CPX terminal

FESTO

Electronics manual

CPX-Front-End Controller

Type CPX-FEC

Manual
538 475
en 0404NH
[677 480]
Contents and general instructions

MODBUS™ Registered trade mark of Schneider Automation
Microsoft Windows™ Registered trade mark of Microsoft Corporation
Microsoft Internet Explorer™ Registered trade mark of Microsoft Corporation
Contents and general instructions

Contents

Designated use .......................................................... VII
Target group ............................................................... IX
Service ................................................................. I
Notes on the use of this manual .................................... X
Important user instructions ........................................ X

1. System summary ...................................................... 1
   1.1 Use the future of automation technology today ............... 1
   1.2 The operating modes of the CPX-FEC ........................... 1
      1.2.1 Stand Alone ........................................... 1
      1.2.2 Remote Controller .................................. 1
      1.2.3 Remote I/O ........................................... 1
      1.2.4 Overview of the operating modes ......................... 1
   1.3 Programming the CPX-FEC ..................................... 1
   1.4 Meaning of the LED displays .................................. 1
      1.4.1 Fault displays of the LEDs ......................... 1
   1.5 Tutorial: Creating a project for the CPX-FEC with FST (Stand Alone) ............. 1

2. Installation ............................................................ 2
   2.1 General notes on installation .................................... 2
   2.2 Setting the switches of the FEC ............................... 2
      2.2.1 Rotary switch ...................................... 2
      2.2.2 DIL switches ...................................... 2
   2.3 Ethernet interface .............................................. 2
      2.3.1 Ethernet cable ..................................... 2
   2.4 Programming interface (RS232) ............................... 2
      2.4.1 Pin assignment ..................................... 2
      2.4.2 Connecting a programming PC .................... 2
      2.4.3 Connecting the Front End Display FED-50/90 .......... 2
   2.5 Service interface for handheld ................................. 2
   2.6 Compliance with protection class IP65/IP67 .................. 2
### Contents and general instructions

3. **General operation (Stand Alone)** ................................................. 3-1

3.1 Configuration ................................................................. 3-4
  3.1.1 Create connection to CPX-FEC ........................................ 3-4
  3.1.2 Creating a project ..................................................... 3-5
  3.1.3 Fast configuration by loading the actual configuration .......... 3-6
  3.1.4 Manual configuration with the Hardware Configurator .......... 3-10

3.2 Addressing ................................................................. 3-19
  3.2.1 Default addressing .................................................... 3-19
  3.2.2 Individual addressing ................................................ 3-20
  3.2.3 Addressing example .................................................. 3-22

3.3 Parametrizing .............................................................. 3-27
  3.3.1 Parametrizing with the handheld ................................... 3-29
  3.3.2 Parametrizing with the Hardware Configurator of the FST ..... 3-29
  3.3.3 Parametrizing from a user program ................................. 3-32
  3.3.4 Forcing ................................................................. 3-35
  3.3.5 Application example for the parametrizing ....................... 3-38

3.4 Programming the CPX-FEC ................................................. 3-39
  3.4.1 Overview of operands ................................................. 3-40
  3.4.2 Overview of operations .............................................. 3-41
  3.4.3 Overview of function modules for the CPX terminal .......... 3-45
  3.4.4 User programs ........................................................ 3-46
  3.4.5 Program example (STL) .............................................. 3-50

3.5 Diagnosis ................................................................. 3-52
  3.5.1 Diagnosis with the controller LEDs RUN, STOP and ERROR .. 3-52
  3.5.2 Diagnosis with the Hardware Configurator ....................... 3-55
  3.5.3 Diagnosis in the user program .................................... 3-59

3.6 Access and control via CI commands ..................................... 3-61

4. **With the CPX terminal to the Ethernet** ................................. 4-1

4.1 Ethernet in automation technology ..................................... 4-4
  4.1.1 Why Ethernet? ......................................................... 4-4

4.2 Basic principles of the Ethernet ........................................... 4-5
  4.2.1 IP addressing in the Ethernet ...................................... 4-6
  4.2.2 Data exchange on layers 1 to 4 .................................... 4-6
4.2.3 Data exchange in the application layers 5 to 7 ........................................ 4-1
4.3 Ethernet with CPX-FEC ................................................................. 4-13
  4.3.1 Connecting the CPX-FEC to the Ethernet .................................. 4-13
  4.3.2 IP addressing of the CPX-FEC .................................................. 4-13
  4.3.3 HTTP .......................................................... 4-16
  4.3.4 TFTP, Telnet .......................................................... 4-16
  4.3.5 E-mails can be sent with the CPX-FEC (SMTP driver) .......... 4-18
  4.3.6 Example of a program for sending an e-mail ......................... 4-19
4.4 The Webserver of the CPX-FEC ................................................. 4-21
  4.4.1 Possibilities and limits of the Webserver .............................. 4-21
  4.4.2 Standard Web pages of the CPX-FEC ................................. 4-21
  4.4.3 Creating own Web pages for the CPX-FEC ......................... 4-24
  4.4.4 Load files into the Webserver ............................................. 4-25
5. Remote Controller mode ................................................................. 5-1
  5.1 Remote Controller Ethernet ..................................................... 5-4
    5.1.1 Configuration .................................................. 5-4
    5.1.2 Operation as Remote Controller ....................................... 5-6
    5.1.3 Diagnosis with the controller LEDs RUN, STOP, ERROR and TP 5-10
  5.2 Remote Controller Field bus .................................................. 5-18
    5.2.1 Configuration .................................................. 5-18
    5.2.2 Configuration example and communication sequence .......... 5-21
    5.2.3 Parametrizing .................................................. 5-24
    5.2.4 Diagnosis .................................................. 5-25
6. Remote I/O Ethernet ................................................................. 6-1
  6.1 General information ................................................................. 6-4
  6.2 Modbus/TCP: Commands and addressing .................................. 6-1
    6.2.1 Modbus commands and address assignment ....................... 6-1
    6.2.2 CPX status information (group A) ................................... 6-1
    6.2.3 Processing data (groups B and D) ................................... 6-10
    6.2.4 Diagnostic memory (groups C and E) ............................... 6-24
    6.2.5 Objects (group F) .................................................. 6-26
    6.2.6 Addressing examples for Modbus/TCP ............................... 6-27
### Contents and general instructions

6.3 EasyIP ................................................................. 6-32
6.4 Diagnosis ............................................................. 6-33
  6.4.1 Diagnosis with the control and Ethernet LEDs ............... 6-33

A. Technical appendix .................................................. A-
  A.1 Technical specifications of the Front End Controller CPX-FEC .......... A-

B. Drivers ............................................................... B-
  B.1 Drivers ............................................................ B-
    B.1.1 Drivers for the CPX-FEC (operating mode Remote Controller) .... B-
    B.1.2 Drivers for the CPX-FEC (operating mode Remote I/O Ethernet) .. B-

C. Function modules .................................................... C-
  C.1 General function modules for the CPX-FEC ......................... C-
    C.1.1 Fault message .................................................. C-
  C.2 Modules for special functions of the CPX-FEC ..................... C-
    C.2.1 Modules for the SMTP driver (send e-mail) ................... C-
    C.2.2 Fault codes .................................................... C-

D. The Command Interpreter .......................................... D-
  D.1 The Command Interpreter (CI) .................................. D-
    D.1.1 Connection to a dialog device ................................ D-
    D.1.2 Selecting the command interpreter (Login) ................. D-
    D.1.3 Exiting the command interpreter ............................ D-
    D.1.4 CI command ................................................... D-
    D.1.5 Displaying operands and statuses with Display (D) ....... D-
    D.1.6 Changing operands with Modify (M) ....................... D-
    D.1.7 Commands for program controller .......................... D-
    D.1.8 Commands for forcing inputs and outputs ................... D-
    D.1.9 Initialising user memory ..................................... D-
    D.1.10 Password ...................................................... D-
    D.1.11 Driver-specific commands .................................. D-
    D.1.12 Linking CI commands ....................................... D-

VI
Contents and general instructions

E. Index ................................................................. E-
The CPX-FEC (Front-End Controller) described in this manual is intended exclusively for use as follows:

- for controlling pneumatic and electric actuators (valves and output modules)
- for interrogating electric sensor signals through the input modules
- communication via Ethernet

The CPX terminal may only be used as follows:

- as intended in industrial installations.
- without any modifications by the user. Only the conversions or modifications described in the documentation supplied with the product are permitted.
- in faultless technical condition.

The maximum values specified for pressures, temperatures, electrical data, torques, etc. must not be exceeded.

If used together with additional commercially available components, such as sensors and actuators, the specified limits for pressures, temperatures, electrical data, torques etc. must be observed.

Observe also the standards specified in the relevant chapters, as well as national and local laws and technical regulations.
Contents and general instructions

Target group

This manual is intended exclusively for technicians trained in control and automation technology who have experience in installing, commissioning, programming and diagnosing slaves on the Ethernet and the relevant field bus protocols.

Service

Please consult your local Festo repair service if you have any technical problems.
Contents and general instructions

Notes on the use of this manual

Please note
This manual refers to the following versions:

<table>
<thead>
<tr>
<th>Hardware/software</th>
<th>Version 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC</td>
<td>– as from software status 31.03.2004</td>
</tr>
<tr>
<td>Festo Software Tools (FST)</td>
<td>– Version 4.1 or higher</td>
</tr>
<tr>
<td>CPX-FB06 Interbus 2)</td>
<td>– as from software status 11.07.2003</td>
</tr>
<tr>
<td>CPX-FB11 DeviceNet 2)</td>
<td>– as from software status 06.10.2003</td>
</tr>
<tr>
<td>CPX-FB13 PROFIBUS-DP 2)</td>
<td>– as from software status 26.02.2004</td>
</tr>
<tr>
<td>CPX-FB14 CANopen 2)</td>
<td>– as from software status 22.08.2003</td>
</tr>
<tr>
<td>CPX-FB23 CC-Link 2)</td>
<td>– as from software status 07.08.2003</td>
</tr>
</tbody>
</table>

1) Software status (SW) see type plate
2) For operating mode Remote Controller Field bus

Tab. 0/1: Hardware and software states for this manual

Note Tab. 0/2 for further manuals on the CPX terminal.
Contents and general instructions

Important user instructions

Danger categories
This manual contains instructions on the possible dangers which may occur if the product is not used correctly. These instructions are marked (Warning, Caution, etc.), printed on a shaded background and marked additionally with a pictogram. A distinction is made between the following danger warnings:

**Warning**
This means that failure to observe this instruction may result in serious personal injury or damage to property.

**Caution**
This means that failure to observe this instruction may result in personal injury or damage to property.

**Please note**
This means that failure to observe this instruction may result in damage to property.

The following pictogram marks passages in the text which describe activities with electrostatically sensitive components.

Electrostatically sensitive components may be damaged if they are not handled correctly.
Marking special information

The following pictograms mark passages in the text containing special information.

**Pictograms**

- **Information:** Recommendations, tips and references to other sources of information.

- **Accessories:** Information on necessary or sensible accessories for the Festo product.

- **Environment:** Information on environment-friendly use of Festo products.

**Text markings**

- The bullet indicates activities which may be carried out in any order.

1. Figures denote activities which must be carried out in the numerical order specified.

   - Hyphens indicate general activities.
## Contents and general instructions

<table>
<thead>
<tr>
<th>Type</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronics manual</td>
<td>“System manual” type P.BE-CPX-SYS-...</td>
<td>Overview of structure, components and method of operation of CPX terminals</td>
</tr>
<tr>
<td></td>
<td>“CPX field bus node” type P.BE-CPX-FB...</td>
<td>Instructions on fitting, installing, commissioning and diagnosing the relevant field bus node</td>
</tr>
<tr>
<td></td>
<td>“CPX I/O modules” type P.BE-CPX-EA-...</td>
<td>Notes on connection and instructions on fitting, installing and commissioning input and output modules of type CPX-..., the MPA pneumatic modules as well as of MPA, CPA and Midi/Maxi pneumatic interface</td>
</tr>
<tr>
<td></td>
<td>“CPX analogue I/O modules” type P.BE-CPX-AX-...</td>
<td>Notes on connection and instructions on fitting, installing and commissioning CPX analogue I/O modules</td>
</tr>
<tr>
<td></td>
<td>“CPX-CP interface” type P.BE-CPX-CP-...</td>
<td>Instructions on commissioning and diagnosing CPX terminals with the CP interface type CPX-CP-4-FB</td>
</tr>
<tr>
<td></td>
<td>“Handheld” type P.BE-CPX-MMI-1-...</td>
<td>Instructions on commissioning and diagnosing CPX terminals with the handheld type CPX-MMI-1</td>
</tr>
<tr>
<td></td>
<td>“CPX-FEC” type P.BE-CPX-FEC-...</td>
<td>Instructions on fitting, installing, commissioning and diagnosing the CPX Front-End Controller.</td>
</tr>
<tr>
<td>Software package</td>
<td>“FST”</td>
<td>Programming in Statement List and Ladder Diagram for the FEC</td>
</tr>
<tr>
<td>Manual pneumatics</td>
<td>“Valve terminals with MPA pneumatics” type P.BE-MPA-...</td>
<td>Information on fitting, installing and commissioning MPA pneumatics (type 32)</td>
</tr>
<tr>
<td></td>
<td>“Valve terminals with CPA pneumatics” type P.BE-CPA-...</td>
<td>Information on fitting, installing and commissioning CPA pneumatics (type 12)</td>
</tr>
<tr>
<td></td>
<td>“Valve terminals with Midi/Maxi pneumatics” type P.BE-MIDI/MAXI-03-...</td>
<td>Instructions on fitting, installing and commissioning Midi/Maxi pneumatics (type 03)</td>
</tr>
</tbody>
</table>

Tab. 0/2: Manuals on the CPX terminal
The following **product-specific** terms and abbreviations are used in this manual:

<table>
<thead>
<tr>
<th>Term/abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0&lt;sub&gt;h&lt;/sub&gt;</td>
<td>Hexadecimal numbers are marked by a subscript “h”.</td>
</tr>
<tr>
<td>AI</td>
<td>Analogue input</td>
</tr>
<tr>
<td>AO</td>
<td>Analogue output</td>
</tr>
<tr>
<td>CP modules</td>
<td>Common term for the various modules which can be incorporated in a CPX terminal</td>
</tr>
<tr>
<td>CPX terminal</td>
<td>Complete system consisting of CPX modules with or without pneumatics.</td>
</tr>
<tr>
<td>DI</td>
<td>Digital input</td>
</tr>
<tr>
<td>DIL switch</td>
<td>Dual-in-line switches usually consist of several switch elements with which settings can be made.</td>
</tr>
<tr>
<td>DO</td>
<td>Digital output</td>
</tr>
<tr>
<td>Field bus node</td>
<td>Provides the connection to certain field buses. Transmits control signals to the connected modules and monitors their functioning.</td>
</tr>
<tr>
<td>FW</td>
<td>Flag word</td>
</tr>
<tr>
<td>Handheld / MMI</td>
<td>Hand-operated device for commissioning and for servicing purposes</td>
</tr>
<tr>
<td>IB, IW</td>
<td>Input byte, input word</td>
</tr>
<tr>
<td>I/O modules</td>
<td>Common term for CPX modules which provide digital inputs and outputs.</td>
</tr>
<tr>
<td>I/Os</td>
<td>Digital inputs and outputs</td>
</tr>
<tr>
<td>OB, OW</td>
<td>Output byte, output word</td>
</tr>
<tr>
<td>PLC/IPC</td>
<td>Programmable logic controller/industrial PC</td>
</tr>
<tr>
<td>Pneumatic interface</td>
<td>The pneumatic interface is the interface between the modular electrical peripherals and the pneumatics.</td>
</tr>
</tbody>
</table>

Tab. 0/3:   Product-specific terms and abbreviations
The following **network-specific** terms and abbreviations are used in this manual:

<table>
<thead>
<tr>
<th>Term/abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP</td>
<td>Creates the relationship between the physical Ethernet address and the logical IP address (Address Resolution Protocol)</td>
</tr>
<tr>
<td>BOOTP</td>
<td>Dynamic protocol for automatic assignment of IP addresses (Boot protocol, predecessor of DHCP)</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic protocol for automatic assignment of IP addresses (Dynamic Host Configuration Protocol)</td>
</tr>
<tr>
<td>EasyIP</td>
<td>Protocol for the fast exchange of operands between Festo controllers (e.g. FEC Standard, PS1, etc.).</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Physical protocol and network for connecting various devices</td>
</tr>
<tr>
<td>FTP, TFTP</td>
<td>Protocol for datatransmission via TCP/IP (File Transfer Protocol, Trivial File Transfer Protocol)</td>
</tr>
<tr>
<td>HTTP</td>
<td>Protocol for data transmission via TCP/IP (Hyper Text Transfer Protocol)</td>
</tr>
<tr>
<td>ICMP</td>
<td>Exchange of fault and status information of the network (Internet Control Message Protocol)</td>
</tr>
<tr>
<td>IP</td>
<td>Protocol for addressing and delivering data (Internet Protocol)</td>
</tr>
<tr>
<td>ISO/OSI layer model</td>
<td>Standardized model for data transmissions (7 layers)</td>
</tr>
<tr>
<td>MAC address</td>
<td>Fixed assigned address for Ethernet device (Media Access Control)</td>
</tr>
<tr>
<td>Modbus/TCP</td>
<td>Communication standard via TCP/IP in automation technology</td>
</tr>
<tr>
<td>TCP</td>
<td>Protocol for data transport and protection (Transfer Control Protocol)</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Combination of the protocols TCP and IP, the most-widely used protocol in communication via Ethernet.</td>
</tr>
<tr>
<td>UDP</td>
<td>Simplified protocol for data transport without test mechanisms (User Datagram Protocol)</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
</tbody>
</table>

Tab. 0/4: Network-specific terms and abbreviations
System summary

Chapter 1
1. System summary

Contents

1. System summary ................................................................. 1

1.1 Use the future of automation technology today ..................... 1

1.2 The operating modes of the CPX-FEC ............................. 1

1.2.1 Stand Alone ......................................................... 1

1.2.2 Remote Controller ............................................... 1

1.2.3 Remote I/O ....................................................... 1

1.2.4 Overview of the operating modes ................................ 1

1.3 Programming the CPX-FEC ............................................. 1

1.4 Meaning of the LED displays ........................................... 1

1.4.1 Fault displays of the LEDs .................................... 1

1.5 Tutorial: Creating a project for the CPX-FEC with FST (Stand Alone) .... 1
1. System summary

Contents of this chapter

This chapter provides an overview of the three operating modes of the CPX-FEC. A tutorial provides a first summary of the most important configuration steps with the software package FST 4.1.

