16#80	N/A	

IN_OUT Parameters

I/O Name	Description	Data Type	Default Value	Locked	Set by Driver Generator
DataXchg	Data exchange: • Bit 0: Maintenance release • Bit 1..31: Not used	DWORD	16#0		C

Additional Information

For further information, please check the following sections:

- Block Description

11 Appendix

11.1 Message Classes

11.1.1 General Information

Message classes are used to group messages according to their cause. The following message classes are used in SIMATIC PCS 7:

- Process messages triggered when process-specific monitoring values (for example, alarms, warnings, high/low tolerances, general process messages) are reached or exceeded.
- Process-control messages; output by the control system (system messages) or the I/O units (errors in the field) or for preventive maintenance.
- Requests for operator input which, in the case of certain operation sequences, draw the operator's attention to the necessity of an operator intervention (for example, request to acknowledge a stepping operation manually in order to enable transition) or operation logs.

11.1.2 Available Message Classes

Message Class	Description	With Ack.
AH	Alarm high (High High Alarm)	Yes
AL	Alarm low (Low Low Alarm)	Yes
WH	Warning high (High Alarm)	Yes
WL	Warning low (Low Alarm)	Yes
TH	Tolerance high (Tolerance High)	Yes
TL	Tolerance low (Tolerance Low)	Yes
F	AS control system message (error)	Yes
S	AS control system message (fault)	Yes
S*	OS control system message (fault)	Yes
M	Preventive maintenance (Maintenance)	Yes
PM	Process message (Process Message)	Yes
-	Operating message	No
OR	Operator request (Operator Request)	No
OM	Operator message (Operation Message)	No
SA	Status AS	No
SO	Status OS	No

Source: PCS 7 Basic Library V7.1 online help.

11.2 Marking Special Information

You can find information on how danger warnings are represented under "Important User Information".

11.2.1 Pictograms

The following pictograms mark passages in the text that contain special information.

Information: Recommendations, tips and references to other sources of information.

Accessory: Information on necessary or useful accessories.

Environment: Information on the environmentally friendly use of Festo products.

11.2.2 Further conventions

[Project] [New] — Menu items are framed in square brackets, for example the [New&ldots;] command in the [Project] menu opens a new project.

"OK" — Names of windows, dialogs and buttons such as "Message Window", "Dearchivate Project" or "OK" are shown in inverted commas.

CTRL — Names of keys on the PC keyboard are represented in the text with uppercase letters (e.g. ENTER, CTRL, C, F1, etc.).

CTRL + C — For some functions you need to press two keys simultaneously. For example, press and hold down the CTRL key and also press the C key. This is represented in the text as CTRL + C.

If "click" or "double-click" is mentioned, this always applies to the left-hand mouse button. If the right-hand mouse button is to be used, this will be explicitly mentioned.

11.3 Contacting Festo

Contact	
Address	Festo AG & Co. KG Ruiter Straße 82 73734 Esslingen Germany
Tel.	++49 (0)711 / 347 - 0
Fax	++49 (0)711 / 347 - 21 44
Internet	www.festo.com
E-mail	service_international@festo.com

11.4 How Diagnostic Information Is Retrieved

Due to internal mechanisms in the CPX system, it is not possible to use the event-based organisational blocks like OB 85 or OB 86 for generating all available diagnostic messages.

Instead, the required information is retrieved via the CPX's diagnostic I/O interface, which consists of two bytes each for input and output data. Since this interface cannot be accessed by several blocks in the PCS 7 application in parallel, communication needs to be routed from the module driver blocks through the terminal driver block (see the Description of RACK_CPX for details).

This routing of diagnostic information, combined with time needed by the terminal to response on a module driver block's diagnostic request, causes a slight delay of the diagnostic data. So, diagnostic information is never fully synchronous, and very short diagnostic events that only last for a couple of milliseconds (depending on the CPU's cycle time) might not be recognized correctly. The respective messages would then not be signalled on the Operator Station or the respective channel driver block's QUALITY output signal.

For "broken wire" messages of analog 4..20 mA-signals, synchronous processing is possible if certain configuration settings are made in HW Config. Please see the Invalid Value section at the description of the CHAI_CPX channel driver block for details.

In the hardware catalog in HW Config, several entries for the CPX-FB13 Profibus node are available, but only the one named "**CPX-FB13: DP-Slave [DPV1]**" provides the input and output ranges for the diagnostic interface. It must therefore be ensured that the terminal's first slot is configured with this entry.

In any other case, the DIAG_IW and DIAG_QW parameters of RACK_CPX will not be configured

correctly by the "Generate module drivers" function. RACK_CPX will then trigger a "Configuration error" message as soon as the system is started. Signal handling will function properly, but diagnostics cannot be provided until the terminal's configuration is corrected in HW Config. It is also necessary to run the "Generate Module Drivers" function after changing the hardware configuration in order to correctly configure the parameters of RACK_CPX.

11.5 Troubleshooting

Problem	Possible Cause	Solution
Error message during "Generate module drivers" run: "The 0/64 object (order number CPX-FB13 [...]) will not be processed"	CPX-FB13 Profibus node is not configured correctly in HW Config: I/O diagnostic interface (STI) is missing	Make sure to use the "**CPX-FB13: DP-Slave [DPV1]**" entry from the GSD file.
Error message during "Generate module drivers" run: "The FB3x PNIO Modul [...] object (order number [...]) will not be processed"	CPX-FB3x Profinet IO node is not configured correctly in HW Config: I/O diagnostic interface (STI) is missing	Make sure to use the "**FB3x PNIO Modul [STI]**" entry from the GSDML file.
"Device x/y/z: Multiple alarm (OB82)" message upon undervoltage	Station parameter "Filter alarm Vout/Vven" not set to "active" in HW Config	Set filter parameter to "active" (HW Config/DP Slave Properties/Parameter Assignment/Device Specific Parameters).
Analog input or output modules do not provide proper data (implausible and/or heavily changing values)	Byte order set to "Motorola" in HW Config, but hardware does not support changing the byte order yet.	a) Change byte order to "Intel" (HW Config/DP Slave Properties/Parameter Assignment/Device Specific Parameters) or b) Replace the CPX-FB13 fieldbus hardware by rev. 23 or newer. Please check the System Requirements chapter for more information about how to find out the revision of your hardware.
Online help cannot be accessed by pressing the F1 key	Installed PCS 7 version does not support HTML help format	Install PCS 7 version 7.1 or newer.