Further information

Detailed information on the CPX terminal and on the FST can be found here:

- manuals on the CPX terminal: CPX system manual, descriptions of the I/O modules, descriptions of the field bus node
- the FST manual (2 volumes, supplied with the FST software)
- text book “Automation with FST” (supplied with the FST software)
- manuals on the pneumatics (MPA, CPA, CPV, Midi/Maxi)
1. System summary

1.1 Use the future of automation technology today

With the CPX-FEC, the CPX terminal offers today the technology of tomorrow for automation tasks. Here is an overview of the possibilities which are offered:

- The CPX-FEC can control a CPX terminal independently (Stand Alone). Use the Festo Software Tool FST 4.1 with the Hardware Configurator for commissioning, programming and diagnosis.

- It can control a CPX terminal and tell you via Ethernet what is running. You can easily access current data via your company’s network. (Remote Controller Ethernet)

- It fits into your existing automation system without problems: Let the CPX-FEC as a PLC control the CPX terminal and at the same time communicate with your existing field bus (at present: PROFIBUS, Interbus, DeviceNet, CANopen and CC-Link). (Remote Controller Field bus)

- Control the CPX terminal straight away via Ethernet, the CPX-FEC makes this possible: In the operating mode “Remote I/O Ethernet” the CPX terminal is a field bus slave on the Ethernet.

A Webserver is incorporated in the CPX-FEC. With a standard Web Browser you can access current data. You can transfer Web pages which you have created into the CPX-FEC in order to perform more complex tasks from a self-created user interface level.

You can set up the CPX-FEC so that it sends e-mails when previously defined events occur. This can be e. g. a message about a diagnostic case or the processing of a certain task.
1. System summary

1.2 The operating modes of the CPX-FEC

The CPX-FEC can be operated in various operating modes depending on the demand:

- Stand Alone
- Remote Controller
- Remote I/O

The operating mode can be set with DIL switch 1 (see section 2.2.2).

You can connect a handheld of type CPX-MMI to the CPX-FEC in any operating mode. The handheld serves for fast preliminary commissioning, parametrizing and diagnosis.
1. System summary

1.2.1 Stand Alone

Fig. 1/1: CPX terminal controlled by the CPX-FEC

**Stand Alone**

The CPX terminal is controlled **independently by the CPX-FEC**. There is no communication connection to other slaves.

Configure the CPX terminal with the FST software package FST 4.1 or higher. The programming or Ethernet interface can be used for the configuration.

Fig. 1/2: Software package FST 4.1 for configuration
1. System summary

1.2.2 Remote Controller

In this operating mode the CPX terminal is controlled by the CPX-FEC (as in Stand Alone), but there is also a communication connection with a higher-order controller. Communication can take place via Ethernet or via a field bus:

**Remote Controller Ethernet**

The CPX terminal is controlled by the CPX-FEC. The FEC also communicates via Ethernet (Modbus TCP, EasyIP) with a higher-order controller. You can use IT technology for the controller, diagnosis and communication (e.g. Webserver, e-mail, ...). An advantage of this operating mode is that you only need to use one bus system.

Configure the CPX terminal with FST via the Ethernet interface.

**Remote Controller Field bus**

The CPX terminal is controlled by the CPX-FEC. The FEC also communicates via the field bus with a higher-order controller. A field bus node (with DIL switch position for “Remote Controller”) must be installed in the CPX terminal for this operating mode.

Configure the CPX terminal with FST.
1. System summary

1.2.3 Remote I/O

![Ethernet diagram]

1. CPX terminal controlled via Ethernet

**Remote I/O Ethernet**

The CPX terminal is controlled by a higher-order controller via Ethernet. The CPX-FEC has the function of an Ethernet field bus slave. In this way you can combine your CPX terminal with existing Ethernet automation systems (e.g. Schneider Electric).

You can use the integrated Webserver for control, diagnosis and communication. An advantage of this operating mode is that you only need to use one bus system.
1. System summary

1.2.4 Overview of the operating modes

<table>
<thead>
<tr>
<th>Function of the CPX-FEC</th>
<th>Stand Alone</th>
<th>Remote Controller Ethernet</th>
<th>Remote Controller Field bus</th>
<th>Remote I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC controlled by</td>
<td>Controller</td>
<td>Control and communication</td>
<td>Ethernet slave</td>
<td></td>
</tr>
<tr>
<td>PLC used</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Communication with higher-order controller</td>
<td>No</td>
<td>Via Ethernet</td>
<td>Via field bus 1)</td>
<td>Via Ethernet</td>
</tr>
<tr>
<td>Webserver</td>
<td>Possible</td>
<td>Possible</td>
<td>Possible</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>FST 4.1 or higher</td>
<td>FST 4.1 or higher</td>
<td>Higher-order controller (e.g. with Schneider Unity)</td>
<td></td>
</tr>
<tr>
<td>Parametrizing with</td>
<td>FST/Handheld</td>
<td>FST/Handheld</td>
<td>FST/Handheld</td>
<td>Handheld/Modbus</td>
</tr>
<tr>
<td>Module code</td>
<td>208</td>
<td>208</td>
<td>210</td>
<td></td>
</tr>
<tr>
<td>Addressing</td>
<td>Default (can be modified)</td>
<td>Default (can be modified)</td>
<td>Specified</td>
<td></td>
</tr>
<tr>
<td>Memory location 2)</td>
<td>250 kB (PLC)</td>
<td>250 kB (PLC)</td>
<td>550 kB (Webserver)</td>
<td>800 kB (Webserver)</td>
</tr>
<tr>
<td>Switch position DIL 1</td>
<td>DIL 1.1: OFF</td>
<td>DIL 1.1: OFF</td>
<td>DIL 1.1: OFF</td>
<td>DIL 1.1: OFF</td>
</tr>
<tr>
<td>Handheld</td>
<td>Can be connected to the CPX-FEC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) A field bus node in the operating mode “Remote Controller” must also be installed here in the CPX terminal.
2) The Webserver can occupy free memory space in the PLC if extra memory space is required.

Tab. 1/1: Overview of the operating modes
1. System summary

1.3 Programming the CPX-FEC

Use the software package FST 4.1 or higher in order to program the CPX-FEC. Programming is carried out in the languages Statement List (STL) or Ladder Diagram (LDR). Further information can be found in the relevant chapters for the different operating modes.

Fig. 1/3: Programming with FST 4.1

You can find detailed information on programming the CPX-FEC in the following documentation:

- the FST manual (supplied with the FST software)
- the text book “Automation with FST” (supplied with the FST software).
1. System summary

1.4 Meaning of the LED displays

The LEDs on the cover indicate the operating status of the CPX-FEC and are arranged in two groups.

1 Control and Ethernet LEDs (see chapters 3, 5 and 6):
RUN
STOP
ERROR
TP

2 CPX-LEDs (see next section):
PS: Power system
PL: Power load
SF: System fault
M: Modify

Fig. 1/4: Two groups of the LEDs of the CPX-FEC
1. System summary

1.4.1 Fault displays of the LEDs

In the following only the **general** CPX-LEDs PS, PL, SF and M are explained. Information on the **controller and Ethernet LEDs** RUN, STOP, ERROR and TP can be found in chapters 3, 5 and 6.

<table>
<thead>
<tr>
<th>PS (Power System) – power sensor/logic supply</th>
<th>LED (green)</th>
<th>Sequence</th>
<th>Status</th>
<th>Meaning/fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED lights up</td>
<td>ON OFF</td>
<td>No fault Operating voltage/sensor supply applied</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>LED flashes</td>
<td>ON OFF</td>
<td>Operating voltage/sensor supply outside the tolerance range</td>
<td>Eliminate undervoltage</td>
<td></td>
</tr>
</tbody>
</table>
|                                             | ON OFF      | Internal fuse for the operating voltage/sensor supply has responded. | 1. Eliminate short circuit/overload on module side.  
2. Depending on the parametrizing of the module (module parameter):  
   • The sensor supply voltage will be switched on again **automatically** when the short circuit has been eliminated (default)  
   • Power Off/On is necessary |
| LED is out                                  | ON OFF      | The operating voltage/sensor supply is not applied | Check the operating voltage connection of the electronics |

Tab. 1/2: Fault diagnosis with the LED PS
1. System summary

| PL (Power Load) – power load supply (outputs/valves) |
|-------------|-----------------|-----------------|-----------------|
| LED (green) | Sequence | Status | Meaning/fault treatment |
| LED lights up | ON ON OFF OFF | No fault Load voltage applied | None |
| LED flashes | ON ON OFF OFF | Load voltage at the system supply or additional supply outside the tolerance range | Eliminate undervoltage |

Tab. 1/3: Fault diagnosis with the LED PL

| SF (System Fail) – system fault |
|-----|-----------------|-----------------|-----------------|
| LED (red) | Sequence | Status | Meaning/fault treatment |
| LED is out | ON OFF | No fault | – |
| LED flashes once | ON ON OFF OFF | Simple fault/information (fault class 1) | See description of fault numbers in the CPX system manual |
| LED flashes twice | ON ON ON OFF OFF | Fault (fault class 2) | |
| LED flashes three times | ON ON ON ON OFF OFF | Serious fault (fault class 3) | |

1) The system fault LED flashes depending on the class of fault which has occurred.
   Fault class 1 (simple fault): one flash, pause
   Fault class 2 (fault): flash twice, pause
   Fault class 3 (serious fault): flash three times, pause

Tab. 1/4: Fault diagnosis with the LED SF
1. System summary

<table>
<thead>
<tr>
<th>M (Modify) – Force active</th>
<th>LED (yellow)</th>
<th>Sequence</th>
<th>Status</th>
<th>Meaning/fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>–</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LED flashes</td>
<td>ON OFF ON OFF</td>
<td>Force is active</td>
<td>The Force function is enabled (see system parameter Force mode; function no. 4402).</td>
</tr>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>–</td>
<td>The CPX-FEC has no parameter “System start”, so that the status “LED lights up” does not exist with the M-LED.</td>
</tr>
</tbody>
</table>

Tab. 1/5: Messages of the LED M

Information on the controller and Ethernet LEDs RUN, STOP, ERROR and TP can be found in chapters 3...6.
1. System summary

1.5 Tutorial: Creating a project for the CPX-FEC with FST (Stand Alone)

This tutorial presents the most important steps for commissioning a CPX terminal with FEC. The tutorial deals only with the operating mode Stand Alone. Read the detailed introduction in the following chapters and in the FST manual.

**Caution**

If you have connected a CPX terminal to your PC for configuration: Test projects and programs at first without active actuators or without compressed air. You will then avoid damage in the test phase.

1. Install your CPX terminal with CPX-FEC in accordance with chapter 2 and set the rotary switch of the CPX-FEC to “0”.

2. Use the programming cable to connect the CPX-FEC to the serial interface of your PC.

3. Start the FST 4.1 and select [Extras][Preferences...] “Communication” register: “Use RS232” Confirm with OK.

4. Select [Project][New...] and assign a project name.

5. Select the CPX-FEC as controller in the window “Project Settings”.

6. Open the Hardware Configurator with a double click on “I/O Configuration” in the project window.

7. Select “Actual-Nominal-Comparison” in the context menu (right-hand mouse click) of the Hardware Configurator. Accept the configuration by clicking on “Apply”.
1. System summary

Fig. 1/5: Carry out the actual-nominal comparison

Fig. 1/6: Apply the actual-nominal comparison
1. System summary

Fig. 1/7: A configured CPX terminal

8. If necessary, parametrize the CPX system or individual modules:

Fig. 1/8: Example: Parametrizing an input module
Creating a program

1. Select [Program] [New...]

2. Select the programming language (e.g. Statement List) in the window “New program”.

3. Confirm the next window without modifications (Type program, Number 0, Version 1).

4. Create a program:

Fig. 1/9: Program window of the FST
1. System summary

5. When you first use operands in your program, they will be automatically interrogated after being entered in the allocation list:

![Figure 1/10: Interrogation for an entry in the allocation list during programming](image)

6. The new program is marked for loading automatically in the project window. Load the project into the CPX-FEC with [Online][Download Project]:

![Figure 1/11: A marked program is loaded into the FEC with the project](image)
1. System summary

Start program 7. Set the rotary switch of the CPX-FEC to 1...F or start the program in the “Online TCPIP” display:

Fig. 1/12: Program start with the “Online TCPIP” display
Installation

Chapter 2
2. Installation

Contents

<table>
<thead>
<tr>
<th>2. Installation</th>
<th>2-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 General notes on installation</td>
<td>2-4</td>
</tr>
<tr>
<td>2.2 Setting the switches of the FEC</td>
<td>2-6</td>
</tr>
<tr>
<td>2.2.1 Rotary switch</td>
<td>2-6</td>
</tr>
<tr>
<td>2.2.2 DIL switches</td>
<td>2-6</td>
</tr>
<tr>
<td>2.3 Ethernet interface</td>
<td>2-11</td>
</tr>
<tr>
<td>2.3.1 Ethernet cable</td>
<td>2-11</td>
</tr>
<tr>
<td>2.4 Programming interface (RS232)</td>
<td>2-13</td>
</tr>
<tr>
<td>2.4.1 Pin assignment</td>
<td>2-13</td>
</tr>
<tr>
<td>2.4.2 Connecting a programming PC</td>
<td>2-14</td>
</tr>
<tr>
<td>2.4.3 Connecting the Front End Display FED-50/90</td>
<td>2-17</td>
</tr>
<tr>
<td>2.5 Service interface for handheld</td>
<td>2-18</td>
</tr>
<tr>
<td>2.6 Compliance with protection class IP65/IP67</td>
<td>2-19</td>
</tr>
</tbody>
</table>
2. Installation

<table>
<thead>
<tr>
<th>Contents of this chapter</th>
<th>In this chapter you will find information on:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Setting the switches of the CPX-FEC</td>
</tr>
<tr>
<td></td>
<td>Connecting devices, to field bus and Ethernet</td>
</tr>
<tr>
<td></td>
<td>The pin assignment of the interfaces</td>
</tr>
<tr>
<td></td>
<td>Compliance with protection class IP65/IP67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Further information</th>
<th>Further information can be found here:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fitting, installation and the power supply of the complete CPX terminal: CPX system manual (P.BE-CPX-SYS-..)</td>
</tr>
<tr>
<td></td>
<td>Installing I/O modules and pneumatic interface: Manual for the CPX pneumatic interface and CPX I/O modules (P.BE-CPX-EA-..)</td>
</tr>
<tr>
<td></td>
<td>Handheld: Manual for the universal handheld CPX-MMI-1 (P.BE CPX-MMI-1-..)</td>
</tr>
</tbody>
</table>

Instructions on installing the pneumatic components can be found in the relevant pneumatics manual (see Tab. 0/2).
2. Installation

2.1 General notes on installation

**Warning**
Before carrying out installation and/or maintenance work, switch off the following:
- the compressed air supply
- the operating voltage supply for the electronics/sensors
- the load voltage supply for the outputs/valves.

You will thereby avoid:
- uncontrolled movements of loose tubing
- uncontrolled movements of the connected actuators
- undefined switching states of the electronics.

**Caution**
The CPX-FEC contains electrostatically sensitive components.
- Do not touch the electrical contacts of the components.
- Observe the regulations for handling electrostatically sensitive components.

In this way you will avoid damage to the electronics.

Information on fitting the CPX terminal can be found in the CPX system manual (P.BE-CPX-SYS...).
2. Installation

Electrical connecting and display elements

The following connecting and display elements can be found on the CPX-FEC:

1. FEC status and CPX-specific LEDs
2. 16-element rotary switch (RUN/STOP, Program selection)
3. Ethernet interface (10/100BaseT, RJ45)
4. Programming interface (RS232, 9-pin sub-D socket)
5. 2-element DIL switch 1 (select operating mode)
6. 2-element DIL switch 2 (reserved)
7. Service interface for handheld (V24)

Fig. 2/1: Connecting and display elements on the CPX field bus node
Dismantling and fitting

The FEC is fitted in a manifold sub-base of the CPX terminal (see Fig. 2/2).

Dismantling

Dismantle the FEC as follows:

1. Loosen the four screws of the FEC with a Torx screwdriver size T10.

2. Pull the FEC carefully and without tilting away from the contact rails of the manifold sub-base.

Fig. 2/2: Dismantling/fitting the CPX-FEC
2. Installation

Fitting

Fit the FEC as follows:

1. Place the FEC in the manifold sub-base. Make sure that the grooves with the terminals for electrical contact on the bottom of the field bus node lie directly above the contact rails. Press the FEC carefully and without tilting as far as possible into the manifold sub-base.

2. Insert the screws so that the self-boring threads can be used. Tighten the screws at first only by hand. Tighten the screws with a Torx screwdriver size T10 with 0.9 ... 1.1 Nm.
2. Installation

2.2 Setting the switches of the FEC

2.2.1 Rotary switch

The 16-element rotary switch (0, 1, ..., F) serves as a RUN/STOP switch for the PLC.

Leave the rotary switch at position “0” (STOP) during installation.

Modifications to the position of the rotary switch are transmitted to the controller with a delay of 500 ms. In this way, you can switch between two switch positions without the intermediate positions having any effect.

<table>
<thead>
<tr>
<th>Rotary switch</th>
<th>Setting</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Rotary Switch" /></td>
<td>0</td>
<td>STOP&lt;br&gt;PLC program has stopped.&lt;br&gt;Programs stop when the rotary switch is modified from 1 ... F to 0.</td>
</tr>
<tr>
<td><img src="image" alt="Rotary Switch" /></td>
<td>1 ... F</td>
<td>RUN&lt;br&gt;User program started.&lt;br&gt;Programs start when the rotary switch is modified from 0 to 1 ... F.&lt;br&gt;The switch position can be interrogated in the user program and used for programming purposes.</td>
</tr>
</tbody>
</table>

Tab. 2/1: Possible settings of the rotary switch for the PLC
2. Installation

2.2.2 DIL switches

Please note
Check that the setting of the DIL switch is correct before commissioning the CPX-FEC.
A change in the operating mode is not recognized until after Power OFF/ON.

The DIL switches must be accessible in order that the CPX-FEC can be set:

- Remove an IP65/IP67 plug, if fitted, from the programming interface.

Setting the operating mode with DIL switch 1
You can set the operating mode of the FEC with switch element 1 of DIL switch 1.
2. Installation

<table>
<thead>
<tr>
<th>Operating mode</th>
<th>Setting DIL switch 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stand Alone / Remote Controller</strong>&lt;br&gt;Controlling the CPX terminal via the CPX-FEC:&lt;br&gt;– Stand Alone&lt;br&gt;or&lt;br&gt;– Ethernet Remote Controller&lt;br&gt;or&lt;br&gt;– Field bus Remote Controller</td>
<td>DIL 1.1: OFF&lt;br&gt;DIL 1.2: OFF</td>
</tr>
<tr>
<td><strong>Remote I/O</strong>&lt;br&gt;Controlling the CPX terminal via Modbus/TCP with a higher-order controller.</td>
<td>DIL 1.1: OFF&lt;br&gt;DIL 1.2: ON</td>
</tr>
<tr>
<td><strong>Reserved</strong></td>
<td>DIL 1.1: ON&lt;br&gt;DIL 1.2: ON</td>
</tr>
<tr>
<td><strong>Reserved</strong></td>
<td>DIL 1.1: ON&lt;br&gt;DIL 1.2: OFF</td>
</tr>
</tbody>
</table>

Tab. 2/2: Setting the operating mode

**Reserved DIL switch 2**

Leave the switch elements of DIL switch 2 set to OFF.
2. Installation

2.3 Ethernet interface

There is a RJ45 socket on the CPX-FEC for connection to the Ethernet.

<table>
<thead>
<tr>
<th>RJ45 socket</th>
<th>Pin</th>
<th>Signal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>TD+</td>
<td>Send data +</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>TD−</td>
<td>Send data −</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>RD+</td>
<td>Receive data +</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>RD−</td>
<td>Receive data −</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Metal covering</td>
<td>Screening/shield</td>
</tr>
</tbody>
</table>

Tab. 2/3: Pin assignment of the Ethernet interface

If the Ethernet interface is not used, seal it with a cover of type AK-RJ45, part no. 534496. You will then comply with protection class IP65/IP67.

2.3.1 Ethernet cable

Use as connecting cable:

Cable specification

Screened flexible Ethernet round cable of category 5;
max. outer diameter: 5.4 mm
core diameter: 0.89 ... 1.0 mm AWG24-26
ready made: Crimping pliers on RJ45

Please note

If the CPX terminal is fitted onto a moving part of a machine, the Ethernet cable on the moving part must be provided with strain relief.
2. Installation

Network connection

In order to connect your CPX-FEC to a network or PC you will require:

- a Patch cable for connecting via hub or switch
- a crossover cable for direct connection of the PC and the CPX-FEC.

**Please note**

Use the RJ45 plug from Festo in order to comply with protection class IP65/IP67.
- Type FBS-RJ45-8-GS, part no. 534494
Observe the fitting instructions for the plug.

![Fig. 2/3: RJ45 plug complying with protection class IP65/IP67](image-url)
2. Installation

2.4 Programming interface (RS232)

There is 9-pin sub-D socket on the CPX-FEC for connecting a programmer, a PC or FEDs.

2.4.1 Pin assignment

<table>
<thead>
<tr>
<th>Socket on the CPX-FEC</th>
<th>Pin</th>
<th>Signal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>RxD</td>
<td>Receive data</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>TxD</td>
<td>Send data</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>GND</td>
<td>Data reference potential</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>n.c.</td>
<td>not connected</td>
</tr>
<tr>
<td>Screening/shield</td>
<td></td>
<td>Screening/shield</td>
<td>Connection to functional earth</td>
</tr>
</tbody>
</table>

Tab. 2/4: Pin assignment of the programming interface

If the programming interface is not used, seal it with the transparent cover.

- type AK-SUB-9/15-B
- part no. 533334
- max. tightening torque 0.4 Nm
2. Installation

2.4.2 Connecting a programming PC

Use the programming cable from Festo in order to connect your PC to the CPX-FEC:

- type KDI-PPA-3-BU9
- part no. 151915

**Please note**
Use the above programming cable only for the programming environment. It has only one screening connection on one side and complies only with protection class IP20. Use an RS232 cable with screening/shield connection on both sides for communication or control.

The next section describes how to produce an RS232 cable which complies with protection class IP65/IP67.

**Please note**
Lay the cable
- not twisted
- without bends
- at a sufficient distance from current-carrying cables.
You will then avoid transmission faults between the PC and the FEC.
2. Installation

RS232 cable with protection class IP65/IP67

For an RS232 cable which complies with protection class IP65/IP67 you will require:

- a programming plug from Festo (fig. 2/4)
  type FBS-SUB-9-GS-1X9POL-B

- a 3-core screened cable.

Please note
Only the programming plug from Festo guarantees compliance with protection class IP65/IP67. If you also need to comply with protection class IP65/IP67 in the programming environment:

- Replace the two flat screws by threaded sleeves (type UNC 4-40/M3x6, part no. 533000) before connecting sub-D plugs of other manufacturers.

1. Fit a programming cable in accordance with the fitting instructions for the plug.

2. When fitting the plug onto the CPX-FEC tighten the fastening screws at first by hand and then with 0.4 Nm.

Please note
Observe the maximum permitted length of the cable as a factor of the baud rate in accordance with RS232 specifications.
2. Installation

1. Hinged cover with viewing window

2. Blanking plug if connection is not used

3. Clamp strap for screening connection

4. Terminal strip for the cable (pin assignment see Tab. 2/4)

5. Fastening screws

Fig. 2/4: Programming plug from Festo, type FBS-SUB-9-GS-1X9POL-B

**Please note**

- Use protective caps or blanking plugs to seal unused connections.

In this way you will comply with protection class IP65/IP67.
2. Installation

2.4.3 Connecting the Front End Display FED-50/90

You can connect an FED-50 or an FED-90 to the programming interface of the CPX-FEC. The FED is an easy-to-use display for operating and observing automation tasks at the field level. FED operating devices are intended for fitting in the front of a metallic housing.

Fig. 2/5: CPX terminal with FED-90

- Connect the FED to the CPX-FEC with connecting cable type FEC-KBG7 or FEC-KBG8 (part no. 539642 resp. 539643).
2. Installation

2.5 Service interface for handheld

The 5-pin M12 socket serves for connecting a handheld for fast preliminary commissioning, diagnosis or parametrizing.

Use only the following original cables for connecting the handheld to the service interface:

<table>
<thead>
<tr>
<th>Type</th>
<th>Cable length in [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>KV-M12-M12-3,5</td>
<td>3.5</td>
</tr>
<tr>
<td>KV-M12-M12-1,5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Tab. 2/5: Connecting cable for handheld

Information on the handheld can be found in the manual “Universal handheld type CPX-MMI-1”.

Fig. 2/6: CPX terminal with handheld
2. Installation

### 2.6 Compliance with protection class IP65/IP67

In order to comply with protection class IP65/IP67, seal unused sockets and the switch with the appropriate covers.

<table>
<thead>
<tr>
<th>Connection/switch</th>
<th>Connection IP65/IP67</th>
<th>Cover (^1) IP65/IP67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet, RJ45</td>
<td>Plug type FBS-RJ45-8-GS part no. 534494</td>
<td>Cover (^2) type AK-RJ45 part no. 534496</td>
</tr>
<tr>
<td>Programming interface, sub-D and DIL switch</td>
<td>Plug from Festo type FBS-SUB-9-GS-1X9POL-B part no. 534497</td>
<td>Transparent cover (^2) type AK-SUB-9/15-B part no. 533334</td>
</tr>
<tr>
<td>Service interface, M12</td>
<td>Connecting cable and plug for the handheld</td>
<td>Protective cap (^2) type ISK-M12 part no. 352059</td>
</tr>
<tr>
<td>Rotary switch</td>
<td>–</td>
<td>Cover (^2) type AK-RJ45 part no. 534496</td>
</tr>
</tbody>
</table>

\(^1\) If connection is not used  
\(^2\) Included in delivery

Tab. 2/6: Connections and covers for protection class IP65/IP67
2. Installation
General operation (Stand Alone)

Chapter 3
3. General operation (Stand Alone)

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Configuration</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Create connection to CPX-FEC</td>
<td>3-4</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Creating a project</td>
<td>3-7</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Fast configuration by loading the actual configuration</td>
<td>3-8</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Manual configuration with the Hardware Configurator</td>
<td>3-10</td>
</tr>
<tr>
<td>3.2</td>
<td>Addressing</td>
<td>3-19</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Default addressing</td>
<td>3-19</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Individual addressing</td>
<td>3-20</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Addressing example</td>
<td>3-21</td>
</tr>
<tr>
<td>3.3</td>
<td>Parametrizing</td>
<td>3-27</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Parametrizing with the handheld</td>
<td>3-29</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Parametrizing with the Hardware Configurator of the FST</td>
<td>3-29</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Parametrizing from a user program</td>
<td>3-34</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Forcing</td>
<td>3-35</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Application example for the parametrizing</td>
<td>3-38</td>
</tr>
<tr>
<td>3.4</td>
<td>Programming the CPX-FEC</td>
<td>3-39</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Overview of operands</td>
<td>3-40</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Overview of operations</td>
<td>3-41</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Overview of function modules for the CPX terminal</td>
<td>3-45</td>
</tr>
<tr>
<td>3.4.4</td>
<td>User programs</td>
<td>3-46</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Program example (STL)</td>
<td>3-50</td>
</tr>
<tr>
<td>3.5</td>
<td>Diagnosis</td>
<td>3-52</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Diagnosis with the controller LEDs RUN, STOP and ERROR</td>
<td>3-52</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Diagnosis with the Hardware Configurator</td>
<td>3-55</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Diagnosis in the user program</td>
<td>3-59</td>
</tr>
<tr>
<td>3.6</td>
<td>Access and control via CI commands</td>
<td>3-60</td>
</tr>
</tbody>
</table>
3. General operation (Stand Alone)

Contents of this chapter

This chapter is the basis for operating the CPX-FEC in the operating modes:

- Stand Alone
- Remote Controller (see chapter 5)

Here you will find a description of general commissioning, parametrizing, programming and diagnosis of the CPX-FEC.

If you operate the CPX-FEC without communication connection to other slaves, this corresponds to the operating mode **Stand Alone**. In this operating mode the FEC controls the CPX terminal independently.

Preparing for commissioning

- Installation in accordance with chapter 2
- The correct operating mode is set with DIL switch 1 (see section 2.2.2)
- FST 4.1 or higher is installed on the programming PC

Overview of the following commissioning steps

1. Configuring the CPX terminal with FST
2. If desired: Parametrize
3. Create program
4. Load project into CPX-FEC
5. Start and test program

Various diagnostic possibilities are available during commissioning and during operation.
3. General operation (Stand Alone)

3.1 Configuration

Use Festo Software Tools (FST 4.1 or higher) with the Hardware Configurator in order to configure your CPX terminal with CPX-FEC. In the following sections you will find the most important commissioning steps with FST 4.1.

**Caution**

If you have connected a CPX terminal to your PC for configuration: Test projects and programs at first without active actuators or without compressed air. You will then avoid damage in the test phase.

A tutorial on configuration with FST can be found in section 1.5. Detailed information on operating the FST can be found in the FST manual (type P.BE-FST-...).

Follow the instructions in order to carry out a first project with the CPX-FEC.

3.1.1 Create connection to CPX-FEC

You will require a connection between your PC and the CPX-FEC if you wish to load projects into the FEC or use the online functions.

You can use all online functions via the programming interface or the Ethernet interface. Access to the Web-Browser is only possible via the Ethernet interface.

Create connection to the CPX-FEC

1. Set the rotary switch of the CPX-FEC to “0”.
2. Connect your PC and the CPX-FEC either
   
   -- via the programming interface with an RS232 cable (see section 2.4) or
3. General operation (Stand Alone)

- via the Ethernet interface with a patch cable (connection via hub/switch) or with a crossover cable (direct connection) (see section 2.3).

3. Access the window “FST Preferences” in FST 4.1 with [Extras][Preferences...]. Select your type of connection in the register “Communication”.

- Programming interface “Use RS232”
  Set the local COM interface of your PC and the baud rate.

- Ethernet interface “Use TCP/IP”
  Set the IP address of your CPX-FEC or click on “Search” in order to display a list of the online available controllers. Select your controller with a double click.
  Information on the IP addressing can be found in section 4.3.2.

4. Confirm with OK.

3.1.2 Creating a project

Create a new project in FST as follows:

1. [Project][New...]

2. Assign a project name.

3. Select the CPX-FEC as controller in the window “Project Settings”.
3. General operation (Stand Alone)

Fig. 3/1: Select the CPX-FEC when creating a new project.

4. Start the Hardware Configurator with a double click on “IO Configuration” in the project window of the FST (see Fig. 3/2).
   If the project window is not displayed: Select [View] [Project Window].

5. The CPX configuration window and the Hardware Catalogue will be displayed.
   If the Hardware Catalogue is not displayed: Select [View] [Catalog].

Please note
Note the register “Download” in the “Controller settings”.
- Select “Download source files” if you wish to download the project later from the CPX-FEC (e.g. on another PC).
3. General operation (Stand Alone)

When a new project is created with the CPX-FEC, a CPX-FEC will automatically be numbered module 0:

Fig. 3/2: FST 4.1 with Hardware Configurator after creation of a new project

1. Project window
2. Hardware Configurator
3. Hardware Catalogue
4. Configuration table
3. General operation (Stand Alone)

3.1.3 Fast configuration by loading the actual configuration

If you have already connected a completely equipped and installed CPX terminal to your PC, there are two ways of loading the configuration:

- Actual-nominal comparison in the editor mode
- Change to the online mode

**Actual-nominal comparison**

1. Click on [Actual-Nominal-Comparison] in the context menu.
   The actual configuration of the connected CPX terminal will be loaded and displayed together with the nominal configuration.

2. You can decide which modules in the actual configuration you wish to transfer by selecting them with the check box in the first column.

3. In the lower range select under “Use actual settings for...” whether current parameter settings, the address assignment (mapping) or the Idle mode/Fail safe settings are to be transferred.

4. Click on “Apply” in order to transfer the selection to the current project.
   The Hardware Configurator remains in the editor mode.
3. General operation (Stand Alone)

Fig. 3/3: Load the configuration of a connected CPX terminal with the function “Actual-Nominal-Comparison” in the editor mode.

If the actual-nominal comparison does not lead to a result:

1. Check the communication settings of the FST under [Extras][Preferences...] register “Communication”.

2. Check the communication connection between the CPX-FEC and your PC.

Change to the online mode

Online mode

The second possibility of loading the current configuration is to change to the online mode.

1. Select the entry [Online] in the menu [Online] or click on “Online” in the context menu:
   - The configuration of the CPX terminal will be loaded and transferred to the Hardware Configurator.

2. Select [Save] in the context menu, in order to save the loaded configuration in your project.
   - The Hardware Configurator is now connected online with the CPX terminal.
3. General operation (Stand Alone)

3.1.4 Manual configuration with the Hardware Configurator

Use this type of configuration if you wish to configure first only on the PC when there is no connection to a CPX terminal. Configuration is carried out with drag & drop:

1. Pull the modules out of the Hardware Catalogue into the Hardware Configurator. You can pull the modules onto the graphic representation of the CPX terminal or directly into the configuration table. Configure the modules from left to right corresponding to the physical sequence in your CPX terminal.

2. The addresses will be assigned automatically in accordance with the default addressing. Modifying addresses: Click directly on the address in the configuration table and modify the value.

Deleting modules

- Mark the module to be deleted and press the “Del” button.
3. General operation (Stand Alone)

1. Configuration with drag & drop
2. Configured modules in the configuration table

Fig. 3/4: Manual configuration of the CPX terminal in the Hardware Configurator
Below you will find information on the CPX modules.

**Module CPX-FEC**
You can configure the controller of your CPX terminal with this module.

<table>
<thead>
<tr>
<th>Module</th>
<th>Module identifiers 1)</th>
<th>Assigned address range</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEC controller</td>
<td>FEC</td>
<td>1 word I</td>
<td>The inputs form the position of the rotary switch. The outputs are not used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 word O</td>
<td></td>
</tr>
</tbody>
</table>

1) Module identifiers on the handheld and in the Hardware Configurator of the FST 4.1

Tab. 3/1: Module CPX-FEC
3. General operation (Stand Alone)

Electric modules

<table>
<thead>
<tr>
<th>Module (order code)</th>
<th>Module identifiers 1)</th>
<th>Assigned address range</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital 4-input module (F: CPX-4DI)</td>
<td>4DI</td>
<td>1 word I</td>
<td>–</td>
</tr>
<tr>
<td>Digital 8-input module (I: CPX-8DI)</td>
<td>8DI</td>
<td>1 word I</td>
<td>–</td>
</tr>
<tr>
<td>Digital 4-output module (O: CPX-4DO)</td>
<td>4DO</td>
<td>1 word O</td>
<td>–</td>
</tr>
<tr>
<td>Digital multi I/O module (Y: CPX-8DE-8DO)</td>
<td>8DI/8DO</td>
<td>1 word I 1 word O</td>
<td>–</td>
</tr>
<tr>
<td>Analogue 2-input module (U: CPX-2AE-U/I)</td>
<td>2AI</td>
<td>2 words I</td>
<td>–</td>
</tr>
<tr>
<td>Analogue 2-output module (P: CPX-2AA-U/I)</td>
<td>2AO</td>
<td>2 words O</td>
<td>–</td>
</tr>
</tbody>
</table>

1) Module identifiers on the handheld and in the Hardware Configurator of the FST 4.1

Tab. 3/2: Overview of electric CPX modules

Technology module CP interface

With the aid of this module you can connect CP components to the CPX terminal.

<table>
<thead>
<tr>
<th>Module</th>
<th>Module identifiers</th>
<th>Assigned address range</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP interface</td>
<td>CPI</td>
<td>max. 8 words I max. 8 words O</td>
<td>The number of assigned addresses depends on the string assignment saved in the CP interface.</td>
</tr>
</tbody>
</table>

Tab. 3/3: Technology module CP interface
### Pneumatic modules

<table>
<thead>
<tr>
<th>Module (order code)</th>
<th>Module identifiers 1)</th>
<th>Assigned address range</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pneumatic interface for MPA valves (type 32)</td>
<td>–</td>
<td>–</td>
<td>This module is not displayed in the Hardware Configurator (passive module).</td>
</tr>
<tr>
<td>MPA pneumatic module (type 32) without electrical isolation (VI: CPX type 32: 1..V..)</td>
<td>MPA1S MPA2S</td>
<td>1 word O</td>
<td>–</td>
</tr>
<tr>
<td>MPA pneumatic module (type 32) with electrical isolation (VI: CPX type 32-G: 1..V..)</td>
<td>MPA1G MPA2G</td>
<td>1 word O</td>
<td>–</td>
</tr>
<tr>
<td>Pneumatic interface for CPA valves (type 12) (VI: CPX type 12: 1..V..)</td>
<td>CPA10/14</td>
<td>1...8 valve coils: 1 word O 1...16 valve coils: 1 word O 1...22 valve coils: 2 words O</td>
<td>The representation of the CPA and Midi/Maxi valves in the Hardware Configurator is only symbolic and can differ from the actual number of valves.</td>
</tr>
<tr>
<td>Pneumatic interface for Midi/Maxi valves (type 03) (VI: CPX type 03: 1..V..)</td>
<td>Type 03</td>
<td>1...8 valve coils: 1 word O 1...16 valve coils: 1 word O 1...24 valve coils: 2 words O 1...26 valve coils: 2 words O</td>
<td>–</td>
</tr>
</tbody>
</table>

1) Module identifiers on the handheld and in the Hardware Configurator of the FST 4.1

Tab. 3/4: Overview of pneumatic interfaces and MPA pneumatic modules
3. General operation (Stand Alone)

Configure the valves according to the pneumatic type used. The pneumatics must always be configured last.

**Valves of type 32 (MPA pneumatic modules)**

The individual MPA pneumatic modules each provide an electric module with digital outputs for controlling the valves. A word of outputs is always assigned for each MPA pneumatic module, irrespective of the number of valves fitted on the pneumatic module (see Tab. 3/4).

Further information on MPA pneumatic modules can be found in the manual for the CPX I/O modules (P.BE-CPX-EA-..) and in the relevant pneumatics manual.

**Valves of type 12 (CPA) and type 03 (Midi/Maxi)**

With these valves only the pneumatic interface is configured. The number of valve coils is set with a DIL switch (see manual CPX I/O modules, P.BE-CPX-EA-..).

**Configuration**

1. Pull the relevant pneumatic interface onto the next line in the configuration table.

2. Double click on the line. The window “Module...” will be displayed.

3. Switch to the register “Module” and select the number of valve coils as is set on the DIL switch in the pneumatic interface:
3. General operation (Stand Alone)

Fig. 3/5: Select the number of valve coils (configuration of the valves of type 12 or type 03)

If you set a value which is greater than the actual number of valve coils, the superfluous addresses will be reserved.

The representation of the CPA and Midi/Maxi valves in the Hardware Configurator is only symbolic and can differ from the actual number of valves. The same number of valves is displayed as the number of outputs which are set, irrespective of whether there are single-solenoid valves or double-solenoid valves in your CPX terminal.
3. General operation (Stand Alone)

Checking the configuration

A successful configuration is distinguished by:

- module assignment without gaps
- clear assignment of the master
- pneumatic modules are situated at the right-hand end of the CPX terminal

Check the configuration

- Click on “Check Configuration” in the context menu:
3. General operation (Stand Alone)

Reset configuration

All parameter settings and the address assignment (Mapping) can be reset to the factory settings.

Reset configuration

The desired settings can be reset to the factory settings as follows:

1. Select the command [Default Settings...] in the context menu of the Hardware Configurator. The following dialogue will then appear:

![Default Settings dialogue](image)

Fig. 3/6: Announcement for configuration check (example)

2. Select the settings which are to be reset and confirm this with OK.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Module parameter without Idle mode and Fail safe</td>
</tr>
<tr>
<td>Mapping</td>
<td>Address assignment of the CPX terminal</td>
</tr>
<tr>
<td>Idle mode/Fail safe</td>
<td>Idle mode and Fail safe parameters</td>
</tr>
<tr>
<td>System settings</td>
<td>System parameters and diagnostic memory parameters</td>
</tr>
</tbody>
</table>

Tab. 3/5: Options – Reset configuration
3. General operation (Stand Alone)

3.2 Addressing

Basic rules for addressing
- Modules occupy the address range in ascending order from left to right according to their physical arrangement in the CPX terminal (module-orientated).
- Addressing is carried out word-orientated (16-bit I/O). Different module types are assigned with different addresses depending on the address range.
- If a module occupies less than 1 word (16-bit I/O), the remaining bits of the word will not be used.
- If a module occupies more than 1 word I/O, the following word in the address area will be used.
- The address assignment can be modified module-orientated (see below).

3.2.1 Default addressing

In default addressing the address assignments of the various module types are specified in ranges. The following table shows the assignment:

<table>
<thead>
<tr>
<th>Module type</th>
<th>Input words</th>
<th>Output words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital inputs and outputs</td>
<td>0 ... 31</td>
<td>0 ... 31</td>
</tr>
<tr>
<td>Valves / pneumatic modules</td>
<td>–</td>
<td>32 ... 63</td>
</tr>
<tr>
<td>Analogue inputs and outputs</td>
<td>64 ... 127</td>
<td>64 ... 127</td>
</tr>
<tr>
<td>FEC, bus node, technology module</td>
<td>128 ... 255</td>
<td>128 ... 255</td>
</tr>
</tbody>
</table>

Tab. 3/6: Pre-assigned address ranges of default addressing
3. General operation (Stand Alone)

3.2.2 Individual addressing

You can modify the addressing individually if you wish to assign other addresses than those in the default addressing.

- Click directly on the address in the configuration table and modify the value.
  Make sure that the new address is not already assigned, otherwise you will receive a fault message.

<table>
<thead>
<tr>
<th>Module</th>
<th>Type</th>
<th>IW</th>
<th>OW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FEC - Controller</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>1</td>
<td>SDI - Input module</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>MOO - Output module</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>NPA1G - Pneumatic module</td>
<td>-</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>NPA1G - Pneumatic module</td>
<td>-</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>NPA1G - Pneumatic module</td>
<td>FST</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>NPA1G - Pneumatic module</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3/7: Modify address in the Hardware Configurator (here with fault message because address is already assigned)
3. General operation (Stand Alone)

3.2.3 Addressing example

**Example 1: CPX terminal with default addressing**

Fig. 3/8: Example 1: CPX terminal with digital and analogue I/O modules as well as MPA pneumatics
3. General operation (Stand Alone)

<table>
<thead>
<tr>
<th>Location</th>
<th>Module</th>
<th>I address</th>
<th>O address</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CPX-FEC</td>
<td>128</td>
<td>128</td>
<td>The inputs map the position of the rotary switch. The outputs are not used.</td>
</tr>
<tr>
<td>1</td>
<td>Digital 8-input module (8DI)</td>
<td>0</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>Digital 4-output module (4DO)</td>
<td>–</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>Analogue 2-input module (2AI)</td>
<td>64 65</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>Analogue 2-output module (2AO)</td>
<td>–</td>
<td>64 65</td>
<td>–</td>
</tr>
<tr>
<td>–</td>
<td>MPA pneumatic interface (type VMPA-FB-EPL-...)</td>
<td>–</td>
<td>–</td>
<td>Passive module, is not shown in the FST</td>
</tr>
<tr>
<td>5</td>
<td>MPA pneumatic module (CPX-type32: 1...8V)</td>
<td>–</td>
<td>32</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>MPA pneumatic module (CPX-type32: 1...8V)</td>
<td>–</td>
<td>33</td>
<td>–</td>
</tr>
<tr>
<td>7</td>
<td>MPA pneumatic module (CPX-type32: 1...8V)</td>
<td>–</td>
<td>34</td>
<td>–</td>
</tr>
</tbody>
</table>

Tab. 3/7: Configuration of example 1
3. General operation (Stand Alone)

<table>
<thead>
<tr>
<th>Addr. 1)</th>
<th>Module</th>
<th>Input word</th>
<th>Output word</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bit</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>0</td>
<td>8DI / 4DO</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>31</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>32</td>
<td>MPA1</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>33</td>
<td>MPA1</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>34</td>
<td>MPA1</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>35</td>
<td>MPA1</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>63</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>64</td>
<td>2AI / 2AO</td>
<td>2 Al (channel 0)</td>
<td>2 AO (channel 0)</td>
</tr>
<tr>
<td>65</td>
<td>2AI / 2 AO</td>
<td>2 Al (channel 1)</td>
<td>2 AO (channel 1)</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>127</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>128</td>
<td>CPX-FEC</td>
<td>–</td>
<td>8 DI 2)</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>255</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Dark grey: assigned and used  
Light grey: assigned but not used

1) Word number, automatically assigned by the default addressing  
2) Rotary switch of the CPX-FEC (interrogation of the switch position possible in user program)

Tab. 3/8: Address assignment for example 1 (default addressing)
3. General operation (Stand Alone)

**Example 2:**
CPX terminal with modified addressing (Mapping)

For the 2nd. addressing example the addresses have been modified so that inputs and outputs or valves lie together in blocks. It is then easier e.g. to understand the composition of machines (sensors, outputs and valves) in the addressing table.
3. General operation (Stand Alone)

<table>
<thead>
<tr>
<th>Location</th>
<th>Module</th>
<th>I address</th>
<th>O address</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CPX-FEC</td>
<td>128</td>
<td>128</td>
<td>The inputs map the position of the rotary switch. The outputs are not used.</td>
</tr>
<tr>
<td>1</td>
<td>Digital 8-input module (8DI)</td>
<td>0</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>Digital 4-output module (4DO)</td>
<td>–</td>
<td>10</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>Analogue 2-input module (2AI)</td>
<td>1, 2</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>Analogue 2-output module (2AO)</td>
<td>–</td>
<td>11, 12</td>
<td>–</td>
</tr>
<tr>
<td>–</td>
<td>MPA pneumatic interface</td>
<td>–</td>
<td>–</td>
<td>Passive module, is not shown in the FST</td>
</tr>
<tr>
<td>5</td>
<td>MPA pneumatic module (CPX-type32: 1-8V)</td>
<td>–</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>MPA pneumatic module (CPX-type32: 1-8V)</td>
<td>–</td>
<td>1</td>
<td>–</td>
</tr>
<tr>
<td>7</td>
<td>MPA pneumatic module (CPX-type32: 1-8V)</td>
<td>–</td>
<td>2</td>
<td>–</td>
</tr>
</tbody>
</table>

Tab. 3/9: Configuration of example 2
3. General operation (Stand Alone)

<table>
<thead>
<tr>
<th>Addr. 1)</th>
<th>Module</th>
<th>Input word</th>
<th>Output word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
<td>15 12 11 8 7 4 3 0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>8DI / MPA1</td>
<td>–</td>
<td>8 DI –</td>
</tr>
<tr>
<td>1</td>
<td>2AI / MPA1</td>
<td>2 Al (channel 0)</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>2AI / MPA1</td>
<td>2 Al (channel 1)</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>10</td>
<td>4DO</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>11</td>
<td>2AO</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>12</td>
<td>2AO</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>32</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>63</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>64</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>127</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>128</td>
<td>CPX-FEC</td>
<td>–</td>
<td>8 DI 2)  –</td>
</tr>
<tr>
<td>...</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>255</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Dark grey: assigned and used

Light grey: assigned but not used

1) Word number, automatically assigned by the default addressing
2) Rotary switch of the CPX-FEC (interrogation of the switch position possible in user program)

Tab. 3/10: Address assignment for example 2 (modified addressing)
3. General operation (Stand Alone)

3.3 Parametrizing

The CPX terminal is supplied with preset parameters. If required, you can set the reaction of the CPX terminal as well as the reaction of individual modules and channels by parametrizing.

A distinction is made between the following parametrizings:

- System parameters, e.g.: switching off fault messages, setting reaction times, etc.
- Module parameters (module and channel-specific), e.g.: Monitoring, settings in the event of faults, settings for Forcing
- Diagnostic memory parameters

A detailed description of the individual parameters and data as well as basic information on parametrizing can be found in the CPX system manual (P.BE-CPX-SYS-..).

The module parameters which are available for the various modules can be found in the manual for the relevant module (e.g. manual for CPX pneumatic interface and CPX I/O modules (P.BE-CPX-EA-..)).
3. General operation (Stand Alone)

Overview of parametrizing possibilities

You can parametrize a CPX terminal with FEC with the Hardware Configurator of the FST, with the handheld or via a user program.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Configurator of the FST:</td>
<td>Access to all parameters of the CPX</td>
<td>– Comfortable parametrizing via a PC</td>
</tr>
<tr>
<td><strong>offline</strong></td>
<td>terminal</td>
<td>– Parametrizing saved in the project</td>
</tr>
<tr>
<td>Hardware Configurator of the FST:</td>
<td>Access to all parameters and data</td>
<td>– Comfortable parametrizing via a PC</td>
</tr>
<tr>
<td><strong>online</strong></td>
<td>of the CPX terminal</td>
<td>– Parametrizing is saved only locally in the CPX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>terminal and is lost with Power OFF/ON</td>
</tr>
<tr>
<td>Handheld</td>
<td>Parametrizing is carried out via</td>
<td>– Parametrizing is saved only locally in the CPX</td>
</tr>
<tr>
<td></td>
<td>menu-guided entries with the</td>
<td>terminal and is lost with Power OFF/ON</td>
</tr>
<tr>
<td></td>
<td>handheld.</td>
<td></td>
</tr>
<tr>
<td>User programs</td>
<td>Access via function modules</td>
<td>– Program-controlled reaction</td>
</tr>
<tr>
<td></td>
<td>(see appendix C.1)</td>
<td>– Preprocessing possible</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Programming very costly</td>
</tr>
</tbody>
</table>

Tab. 3/11: Properties of the different parametrizing possibilities
3. General operation (Stand Alone)

3.3.1 Parametrizing with the handheld

The parameters of the CPX terminal can be read and modified via a connected handheld. The parametrizings with the Hardware Configurator of the FST and the handheld are equal, but the following applies:

Please note
Parametrizings via the handheld are saved only locally in the CPX terminal and are lost with Power OFF/ON.

Further information on parametrizing with the handheld can be found in the manual for the handheld P.BE-CPX-MMI-1-DE.

3.3.2 Parametrizing with the Hardware Configurator of the FST

You can parametrize the CPX terminal comfortably with the Hardware Configurator.

Parametrizing can be undertaken for:

- the nominal configuration (offline): Parameters are transferred when the project is loaded.

- the actual configuration (online): Parameters can be transferred immediately. However, parameters are saved only locally in the CPX terminal and are lost with Power OFF/ON.

When the relevant dialogue has been opened, you can view and modify the individual parameters.
3. General operation (Stand Alone)

Parametrizing when switching on

Please note
There is no system parameter “System start” with the CPX-FEC.

When the CPX terminal is switched on, parametrizing is carried out as “Start parametrizing” by the project saved in the CPX-FEC. Parametrizations can be carried out as test functions with the handheld. They are not saved in the project.

1. Parametrizing from the FEC project at start: The parameter set is distributed amongst modules

2. Parametrizing as a test function with the handheld

Fig. 3/10: Sequence of start parametrizing
3. General operation (Stand Alone)

Access to the various parameters is explained in the following.

System parameters

1. Click on “System Settings” in the context menu of the Hardware Configurator.

2. Set the system parameters in the register “System Parameters”:

![Image of system parameters settings](image_url)

Fig. 3/11: Set the system parameters with the Hardware Configurator
3. General operation (Stand Alone)

Module parameters

1. Double click on the module in the Hardware Configurator which you wish to parametrize.

2. Set the desired parameters in the register “Parameters”:

![Module parameters configuration](image)

Fig. 3/12: Set the module parameters with the Hardware Configurator
3. General operation (Stand Alone)

Diagnostic memory parameters

1. Click on “System Settings” in the context menu of the Hardware Configurator.

2. Set the diagnostic memory parameters in the register “Trace Parameters”.

![Diagnostic memory parameters with the Hardware Configurator](image)

Fig. 3/13: Set the diagnostic memory parameters with the Hardware Configurator
3. General operation (Stand Alone)

3.3.3 Parametrizing from a user program

If you wish to modify parameters of the CPX terminals via a program, you will require special function modules.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_ST_rd</td>
<td>Read CPX internal parameters and data</td>
</tr>
<tr>
<td>C_ST_wr</td>
<td>Write CPX internal parameters</td>
</tr>
<tr>
<td>C_MP_rd</td>
<td>Read general module parameters</td>
</tr>
<tr>
<td>C_MD_rd</td>
<td>Read module diagnostic data</td>
</tr>
<tr>
<td>C_AP_rd</td>
<td>Read analogue module parameters</td>
</tr>
<tr>
<td>C_AP_wr</td>
<td>Write analogue module parameters</td>
</tr>
</tbody>
</table>

Tab. 3/12: Function modules for parametrizing the CPX terminal

Further information on function modules and on parametrizing from a user program can be found in:
- the appendix C.1
- in the FST system manual under “Drivers and modules”.
3. General operation (Stand Alone)

3.3.4 Forcing

By forcing you can manipulate input and output signals. Actual input signals or status changes by program are ignored and replaced by the force values.

**Warning**
Depending on the functioning of the machine/system, the manipulation of signal states may cause serious injury to human beings or damage to property.
Be very careful when forcing in order to avoid damage.

Forcing inputs
Forcing an input does **not** modify the input signal itself and **cannot** be observed at the relevant status LED. The logical status of the input is modified internally and in some cases has an effect on the program. The forced input status is transferred to the processing image of the inputs. Therefore the online display of the FST shows the forced input signal.

Forcing outputs
In contrast, forcing an output does modify the actual output signal and can be observed at the relevant status LED. However, the forced output signal is **not** transferred to the processing image of the outputs. The online display of the FST does **not** show the forced, physical output signal, but the status from the processing image.
3. General operation (Stand Alone)

Please note
The online display of the FST always shows the valid signal status in the processing image. Therefore, observe the following when forcing:

- Forced input states are transferred to the processing image and therefore recognized by the controller. They are visible in the online display.
- Forced output states are not transferred to the processing image and are therefore not recognized by the controller. They are not therefore represented in the online display.

You have various possibilities of influencing input or output signals, e.g. for test purposes:

- Forcing via the parametrizing of the CPX terminal
- with the online display of the FST
- in the user program of the PLC.

If several functions are active at the same time, the following applies:

- force signals have the highest priority.

Forcing with the Hardware Configurator
Forcing is available via the module parametrizing providing FST online is connected to your CPX terminal.

Please note
If you block or enable forcing globally, all signal states which are forced will become ineffective or effective immediately.

1. Use the right-hand mouse button to click the module for which you wish to set the force parameters.
2. Select [Properties].
3. General operation (Stand Alone)

3. Select the register “Force Mode” (see following diagram).

4. Set the desired value or status for the relevant channel in the column “Force” [1].

5. Enable forcing for the relevant channel by placing a tick in the column “Channel” [2].

6. Activate the Force mode (global system parameter) as soon as you wish to accept the set values [3].

7. Accept the settings and close with OK [4].

Fig. 3/14: Example: Forcing outputs with the Hardware Configurator

Further information on forcing can be found in the CPX system manual and in volume 1 of the FST manual (e.g. Forcing with the online display).

Forcing via CI commands

Information on forcing with CI commands can be found in appendix D and in the chapter 2 “Basic functions of the FST software” in volume 1 of the FST manual.
3.3.5 Application example for the parametrizing

1. Input with default parametrizing

2. Input with reduction of the input debounce time and increase in the signal lengthening time

Fig. 3/15: Example of application for parametrizing debounce time and pulse lengthening (here on the right-hand sensor)

In the above application packets are transported on a fast-moving conveyor belt. With the following parametrizing the registering and processing of the signals has been improved.

- Reduction of the input debounce time of 3 ms (factory setting) to 0.1 ms: Shorter signals can also be registered (applies to the complete module).

- Modification of the signal lengthening time to 50 ms: The signal will be registered reliably by the controller. (applies only to the input channel of the 2nd. sensor).
3. General operation (Stand Alone)

3.4 Programming the CPX-FEC

Use the FST software version 4.1 or higher in order to create programs for the CPX-FEC, for testing and for commissioning.

User programs and program modules can be created in the programming languages Statement List (STL) and Ladder Diagram (LDR).

You will find here a description of the programming prerequisites, the programming languages, programming techniques, operands and operations:

- FST system manual
- Text book “Automation with FST” (supplied with the FST software)

If you are a beginner in the subject, you should refer to the relevant basic manuals for more information about the programming language.

When you have planned and structured a project, you can start with programming. The following are required:

- operands (retentive, non-retentive)
- operations (set of commands)
- function modules (depending on application).
3. General operation (Stand Alone)

3.4.1 Overview of operands

<table>
<thead>
<tr>
<th>Operand</th>
<th>Design.</th>
<th>Range</th>
<th>Retentive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input word</td>
<td>IW</td>
<td>IW0 ... IW255</td>
<td>No</td>
</tr>
<tr>
<td>Input bit</td>
<td>I</td>
<td>I0.0 ... I255.15</td>
<td>No</td>
</tr>
<tr>
<td>Output word</td>
<td>OW</td>
<td>OW0 ... OW255 can be addressed as word or bit</td>
<td>No</td>
</tr>
<tr>
<td>Output bit</td>
<td>O</td>
<td>O0.0 ... O255.15</td>
<td>No</td>
</tr>
<tr>
<td>Error word</td>
<td>EW</td>
<td>0: No fault, &gt; 0: Error number</td>
<td>No</td>
</tr>
<tr>
<td>Error status</td>
<td>E</td>
<td>0: No fault, 1: Fault</td>
<td>No</td>
</tr>
<tr>
<td>Initial execution flag</td>
<td>FI</td>
<td>0: Program cycle &gt; 1, 1: first program cycle (for each program)</td>
<td>No</td>
</tr>
<tr>
<td>Flag word</td>
<td>FW</td>
<td>FW0 ... FW9999</td>
<td>Yes</td>
</tr>
<tr>
<td>Flag bit</td>
<td>F</td>
<td>F0.0 ... F9999.15</td>
<td>Yes</td>
</tr>
<tr>
<td>Timer</td>
<td>T</td>
<td>T0 ... T255 (also TP and TW for each timer TON, TOFF)</td>
<td>No</td>
</tr>
<tr>
<td>Switch-on delay timer</td>
<td>TON</td>
<td>TON0 ... TON255</td>
<td>No</td>
</tr>
<tr>
<td>Switch-off delay timer</td>
<td>TOFF</td>
<td>TOFF0 ... TOFF255</td>
<td>No</td>
</tr>
<tr>
<td>Timer pre-setting</td>
<td>TP</td>
<td>TP0 ... TP255</td>
<td>Yes</td>
</tr>
<tr>
<td>Timer word</td>
<td>TW</td>
<td>TW0 ... TW255</td>
<td>No</td>
</tr>
<tr>
<td>Counter</td>
<td>C</td>
<td>C0 ... C255 (also as CV and CW for each counter)</td>
<td>Yes</td>
</tr>
<tr>
<td>Counter pre-setting</td>
<td>CP</td>
<td>CP0 ... CP255</td>
<td>Yes</td>
</tr>
<tr>
<td>Counter word</td>
<td>CW</td>
<td>CW0 ... CW255</td>
<td>Yes</td>
</tr>
<tr>
<td>Register</td>
<td>R</td>
<td>R0 ... R255</td>
<td>Yes</td>
</tr>
<tr>
<td>Function units</td>
<td>FU</td>
<td>FU0 ... FU31, FU39 ... FU255 per program</td>
<td>Yes</td>
</tr>
<tr>
<td>Function units</td>
<td>F</td>
<td>FU32 ... FU38 per program</td>
<td>No</td>
</tr>
<tr>
<td>Programs</td>
<td>P</td>
<td>P0 ... P63</td>
<td>No</td>
</tr>
<tr>
<td>Program status</td>
<td>PS</td>
<td>PS0 ... PS63</td>
<td>No</td>
</tr>
</tbody>
</table>
3. General operation (Stand Alone)

### Operands CPX-FEC

<table>
<thead>
<tr>
<th>Operand</th>
<th>Design.</th>
<th>Range</th>
<th>Retentive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function modules</td>
<td>CFM</td>
<td>CFM0 ... CFM99 (predefined by Festo)</td>
<td>No</td>
</tr>
<tr>
<td>Program modules</td>
<td>CMP</td>
<td>CMP0 ... CMP99 (user-defined)</td>
<td>No</td>
</tr>
</tbody>
</table>

Retentive operands retain their value even when the controller is switched off.

3.4.2 Overview of operations

Depending on the programming language selected (STL or LDR), various operands are available for creating a program. The permitted operations (set of commands) are listed below.

Refer to the FST system manual for a detailed description of the operations.
3. General operation (Stand Alone)

<table>
<thead>
<tr>
<th>STL</th>
<th>LDR</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP</td>
<td>–</td>
<td>Symbolic step marks are permitted for sequence programs</td>
</tr>
<tr>
<td>IF</td>
<td>(graphical)</td>
<td>Introduces a conditional part</td>
</tr>
<tr>
<td>THEN</td>
<td>(graphical)</td>
<td>Introduces an executive part providing the condition under IF is fulfilled.</td>
</tr>
<tr>
<td>OTHRW / ELSE</td>
<td>(graphical)</td>
<td>Introduces an alternative executive part if the condition under IF is not fulfilled.</td>
</tr>
<tr>
<td>SET</td>
<td>S</td>
<td>Single-bit operands are set to logical “1”, timers, counters or programs are started. Instruction follows in an executive part. The command is saved.</td>
</tr>
<tr>
<td>RESET</td>
<td>R</td>
<td>Single-bit operands are set to logical “0”, timers, counters or programs are stopped. Instruction follows in an executive part. The command is saved.</td>
</tr>
<tr>
<td>SHIFT</td>
<td>–</td>
<td>Swaps the specified single-bit operand with the value in the single-bit accumulator</td>
</tr>
<tr>
<td>LOAD</td>
<td>–</td>
<td>Single and multibit operands and constants are loaded into the accumulator with this command.</td>
</tr>
<tr>
<td>TO</td>
<td>TO</td>
<td>Transfers operand 1 to operand 2 in combination with LOAD</td>
</tr>
<tr>
<td>NOP</td>
<td>NOP</td>
<td>Empty operation Unconditional in conditional part (IF NOP), substitute symbol in executive part</td>
</tr>
<tr>
<td>N</td>
<td>/</td>
<td>Negation: Negates operands, i.e. they are interrogated for logical zero</td>
</tr>
<tr>
<td>AND</td>
<td>AND</td>
<td>Logical command for bit-by-bit AND linking</td>
</tr>
<tr>
<td>OR</td>
<td>OR</td>
<td>Logical command for bit-by-bit OR linking</td>
</tr>
<tr>
<td>EXOR</td>
<td>EXOR</td>
<td>Logical command for bit-by-bit exclusive OR linking</td>
</tr>
</tbody>
</table>

Tab. 3/14: Overview of operations of the CPX-FEC (part 1)
### General operation (Stand Alone)

<table>
<thead>
<tr>
<th>STL</th>
<th>LDR</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+, −, *, /</td>
<td>+, −, *, /</td>
<td>Arithmetical commands for addition, subtraction, multiplication and division</td>
</tr>
<tr>
<td>INC</td>
<td>I</td>
<td>The value of multibit operands is therefore increased by 1 (incremented)</td>
</tr>
<tr>
<td>DEC</td>
<td>D</td>
<td>The value of multibit operands is therefore reduced by 1 (decimated)</td>
</tr>
<tr>
<td>SWAP</td>
<td>SWAP</td>
<td>The higher-value and lower-value bytes in the multibit accumulator are swapped.</td>
</tr>
<tr>
<td>BID</td>
<td>BID</td>
<td>Converts the contents of the multibit accumulator from BINARY to BCD display.</td>
</tr>
<tr>
<td>DEB</td>
<td>DEB</td>
<td>Converts the contents of the multibit accumulator from BCD to BINARY display.</td>
</tr>
<tr>
<td>SHL</td>
<td>SHL</td>
<td>All bits in the multibit accumulator are shifted one position to the left. Bits pushed out to the left are lost.</td>
</tr>
<tr>
<td>SHR</td>
<td>SHR</td>
<td>All bits in the multibit accumulator are shifted one position to the right. Bits pushed out to the right are lost.</td>
</tr>
<tr>
<td>ROL</td>
<td>ROL</td>
<td>In the multibit accumulator all the bits are rotated to the left; the last bit becomes the first, the penultimate becomes the last, etc.</td>
</tr>
<tr>
<td>ROR</td>
<td>ROR</td>
<td>In the multibit accumulator all the bits are rotated to the right; the first bit becomes the last, the second becomes the first, etc.</td>
</tr>
<tr>
<td>INV</td>
<td>INV</td>
<td>Complements multibit operands according to the method of the 1st. complement.</td>
</tr>
<tr>
<td>CPL</td>
<td>CPL</td>
<td>Complements multibit operands according to the method of the 2nd. complement.</td>
</tr>
</tbody>
</table>

Tab. 3/15: Overview of operations of the CPX-FEC (part 2)
3. General operation (Stand Alone)

<table>
<thead>
<tr>
<th>STL</th>
<th>LDR</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Arithmetical comparison (smaller than ...)</td>
</tr>
<tr>
<td>&lt;=</td>
<td>&lt;=</td>
<td>Arithmetical comparison (smaller than or equal to ...)</td>
</tr>
<tr>
<td>=</td>
<td>=</td>
<td>Arithmetical comparison (equal to ...)</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;</td>
<td>Arithmetical comparison (equal to or larger than ...)</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>Arithmetical comparison (larger than ...)</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>&lt;&gt;</td>
<td>Arithmetical comparison (not equal to ...)</td>
</tr>
<tr>
<td>( )</td>
<td>[ ]</td>
<td>Open bracket, close bracket: Summary of other instructions</td>
</tr>
<tr>
<td>CFMn, CMPn</td>
<td>CFMn, CMPn</td>
<td>Access function modules (CFM0 ... CFM99) or program modules (CMP0 ... CMP99)</td>
</tr>
<tr>
<td>WITH</td>
<td>(graphical)</td>
<td>When modules are accessed, the parameter transfer is started (CMP ... WITH ...)</td>
</tr>
<tr>
<td>JMP TO</td>
<td>(via jump marks)</td>
<td>Jump to a jump mark, instruction follows in an executive part.</td>
</tr>
</tbody>
</table>

Tab. 3/16: Overview of operations of the CPX-FEC (part 3)
3. General operation (Stand Alone)

3.4.3 Overview of function modules for the CPX terminal

The run time library supplied with the FST contains a lot of ready-to-use function modules (CFMs). Some modules work especially for the CPX terminal.

Further information on function modules can be found
- in the appendix C.1
- in the FST system manual under “Drivers and modules”.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_ST_rd</td>
<td>Read CPX internal parameters and data</td>
</tr>
<tr>
<td>C_ST_wr</td>
<td>Write CPX internal parameters</td>
</tr>
<tr>
<td>C_MP_rd</td>
<td>Read general module parameters</td>
</tr>
<tr>
<td>C_MP_wr</td>
<td>Write general module parameters</td>
</tr>
<tr>
<td>C_AP_rd</td>
<td>Read special analogue module parameters</td>
</tr>
<tr>
<td>C_AP_wr</td>
<td>Write special analogue module parameters</td>
</tr>
</tbody>
</table>

Tab. 3/17: Function modules for parametrizing the CPX terminal

<table>
<thead>
<tr>
<th>Modules</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_STATUS</td>
<td>Interrogate diagnostic status</td>
</tr>
<tr>
<td>C_TR_rd</td>
<td>Read entries in diagnostic memory</td>
</tr>
<tr>
<td>C_MD_rd</td>
<td>Read module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 3/18: Function modules for diagnosing the CPX terminal
3. General operation (Stand Alone)

3.4.4 User programs

Create user programs for the CPX-FEC with the software package FST 4.1 or higher.

Create program

1. Select in the menu [Program][New...]

2. Select the programming language (Statement List or Ladder Diagram) in the window “New program”.

3. Select the following in the next window:
   - the program type: program module (CMP) or function module (CFM)
   - the number of the program
   - the version and comments.

Depending on the programming language selected, the STL or LDR editor will open.

4. Create your program in the editor.

Load the project into the CPX-FEC

Programs are always part of a project and will therefore be loaded into the CPX-FEC with the complete project.

Load the project into the CPX-FEC:

1. In the project window mark the program which you wish to load into the FEC:
3. General operation (Stand Alone)

2. Select [Online] [Download Project]. The project will be loaded into the FEC.

Start the program manually

3. Turn the rotary switch to 1 ... F or start the program (see FST system manual) either:
   - with the online control panel or
   - in the online display in the register “Programs” or
   - with a CI command.

Please note
Program 0 starts automatically when the project is loaded into the CPX-FEC, if the function “Autostart” is activated in the register “Download” in the controller settings. This also applies if the rotary switch is in the “0” position.

Stop program

4. In order to stop the program (see FST system manual):
   Turn the rotary switch to “0” or stop the program:
   - with the online control panel or
   - in the online display in the register “Programs” or
   - with a CI command.
3. General operation (Stand Alone)

Run time behaviour of the controller

There are various ways of influencing the start and stop behaviour of programs:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Possibilities</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotary switch CPX-FEC</td>
<td>0 ... F</td>
<td>0: Stop 1 ... F: Run Programs also start when the rotary switch is set to “0”. Programs also stop when the rotary switch is set to “0”. The behaviour depends therefore on the signal edge.</td>
</tr>
<tr>
<td>Controller settings in FST</td>
<td>Start/stop input</td>
<td>You can define an input as a start/stop input. Enter the input in the entry field. Programs do not start until the start/stop input has a 1-signal hat <strong>and</strong> the rotary switch is set to “Run”.</td>
</tr>
<tr>
<td>Reset programs</td>
<td></td>
<td>Defines whether all programs are to be reset (not only stopped), when the rotary switch is set to “0” (Stop) (negative edge).</td>
</tr>
<tr>
<td>Stop program</td>
<td></td>
<td>Defines the program which is to be started when the project is stopped (by rotary switch or online button). Enables a system to be put into a safe operating status after a stop. The stop program is not started if the controller is stopped due to a fault. “0” stands for “No stop program”.</td>
</tr>
<tr>
<td>Error output</td>
<td></td>
<td>You can define an output as an error output. Enter the output in the entry field. If there is a fault, the output will be set for as long as the fault lasts.</td>
</tr>
<tr>
<td>Error program</td>
<td></td>
<td>Defines the program which is to be started when there is a fault. “0” stands for “No error program”.</td>
</tr>
</tbody>
</table>

Tab. 3/19:  Set the run mode behaviour of the controller
3. General operation (Stand Alone)

![Controller Settings](image)

Fig. 3/16: Set the run mode behaviour of the controller

Further information can be found in the FST manual.
3. General operation (Stand Alone)

3.4.5 Program example (STL)

The following example shows an extract from a program for controlling a drilling machine.

<table>
<thead>
<tr>
<th>STEP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aplus</td>
<td>IF untensioned THEN RESET Aminus</td>
</tr>
<tr>
<td></td>
<td>AND Drill_abo THEN SET Aplus</td>
</tr>
<tr>
<td></td>
<td>AND Emergency THEN SET Start</td>
</tr>
<tr>
<td></td>
<td>THEN SET Aplus</td>
</tr>
<tr>
<td>Bplus</td>
<td>IF tensioned THEN SET Bohrer</td>
</tr>
<tr>
<td>Bminus</td>
<td>IF Drilled THEN RESET Bohrer</td>
</tr>
<tr>
<td>Aminus</td>
<td>IF Drill_abo THEN SET Aminus</td>
</tr>
<tr>
<td>Cplus</td>
<td>IF untensioned THEN SET Ejector</td>
</tr>
<tr>
<td>Cminus</td>
<td>IF Ejected THEN SET Ejector</td>
</tr>
</tbody>
</table>

Fig. 3/17: Program example Extract from a drilling machine controller

The symbolic operands ("Untensioned", "Drill_above", ...) will be linked to the absolute operands (I0.0, O0.0, ...) via the allocation list:
3. General operation (Stand Alone)

Fig. 3/18: Assignment list for the programming example from Fig. 3/17
3. General operation (Stand Alone)

### 3.5 Diagnosis

<table>
<thead>
<tr>
<th>Diagnostic possibility</th>
<th>Advantages</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEDs</td>
<td>Fast “on-the-spot” recognition of faults</td>
<td>See sections 1.4.1 and 3.5.1.</td>
</tr>
<tr>
<td>Handheld</td>
<td>Fast “on-the-spot” recognition of faults</td>
<td>See manual for the handheld: P.BE-CPX-MMI-1-..</td>
</tr>
<tr>
<td>Hardware Configurator</td>
<td>Online diagnosis without programming</td>
<td>See section 3.5.2</td>
</tr>
<tr>
<td>In the user program</td>
<td>– Error word EW &gt; 0</td>
<td>With function modules (see section 3.5.3 and appendix C.1)</td>
</tr>
<tr>
<td></td>
<td>– Complete diagnostic information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Program-controlled reaction to faults</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Display on FED/SCADA</td>
<td></td>
</tr>
</tbody>
</table>

Tab. 3/20: Overview of the diagnostic possibilities of the CPX-FEC
3. General operation (Stand Alone)

3.5.1 Diagnosis with the controller LEDs RUN, STOP and ERROR

Information on the general CPX LEDs PS, PL, SF and M can be found in section 1.4.1.
In the operating mode Stand Alone the controller LEDs have the following meanings:

<table>
<thead>
<tr>
<th>RUN</th>
<th>LED (green)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="LED lights up" /></td>
<td>ON</td>
<td>PLC program started</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="LED is out" /></td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STOP</th>
<th>LED (yellow)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="LED is out" /></td>
<td>ON</td>
<td>PLC program started</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="LED lights up" /></td>
<td>OFF</td>
<td>PLC program stopped</td>
</tr>
</tbody>
</table>

Tab. 3/21: Status displays of the LEDs RUN and STOP in the operating mode Stand Alone
### 3. General operation (Stand Alone)

**ERROR (only in the operating modes Stand Alone and Remote Controller)**

<table>
<thead>
<tr>
<th>LED (red)</th>
<th>Sequence</th>
<th>Status</th>
<th>Fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED is out</td>
<td></td>
<td>No fault</td>
<td></td>
</tr>
<tr>
<td>LED lights up</td>
<td>ON-OFF</td>
<td>PLC program fault</td>
<td>Read out fault code with FST or handheld (see section 5.2)</td>
</tr>
</tbody>
</table>

**TP (only in the operating modes Remote Controller Ethernet and Remote I/O)**

<table>
<thead>
<tr>
<th>LED (green)</th>
<th>Sequence</th>
<th>Status</th>
<th>Fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED lights up</td>
<td></td>
<td>Ethernet connection OK</td>
<td></td>
</tr>
<tr>
<td>LED flashes</td>
<td></td>
<td>Data traffic</td>
<td></td>
</tr>
<tr>
<td>LED is out</td>
<td></td>
<td>No Ethernet connection or Ethernet cable not connected</td>
<td>If necessary, check Ethernet connection</td>
</tr>
</tbody>
</table>

Tab. 3/22: Status displays of the LEDs ERROR and TP (Link/Traffic)
3. General operation (Stand Alone)

3.5.2 Diagnosis with the Hardware Configurator

With the Hardware Configurator you can carry out a complete diagnosis of the CPX terminal. For this the CPX terminal must be connected **online** to your PC: Diagnostic messages of the modules are displayed directly in the Hardware Configurator with an icon on the appropriate module:

1 Looking at the current diagnosis (context menu)  
2 Looking into the diagnostic memory (context menu)

Fig. 3/19: Warning icons as diagnostic message in the Hardware Configurator
3. General operation (Stand Alone)

Looking at the diagnostic message

- Double click on the module with the warning icon. Read the diagnostic message in the window “Module…”

Alternatively you can reach the diagnostic message as follows:

1. Click on “Properties” in the context menu of the module. The window “Module…” will be displayed.

2. Select the register “Diagnosis” and read the diagnostic message.

![Module #5](image)

Fig. 3/20: Read out the module and channel-related diagnostic message with the Hardware Configurator
3. General operation (Stand Alone)

Diagnosis Trace

- Click on “Diagnosis Trace” in the context menu of the Hardware Configurator. The “Trace-Memory” will be displayed:

![Diagram of Trace-Memory]

Fig. 3/21: Read out the Diagnosis Trace with the Hardware Configurator
3. General operation (Stand Alone)

Diagnosis with the online control panel

- Select [Online] [Control Panel].
  The online control panel will be displayed.

You can read out coded diagnostic information in the field “Error” in the online control panel. The field is composed as follows:

Error type, CPX error number, module number

![Online Control Panel](image)

Fig. 3/22: Coded diagnostic information in the online control panel of the FST

The illustration above shows as an example:
42 = CPX diagnosis
16 = module code not permitted
1 = module no. 1 registers the fault
3. General operation (Stand Alone)

3.5.3 Diagnosis in the user program

You can read out diagnostic information in your user program via function modules (CFM).

<table>
<thead>
<tr>
<th>Modules</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_STATUS</td>
<td>Interrogate diagnostic status</td>
</tr>
<tr>
<td>C_TR_rd</td>
<td>Read entries in diagnostic memory</td>
</tr>
<tr>
<td>C_MD_rd</td>
<td>Read module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 3/23: Function modules for diagnosing the CPX terminal

Further information on function modules can be found in appendix C.1.

Error program

If a fault occurs during running time, an error number will be entered in the error word (EW). Depending on whether an error program has been configured, the following applies:

- Error program = 0 (no error program defined): Programs will be stopped
- Error program > 0: Programs will be stopped and the error program with the entered number will be started.

The following example shows a program for error treatment. Enter it as “Error Program” in the register “Run mode” in the “Controller Settings” (see Fig. 3/16).
3. General operation (Stand Alone)

(Program 63 — fault treatment)

STEP 1

IF I0.7
THEN RESET E
LOAD V0
TO EW
RESET P63
SET P0

"Wait for fault quitting
‘Reset FEC Error
‘Error
‘Error word
‘Error quitting
‘General — organisation

Fig. 3/23: Example of a error program
3. General operation (Stand Alone)

3.6 Access and control via CI commands

The Command Interpreter (CI) enables simple external operation of the controller with a terminal or terminal emulator. The CI also represents the interface for online operation of the FST (e.g. for the Webserver).

Access via CI commands is available in this way with the CPX-FEC:

1. Create an online connection with the CPX-FEC.
2. Start your terminal program or select [Online] [Terminal] in the FST

![Terminal interface](image.png)

Fig. 3/24: Access to the Command Interpreter with the CI terminal of the FST

Further information on CI commands can be found here:

- in section 4.4: The Webserver of the CPX-FEC
- in appendix D: The Command Interpreter
3. General operation (Stand Alone)
With the CPX terminal to the Ethernet

Chapter 4
4. With the CPX terminal to the Ethernet

Contents

4. With the CPX terminal to the Ethernet ........................................... 4-1
  4.1 Ethernet in automation technology ............................................. 4-4
    4.1.1 Why Ethernet? ......................................................... 4-4
  4.2 Basic principles of the Ethernet .............................................. 4-5
    4.2.1 IP addressing in the Ethernet ....................................... 4-6
    4.2.2 Data exchange on layers 1 to 4 ..................................... 4-9
    4.2.3 Data exchange in the application layers 5 to 7 ..................... 4-11
  4.3 Ethernet with CPX-FEC ..................................................... 4-13
    4.3.1 Connecting the CPX-FEC to the Ethernet ............................ 4-13
    4.3.2 IP addressing of the CPX-FEC ..................................... 4-13
    4.3.3 HTTP ................................................................ 4-16
    4.3.4 TFTP, Telnet ........................................................... 4-17
    4.3.5 E-mails can be sent with the CPX-FEC (SMTP driver) .......... 4-18
    4.3.6 Example of a program for sending an e-mail ....................... 4-19
  4.4 The Webserver of the CPX-FEC .................................................. 4-21
    4.4.1 Possibilities and limits of the Webserver ............................. 4-21
    4.4.2 Standard Web pages of the CPX-FEC ................................. 4-22
    4.4.3 Creating own Web pages for the CPX-FEC ............................ 4-25
    4.4.4 Load files into the Webserver ......................................... 4-26
4. With the CPX terminal to the Ethernet

Contents of this chapter

If you wish to implement your automation tasks in future via Ethernet or if you wish to extend with Ethernet, you will meet with terms which may be new to you. You may have to ask network technicians and administrators for assistance in incorporating your CPX terminal in the network.

In this chapter we will give you some basic information on Ethernet. This concerns mainly the relation to automation technology as well as the performance and properties of the CPX-FEC. You should be familiar with some standard network terms.
4. With the CPX terminal to the Ethernet

4.1 Ethernet in automation technology

4.1.1 Why Ethernet?

Let us assume that you have a manufacturing plant with various controllers: conveyor belt controller, parts supply, parts storage, screwing station, testing station, packing station. All the controllers can function independently. If you wish to undertake a modification to this system or diagnose a fault, then you must take your programming PC and go from controller to controller: plug in – modify – test – unplug. Then go to the next controller: plug in – modify – test – unplug. Then go to the next controller ...

It would be much easier if the controllers were linked with Ethernet, as is made possible with the CPX-FEC. Each controller receives an IP address. You simply have to connect your programming PC to any point in the network in order to be able to access all the stations. You require only the cabling and the installation of the TCP/IP driver.

Fig. 4/1: Protocols and services of the CPX-FEC in the Ethernet
4. With the CPX terminal to the Ethernet

4.2 Basic principles of the Ethernet

Ethernet is a physical protocol which connects various devices in the network with each other. Ethernet transports Ethernet packages from a sender to one or several receivers. This transfer runs without confirmation of receipt and without repetition of the lost packages.

For reliable data communication additional logical protocols must therefore be placed on the Ethernet protocol. The most widely used protocol for data exchange is a combination of the protocols:

- TCP for data transport and security and
- IP for addressing and delivery.

Together this results in the familiar “TCP/IP” (Transport Control Protocol/Internet Protocol).

Ethernet and TCP/IP form the first 4 layers in the ISO/OSI layer model of network communication:

Layer 4: TCP (data transport and security)
Layer 3: IP (addressing and delivery)
Layer 2: Ethernet protocol (Ethernet data packages)
Layer 1: Ethernet hardware (fixed address (MAC-ID) per device)

Fig. 4/2: The first 4 layers of the OSI layer model with example of Ethernet TCP/IP
4. With the CPX terminal to the Ethernet

UDP

In addition to TCP, the transport protocol UDP (User Datagram Protocol) is of importance. While the TCP uses checksums and reply messages to ensure that the data reach the receiver correctly, the UDP sends the data with a more simple checksum, but without rely. This has the advantage of a faster exchange of data. Due to the lack of a reply, the correct transmission must be ensured, e.g. by means of a user program.

ICMP

The Internet Control Message Protocol ICMP serves for the exchange of fault and status information between power units and works on the same layer as IP.

4.2.1 IP addressing in the Ethernet

Due to the separation into logical and physical protocol layers (Ethernet and TCP/IP), there exist two types of addresses in a network:

- a fixed Ethernet address (MAC-ID) for each device and
- an IP address which is assigned to every device in the network.

ARP

Before the application, data are always sent to or received from an IP address. In order that the data reach the receiver, a correlation must be created between the logical IP address and the physical Ethernet address. The Address Resolution Protocol ARP is used for this purpose: An ARP table is saved in each network PC. This table lists the relevant physical Ethernet address for each IP address in the network. If an Ethernet address is not listed in the ARP table, the IP driver can ascertain it with the aid of an ARP request.
4. With the CPX terminal to the Ethernet

<table>
<thead>
<tr>
<th><strong>IP address</strong></th>
<th>An IP address as per standard IPv4 is usually specified with 4 decimal numbers separated by points (per 1 byte). <strong>Example of an IP address: 10.8.64.251</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With an IP address, a network as well as an individual slave in the network can be addressed. The IP address therefore contains</td>
</tr>
<tr>
<td></td>
<td>– the net ID (specifies the address of a network) and</td>
</tr>
<tr>
<td></td>
<td>– the host ID (specifies the address of an individual slave in this network).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Net mask</strong></th>
<th>The numbers in an IP address which represent the net ID and the host ID are now determined by the specification of a so-called “net mask”.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The telephone number of Festo Germany can be used as an example to explain the IP address and the net mask: 00497113470</td>
</tr>
<tr>
<td></td>
<td>Which of these numbers represents the dialling code and which represents the subscriber number becomes clear when you also know: “that the first 7 figures represent the dialling code, the last 4 represent the subscriber number”. That is the “net mask” for the telephone number above.</td>
</tr>
</tbody>
</table>

| **Net classes** | The net mask for IP addresses defines with “0” as a position marker, the bytes which are used for addressing the slave (host ID). Networks belong to different classes depending on the number of these bytes: |
4. With the CPX terminal to the Ethernet

<table>
<thead>
<tr>
<th>Net class</th>
<th>Net mask</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>255.0.0.0</td>
<td>Large network</td>
</tr>
<tr>
<td>B</td>
<td>255.255.0.0</td>
<td>Medium network</td>
</tr>
<tr>
<td>C</td>
<td>255.255.255.0</td>
<td>Small network with max. 254 slaves</td>
</tr>
</tbody>
</table>

Tab. 4/1: The most important net classes with the appropriate net masks (example)

Gateway

Networks with different Net-IDs will be connected to each other via a router or gateway. If a slave in a network is to send data to slaves in other networks, the IP address of the gateway must be specified.

**Three entries are therefore required for addressing in the Internet Protocol IP:**

- the IP address
- the IP net mask
- the IP address of the gateway.

Assigning addresses

The settings for the IP addressing can be configured manually on each end-of-line terminal.

DHCP

In large networks this is usually accomplished centrally and automatically with DHCP (Dynamic Host Configuration Protocol). Here a DHCP server manages a range of IP addresses and distributes them to the DHCP-capable end terminals. The predecessor of DHCP was the protocol BootP.
4. With the CPX terminal to the Ethernet

4.2.2 Data exchange on layers 1 to 4

Each data package always consists of a header and a data range. The protocols now build up on each other so that each data package of a protocol is transported in the data range of the next higher-order protocol.

**Basic sequence for sending data**

1. The data are packed into a TCP data package by the TCP/IP driver and therefore made suitable for transport. The TCP data package contains test and confirmation information in the header. In this way the receiver can later check that the data have been transmitted correctly.

2. The TCP data package is transferred in the data range of the IP package. The IP addresses of the sender and of the receiver are added to the header of the IP package. This is also accomplished by the TCP/IP driver. If data are sent with UDP instead of with TCP, the UDP data will also be transmitted in the data range of an IP package.

3. Before the IP package is sent on its way as part of an Ethernet package, the TCP/IP driver must ascertain to whom the Ethernet package is to be sent. With the aid of the “Address Resolution Protocol” (ARP) the driver ascertains the Ethernet address of the target IP address.
4. With the CPX terminal to the Ethernet

4. The Ethernet package is then sent. All receivers in the net continuously monitor the exchange of data. If a receiver recognizes his address in a package, he will load the package into his TCP/IP stack.

5. The TCP/IP driver of the receiver now gradually unpacks the package, checks it and makes the data available to the process. If the package is faulty, it will be sent again.

Increasing the efficiency of the network

Originally all network slaves in a local Ethernet network received all data packages, but processed only those which were addressed to them. If a network contains different ranges which do not normally have to communicate with each other (e.g. office and production ranges), overloading may occur, depending on the amount of data being sent.

Switch

Various network ranges can be separated by means of a so-called switch. A switch analyses the data packages and passes them on only to the slave addressed. This increases the data throughput of the network. Formerly hubs simply passed on the data packages, but nowadays switches are used. An Ethernet network therefore has a real-time facility.
4. With the CPX terminal to the Ethernet

4.2.3 Data exchange in the application layers 5 to 7

Supplementary to the first four layers of the ISO/OSI layer model (see section 4.2.2), the layers 5 to 7 define data exchange at the application level. Further protocols and services are defined here. These are responsible for data access by means of programs and applications.

| Layers 1, 2: Physical Ethernet |
| Layers 3, 4: TCP/IP, UDP/IP, ARP |
| Layers 5, 6, 7: Application data (HTTP, TFTP, Telnet, Modbus/TCP, EasyIP, ...) |

Fig. 4/4: Complete ISO/OSI layer model for Ethernet

The data from the application layers are packed into the TCP data range and then transmitted as shown in Fig. 4/3. The protocols Modbus/TCP and EasyIP, which are important for automation technology, also work here.
4. With the CPX terminal to the Ethernet

The following table explains further important protocols and services of the application layers:

<table>
<thead>
<tr>
<th>Protocol/service</th>
<th>Meaning</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
<td>Transmission of data e.g. in the format HTML for Web pages.</td>
</tr>
<tr>
<td>TFTP</td>
<td>Trivial File Transfer Protocol</td>
<td>Simple file transmission (simplified version of FTP)</td>
</tr>
<tr>
<td>Telnet</td>
<td>Terminal over Network</td>
<td>Possibility of direct access to a distant computer with a terminal program.</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
<td>Sending e-mails</td>
</tr>
<tr>
<td>Modbus/TCP</td>
<td>–</td>
<td>Communication standard via Ethernet-TCP/IP in automation technology. Open communication protocol based on the master-slave architecture.</td>
</tr>
<tr>
<td>EasyIP</td>
<td>–</td>
<td>Simple communication protocol via Ethernet-TCP/IP in automation technology. Open communication protocol based on the master-slave architecture.</td>
</tr>
</tbody>
</table>

Tab. 4/2: Protocols and services in the user layers
4. With the CPX terminal to the Ethernet

4.3 Ethernet with CPX-FEC

The CPX-FEC masters all protocols and services described in section 4.2.3.

4.3.1 Connecting the CPX-FEC to the Ethernet

The Ethernet interface of the CPX-FEC complies with standard 10BaseT/100BaseTX for 100 Mbit networks (RJ45 connection).

Network connection In order to connect the CPX-FEC to a network via a hub or switch, you will require (see section 4.3):

- a Patch cable for connecting via hub or switch
- a crossover cable for direct connection of the PC and the CPX-FEC.

4.3.2 IP addressing of the CPX-FEC

The Ethernet address (MAC-ID, see type plate) of each CPX-FEC is assigned at the factory. The FEC also requires an IP address in the network as described above.

IP addresses can also be assigned automatically in a network via DHCP or BootP. The CPX-FEC is DHCP-capable, this means that it receives the IP address from the DHCP server in the network. During booting the FEC asks the DHCP server for its IP address via the network. The FEC also supports the dynamic addressing via the older BOOTP.

IP addressing with FST

1. Connect your programming PC to the CPX-FEC via the programming interface or the Ethernet interface (see section 3.1.1 or 4.3.1).

2. Start the FST, open a project or create a new project.
4. With the CPX terminal to the Ethernet

3. Double click the entry “Driver Configuration” in the project window. The window “Driver Configuration” will be displayed.

4. Double click on the TCP/IP driver “TCP/IPCPX”.

5. Manual addressing: Enter the IP address, the net mask and the gateway address. If necessary, ask your network administrator for the values for the net mask and the gateway address.

   Automatic addressing (DHCP/BootP), enter:
   – the IP address: 0.0.0.0
   – the net mask and the gateway address are transmitted and automatically set by DHCP.

Fig. 4/5: Example: Settings for automatic addressing via DHCP or BootP

6. Load the project into the CPX-FEC in order that the addressing can become effective.

Please note
If you have loaded an incorrect IP address into the CPX-FEC by mistake, you can no longer modify this via the network. In this case use the communication via the programming interface (RS232).
4. With the CPX terminal to the Ethernet

IP addressing with the handheld (Remote I/O)

With the handheld the IP address of the CPX-FEC can be set only in the operating mode and only for the operating mode Remote I/O.

1. Select the module CPX-FEC on the handheld.
2. Select “Parameters”.
3. Manual addressing: Enter the IP address, the net mask and the gateway address. If necessary, ask your network administrator for the values for the net mask and the gateway address.

Automatic addressing (DHCP/BootP), enter:

- the IP address: 0.0.0.0
- the net mask and the gateway address are transmitted and automatically set by DHCP.
4. With the CPX terminal to the Ethernet

4.3.3 HTTP

Composition of an URL
With the URL you can specify the protocol and the address from which the Browser or the Client is to fetch the data. The following minimum entries are required:

```
protocol://hostname
```

Examples
http://www.festo.com

Instead of the host name you can enter directly an IP address, e.g. the address of your CPX-FEC, in the Browser:

Example
http://10.8.64.251
The Web page saved in the Webserver of the CPX-FEC will now be displayed.

Use URL with CPX-FEC
As an option to the above, you can enter supplementary specifications:

```
/pathname/filename?further_parameters
```

Path name and file name correpond to the directory and file structures on the Webserver. Further parameters are transferred to the Webserver with the question mark.

Example
http://10.8.64.251/main.htm?ci:maw9=511
Here a CI command is transferred to the CPX-FEC with “?ci:maw9=511”: Output word 9 is set to 511d (1FFh). More information on this can be found in section 4.4 (Webserver) and in appendix D (CI commands).
4. With the CPX terminal to the Ethernet

4.3.4 TFTP, Telnet

Projects and files for the Webserver are loaded into the CPX-FEC or read from there with TFTP.

The Telnet protocol is used when you wish to access the CPX-FEC directly with a terminal program. Here you can perform numerous operations by means of CI commands.

Further information can be found here:

- Webserver: Section 4.4
- CI commands: Appendix D
4. With the CPX terminal to the Ethernet

4.3.5 E-mails can be sent with the CPX-FEC (SMTP driver)

With the SMTP driver the CPX-FEC can send e-mails. With certain faults, for example, a fault message can be sent.

The following conditions must be fulfilled in order that the CPX-FEC can send mails:

- The SMTP driver and the string driver must be loaded.
- An e-mail account with a Mail Server (mail host) must be set up.

In the same way that you require an account with an Internet provider in order to send e-mails, you must set up an account with a Mail Server (mail host) for the CPX-FEC. If possible, this account should be in your local network, because the SMTP driver does not offer automatic repetition and only limited diagnostic possibilities.

The string driver provides the additional data format “String” for strings of characters. In these string variables you can save prepared texts for the e-mails (addressees, prepared messages for various diagnostic cases, etc.)

Information on drivers can be found in appendix B, information on modules can be found in appendix C.
4. With the CPX terminal to the Ethernet

4.3.6 Example of a program for sending an e-mail

This example program expects the following strings to have a fixed assignment:

<table>
<thead>
<tr>
<th>String no.</th>
<th>Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td><a href="mailto:ipc@somedomain.com">ipc@somedomain.com</a></td>
<td>E-mail address (sender)</td>
</tr>
<tr>
<td>11</td>
<td>Mail.somedomain.com</td>
<td>Mail host (name or IP address)</td>
</tr>
<tr>
<td>12</td>
<td><a href="mailto:destination@someotherdomain.com">destination@someotherdomain.com</a></td>
<td>E-mail address (receiver)</td>
</tr>
<tr>
<td>13</td>
<td>Message from the PLC/IPC</td>
<td>Mail reference</td>
</tr>
<tr>
<td>14</td>
<td>Hallo,</td>
<td>E-mail text (string 1)</td>
</tr>
<tr>
<td>15</td>
<td>Here a current e-mail from the PLC/IPC</td>
<td>E-mail text (string 2)</td>
</tr>
</tbody>
</table>
4. With the CPX terminal to the Ethernet

In the following program extract the user name and the mail host are first defined and then an e-mail is sent:

**Example**

```plaintext
STEP
"" Set user name and mail host
IF   NOP
THEN  CMP 30      'E–MAIL module
       WITH V1    "1: set name and mail host
       WITH V10   "string#10: user name
       WITH V11   "string#11: mail host
IF     FU32        'Parameter 1
      = V0
THEN   NOP

STEP
"" Send e-mail
IF   NOP
THEN  CMP 30      'E–MAIL module
       WITH V2    "2 send mail
       WITH V12   "string#12: Target address
       WITH V13   "string#13: Re:
       WITH V14   "string#14: Mail text
       WITH V2    "Number of strings with mail text (here 2)
IF     FU32        'Parameter 1
      = V0
THEN   NOP

STEP
"" Wait until E-mail is sent
IF   NOP
THEN  CMP 30      'E–MAIL module
       WITH V0    "0: Status interrogation
       WITH V20   "string#20: Reply from mail host
IF     FU32        'Parameter 2
      = V0
THEN   LOAD FU34   'Parameter 3
         TO FW34   'SMTP fault code
       LOAD FU35   'Parameter 4
         TO FW35   'SMTP additional fault code
```
4. With the CPX terminal to the Ethernet

4.4 The Webserver of the CPX-FEC

A Webserver is a computer which provides data in a network. This data can be accessed with the aid of a Webbrowser. The computer, which accesses the available data with the aid of a Webbrowser, is called a Client. The data are usually provided in HTML format. The Browser loads the data provided, e.g. the HTML pages of the Webserver, and displays them.

The driver for the Webserver is loaded automatically in the FST when a new CPX-FEC project is created (see section 3.1.1).

4.4.1 Possibilities and limits of the Webserver

Possibilities

- HTML pages, media files and Java-Applets can be loaded into the controller. All media formats and all representation elements of the HTML format are permitted. Access can be made to these data via the Ethernet interface with the aid of any Browser.

- CI commands can be incorporated in the HTML pages in HTML code. In this way the operands of the controller can be observed or modified with the aid of a Browser.

- With the aid of JavaScripts and Java-Applets, Web pages can be made dynamic (e.g. for representing processes).

- When an HTML page is accessed, CI commands can be added to the page names as an HTTP query.

- The driver WEB_SRVR already contains standard HTML pages. The standard Homepage is called Index.htm (see also Fig. 4/6).
4. With the CPX terminal to the Ethernet

Limits

- Web pages cannot be dynamically generated with CGI or PHP programming or similar.
- The Webserver provides the following memory space depending on the operating mode:
  - Remote Controller: approx. 550 kB
    The Webserver can occupy free memory space in the PLC if extra memory space is required. There is then less memory space available for an extension of the PLC programs.
  - Remote I/O: approx. 800 kB
- File names must conform with the name conventions of MS-DOS (8+3 characters).

Please note
Please note that a control task in the controller has priority over communication with devices which request data from the Webserver driver.

4.4.2 Standard Web pages of the CPX-FEC

The Webserver driver already contains standard Web pages. The standard Homepage is called Index.htm. If you specify the IP address of the CPX-FEC in the Browser:

- the standard Homepage Index.htm will be displayed, if no Web page with the name main.htm exists.
- the page main.htm will be displayed, if it exists.

The following diagram shows the standard homepage of the CPX-FEC:
4. With the CPX terminal to the Ethernet

![FESTO WebServer](image)

**Fig. 4/6: Standard homepage of the CPX-FEC**

The standard HTML pages offer read access to the operands of the controller. With the links on the standard homepage you can display the most important information on the CPX-FEC and the loaded project.
4. With the CPX terminal to the Ethernet

<table>
<thead>
<tr>
<th>Link</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>user homepage</td>
<td>Switches to your self-created homepage main.htm, if it exists. See section</td>
</tr>
<tr>
<td>online information</td>
<td>Displays states of the inputs, outputs, flag words, timers, counters and</td>
</tr>
<tr>
<td>– local inputs</td>
<td>registers. The pages are updated every 5 seconds.</td>
</tr>
<tr>
<td>– local outputs</td>
<td></td>
</tr>
<tr>
<td>– flagwords</td>
<td></td>
</tr>
<tr>
<td>– timers</td>
<td></td>
</tr>
<tr>
<td>– counters</td>
<td></td>
</tr>
<tr>
<td>– registers</td>
<td></td>
</tr>
<tr>
<td>– error status</td>
<td>Displays the fault status.</td>
</tr>
<tr>
<td>– program status</td>
<td>Displays the status of various programs.</td>
</tr>
<tr>
<td></td>
<td>The pages are updated every 5 seconds.</td>
</tr>
<tr>
<td>terminal mode</td>
<td>Starts a standard terminal program for direct access to the CPX-FEC via CI commands.</td>
</tr>
<tr>
<td>controller information</td>
<td>Gives information on the CPX-FEC: Controller type and versions status, loaded project, driver information.</td>
</tr>
</tbody>
</table>

- Click on “Index” on the pages in order to return to the homepage.
- Scroll through the list of operands with the links “PageUp” and “PageDown.”

Tab. 4/3: Function of the links on the standard homepage of the CPX-FEC Webserver
4. With the CPX terminal to the Ethernet

4.4.3 Creating own Web pages for the CPX-FEC

You can create your own Web Pages for your system for control, monitoring and diagnosis. You can load these Web pages into the CPX-FEC and then look through them with a Browser. For creating simple Web pages you should have a basic knowledge of HTML or use an HTML editor. The file name of your own homepage must be **main.htm**, if the page is to be accessed automatically in the Browser when the IP address of the CPX-FEC is entered.

Detailed information on creating HTML pages can be found e.g. in Internet under http://selfhtml.org

**CI commands as HTTP query**

HTTP query CI commands can be added to the HTML page names as an HTTP query. The HTML page named will then be accessed and at the same time the CI command will be carried out.

**Example**

```
http://10.8.64.251/main.htm?ci:maw0=128
```

Accesses the page main.htm and at the same time sends the CI command maw0=128 to the command interpreter (maw0 stands for modify output: word 0).

**http_in_ci** An internal page, which shows only the result of a CI command, can also be accessed with an HTTP query.

**Example**

```
http://10.8.64.251/http_in_ci?ci:daw0
```
4. With the CPX terminal to the Ethernet

An HTML page is shown which shows the contents of output word 0 (daw0 stands for display outputword 0).

Incorporating CI commands in HTML

FSTCI tag
The Webserver driver supports a special HTML tag. This consists of the abbreviation fstci and the desired CI command.

Example
Display IW0: <fstci dew0>

When the HTML page is accessed, the text “Display IW0” and the contents of input word 0 will be shown (dew0 stands for display inputword 0).

Link tag
By means of a Link tag, you can send CI commands by clicking a link on the command interpreter.

Example
<A href="main.htm?ci:maw0=255"> output word 0 = 255</A>

By clicking the text “Output word 0 = 255” the CI command “maw0=255” will be sent (maw0 stands for modify outputword 0).

Form tag
With the Form Tag you can group several CI commands in a form. You can transfer the CI commands by clicking the Send button.
Example of Form tag

```html
<form method="POST" action="">
  Load to flag word 1:
  <input type="text" name="MFW1" value="<FSTCI dmw1" size="6" maxlenght="6"></input>
  Load to flag word 2: <select name="mmw3">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
  </select>
  Load 1 to flag word 3
  <INPUT type="radio" name="MMW2" value="1" checked>
  Load 2 to flag word 3
  <INPUT type="radio" name="MMW2" value="2">
  Load 3 to flag word 3
  <INPUT type="radio" name="MMW2" value="3">
  <input type="submit" name="send" value="Send">
  <input type="reset" value="Lösch" name="zurückstellen">
</form>
```

(result see Fig./hardspace4/7)

Fig. 4/7: Example of Form tag
4. With the CPX terminal to the Ethernet

4.4.4 Load files into the Webserver

Please note
Due to the transmission protocol TFTP, you can only transfer the files of the Web pages with the FST. FTP programs usually used by Webmasters are not suitable.

In order to transfer files (e.g. HTML pages) for the Webserver into the controller with FST, proceed as follows:

1. Open the window File transfer with [Online] [File Transfer].

2. Select drive “B:/” in the upper part of the list in the window “File Transfer”.
   Switch to the sub-directory “web”.

3. Now click on the blue arrow which points downwards, in order to load a file into the controller.

4. Select the desired file in the subsequent dialogue and confirm your selection with “Open”. The file will then be transferred to the controller.

Fig. 4/8: Loading files into the Webserver with FST
4. With the CPX terminal to the Ethernet

Unused memory space

You can see the amount of unused memory space in the status line of the window. The sum of the Webserver memory and the PLC memory is displayed.

Now you can access your HTML pages with a Browser.

Accessing the HTML pages

You can access the HTML pages in the controller as follows:

1. Open your Browser.

2. If you are using a direct connection without the network, make sure that use of the Proxy Server is switched off in the Browser settings.

3. Enter the IP address of the controller in the box “Address” and confirm the entry with Enter.

If you have already loaded a self-created HTML page with the name Main.htm into the controller, this will be displayed. If no main.htm exists, the standard Homepage of the controller will be displayed (see Fig. 4/6).
4. With the CPX terminal to the Ethernet
Remote Controller mode

Chapter 5
5. Remote Controller mode

Contents

5. Remote Controller mode .................................................. 5-1

5.1 Remote Controller Ethernet ........................................... 5-4
  5.1.1 Configuration ...................................................... 5-4
  5.1.2 Operation as Remote Controller ............................... 5-5
  5.1.3 Diagnosis with the controller LEDs RUN, STOP, ERROR and TP ... 5-11

5.2 Remote Controller Field bus ......................................... 5-13
  5.2.1 Configuration ...................................................... 5-15
  5.2.2 Configuration example and communication sequence ........ 5-16
  5.2.3 Parametrizing ..................................................... 5-21
  5.2.4 Diagnosis .......................................................... 5-21
5. Remote Controller mode

Contents of this chapter

General commissioning of the CPX-FEC is described in chapter 3. This chapter contains additional information on the operating mode Remote Controller. There is a communication connection between the CPX-FEC and other slaves with one of the following possibilities:

- **Remote Controller Ethernet:**
  The CPX terminal is controlled by the CPX-FEC as in the operating mode Stand Alone. The FEC communicates with a higher-order controller via Ethernet. It can also be configured via the Ethernet interface.
  With the information technology components introduced in chapter 4 you can access the CPX-FEC for control, diagnosis and communication (e.g. via Webserver, e-mail, ...).

  Modbus/TCP and EasyIP are available as protocols in the user layer. Modbus is an open communication protocol based on the master-slave architecture. It is an established standard for communication via Ethernet-TCP/IP in automation technology. EasyIP is a protocol for fast exchange of operands between Festo controllers.

- **Remote Controller Field bus:**
  The CPX terminal is controlled by the CPX-FEC as in the operating mode Stand Alone. The FEC communicates with a higher-order controller via the field bus. For this operating mode a field bus node must be installed in the CPX terminal. The field bus node must be set to the operating mode “Remote Controller” with its DIL switches.
  The advantages: You can incorporate the PLC of the CPX-FEC as a preprocessor in your existing field bus system. If the system is to be converted later to Ethernet, you can use the CPX-FEC further for this.
5. Remote Controller mode

5.1 Remote Controller Ethernet

The CPX-FEC controls the CPX terminal in the operating mode “Remote Controller Ethernet”. The following possibilities are then available via Ethernet:

- configuration, parametrizing and programming via the software package FST
- communication with a higher-order controller via Modbus/TCP or EasyIP
- access to the CPX terminal via Webserver.

Fig. 5/1: Example of a CPX terminal with CPX-FEC as Remote Controller on the Ethernet
5. Remote Controller mode

5.1.1 Configuration

The configuration and commissioning in Remote Controller mode corresponds to a large extent to general commissioning as described in section 3.1.

Create connection to the CPX-FEC

1. Set the rotary switch of the CPX-FEC to “0”.
2. Connect your PC and the CPX-FEC either
   - via the programming interface with an RS232 cable (see section 2.4) or
   - via the Ethernet interface with a patch cable (connection via hub/switch) or with a crossover cable (direct connection) (see section 2.3).

The connection via the programming interface serves mainly for configuration. Only when the Ethernet interface is used, the CPX-FEC appears in the network and the Web-server becomes useful.

3. Access the window “FST Preferences” in FST 4.1 with [Extras][Preferences...]. Select your type of connection in the register “Communication”:
   - Programming interface: “Use RS232”
     Set the local COM interface of your PC and the baud rate.
   - Ethernet interface “Use TCP/IP”
     Set the IP address of your CPX-FEC or click on “Search” in order to display a list of the online available controllers. Select your controller with a double click.
     Information on the IP addressing can be found in section 4.3.2.
4. Confirm with OK.
5. Remote Controller mode

Configuring

5. Create a new project (see section 3.1.2).

6. Configure the modules of your CPX terminal as described in section 3.1.2.
5. Remote Controller mode

5.1.2 Operation as Remote Controller

For operation as Remote Controller the specifications on addressing, parametrizing and diagnosis, which you can find in sections 3.2 to 3.6, apply here.

Ethernet

Operation as Remote Controller corresponds to the Stand Alone mode, however the CPX terminal can also communicate with a higher-order controller via the Ethernet interface.

Communication in the user layer (see section 4.2) takes place here with the protocols Modbus/TCP or EasyIP.

Modbus/TCP

The driver for Modbus/TCP is loaded automatically with the creation of a new project.

Driver options

- Set in the driver options of the Modbus/TCP driver the start flag word for data exchange of your CPX terminal.

![Driver Options - MODBUS/TCP](image)

Fig. 5/2: Setting the first flag word for the Modbus/TCP communication (example)

The following tables show the Modbus commands and address assignments required for communication.
### Modbus commands and address assignment

The following table shows the correlation between the Modbus address and the processing data. The processing data of the inputs are assigned to group G and that of the outputs to group H.

<table>
<thead>
<tr>
<th>Modbus command</th>
<th>Function code</th>
<th>Modbus address</th>
<th>Meaning</th>
<th>Remote I/O 16-bit access</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>read 4 x registers</td>
<td>3</td>
<td>45392...45647</td>
<td>Processing data for inputs</td>
<td>read</td>
<td>G</td>
</tr>
<tr>
<td>write 4 x registers</td>
<td>6, 16</td>
<td>40001...40256</td>
<td>Processing data for outputs</td>
<td>write</td>
<td>H</td>
</tr>
</tbody>
</table>
| read/write 4 x registers | 23         | 45392...45647 40001...40256 | Processing data for inputs  
Processing data for outputs | read   
write | G  
H |
| Read device identification | 43            | Objects              | Objects ID0, 1, 2, 3, 4, 5           | read                      | F     |

Tab. 5/1: Overview of the Modbus function codes for the CPX-FEC in the operating mode Remote Controller Ethernet
5. Remote Controller mode

Data exchange takes place via flag words (FW). The processing image for the input and output data is composed of a data field of up to 256 flag words as follows:

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Processing data for inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bit</td>
</tr>
<tr>
<td>45392</td>
<td>FWx</td>
</tr>
<tr>
<td>45393</td>
<td>FWx+1</td>
</tr>
<tr>
<td>45394</td>
<td>FWx+2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>45647</td>
<td>FWx+255</td>
</tr>
</tbody>
</table>

Tab. 5/2: Input processing data of group G (FW = flag word)

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Processing data for outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bit</td>
</tr>
<tr>
<td>40001</td>
<td>FWx</td>
</tr>
<tr>
<td>40002</td>
<td>FWx+1</td>
</tr>
<tr>
<td>40003</td>
<td>FWx+2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>40256</td>
<td>FWx+255</td>
</tr>
</tbody>
</table>

Tab. 5/3: Output processing data of group H (FW = flag word)
5. Remote Controller mode

EasyIP

This protocol is used for the fast exchange of operands between Festo controllers (e.g. CPX-FEC, FEC Standard, PS1, etc.).

In the operating mode Remote Controller Ethernet the CPX-FEC behaves like an EasyIP-Server as well as like an EasyIP-Client.

As server the following EasyIP operand types are supported:

<table>
<thead>
<tr>
<th>Operand</th>
<th>Operand number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1, flag word</td>
<td>0 ... 9999</td>
</tr>
<tr>
<td>Type 2, input word</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 3, output word</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 4, register</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 5, timer preselect</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 11, strings 1)</td>
<td>0 ... max. 1023 2)</td>
</tr>
</tbody>
</table>

1) For this the string driver must be loaded.
2) Depending on the number of configured strings in the driver.

Tab. 5/4: EasyIP operand types supported by the CPX-FEC

You must program the functioning as EasyIP-Client with function modules via FST (see description of the TCP/IP driver in the FST manual, volume “Drivers and modules”).

5-10
5. Remote Controller mode

5.1.3 Diagnosis with the controller LEDs RUN, STOP, ERROR and TP

Information on the general CPX LEDs PS, PL, SF and M can be found in section 1.4.1.

In the operating mode Remote Controller the controller and Ethernet LEDs have the following meanings:

<table>
<thead>
<tr>
<th>RUN</th>
<th>LED (green)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>PLC program started</td>
</tr>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>PLC program stopped</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STOP</th>
<th>LED (yellow)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>PLC program started</td>
</tr>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>PLC program stopped</td>
</tr>
</tbody>
</table>

Tab. 5/5: Status displays of the LEDs RUN and STOP in the operating mode Stand Alone
5. Remote Controller mode

<table>
<thead>
<tr>
<th>ERROR</th>
<th>LED (red)</th>
<th>Sequence</th>
<th>Status</th>
<th>Fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>No fault</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>PLC program fault</td>
<td>Read out fault code with FST or handheld (see section 3.5.2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TP (Link/Traffic)</th>
<th>LED (green)</th>
<th>Sequence</th>
<th>Status</th>
<th>Fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>Ethernet connection OK</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>LED flashes</td>
<td>ON OFF</td>
<td>Data traffic (Traffic)</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>No Ethernet connection or</td>
<td>If necessary, check Ethernet connection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ethernet cable not connected</td>
<td></td>
</tr>
</tbody>
</table>

Tab. 5/6: Status displays of the LEDs ERROR and TP
5. Remote Controller mode

5.2 Remote Controller Field bus

The CPX-FEC controls the CPX terminal in the operating mode “Remote Controller Field bus”. The CPX terminal thereby communicates with a higher-order controller via the field bus node. Communication between the CPX-FEC and the field bus node is carried out with 8 I-bytes and 8 O-bytes. The Ethernet interface of the CPX-FEC can be used for configuration and programming.

![Diagram of CPX terminal with field bus node and CPX-FEC](image)

**Fig. 5/3:** Example of a CPX terminal with field bus node and CPX-FEC as Remote Controller on a field bus
5. Remote Controller mode

Field bus node (Remote Controller)

The following table shows the field bus nodes which can be configured for the operating mode Remote Controller Field bus. Data exchange takes place on all field bus nodes via 8 bytes of inputs and 8 bytes of outputs.

<table>
<thead>
<tr>
<th>Module</th>
<th>Module identifiers 1)</th>
<th>Assigned address range</th>
<th>Remarks, Version 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB06 Interbus Remote Controller</td>
<td>FB06-RC</td>
<td>4 words / 8 bytes I</td>
<td>as from software status 11.07.2003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 words / 8 bytes O</td>
<td></td>
</tr>
<tr>
<td>FB11 DeviceNet Remote Controller</td>
<td>FB11-RC</td>
<td>4 words / 8 bytes I</td>
<td>as from software status 06.10.2003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 words / 8 bytes O</td>
<td></td>
</tr>
<tr>
<td>FB13 Profibus Remote Controller</td>
<td>FB13-RC</td>
<td>4 words / 8 bytes I</td>
<td>as from software status 26.02.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 words / 8 bytes O</td>
<td></td>
</tr>
<tr>
<td>FB14 CANopen multi-I/O module</td>
<td>FB14-RC</td>
<td>4 words / 8 bytes I</td>
<td>as from software status 22.08.2003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 words / 8 bytes O</td>
<td></td>
</tr>
<tr>
<td>FB23 CC-Link Remote Controller</td>
<td>FB23-RC</td>
<td>4 words / 8 bytes I</td>
<td>as from software status 07.08.2003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 words / 8 bytes O</td>
<td></td>
</tr>
</tbody>
</table>

Status: April 2004
1) Module identifiers on the handheld and in the Hardware Configurator of the FST 4.1
2) Software status (SW) see type plate

Tab. 5/7: Overview of function modules of the CPX terminal with FEC for the operating mode Remote Controller Field bus
5. Remote Controller mode

5.2.1 Configuration

The most important commissioning steps

1. Use the DIL switches in the field bus node and in the CPX-FEC to set the operating mode Remote Controller or check the setting: See the manual for the relevant field bus node or in chapter 2.

2. If necessary, switch off the power supply to the CPX terminal and then switch on again. The modified operating mode is not recognized until after Power OFF/ON.

3. Connect your PC and the CPX-FEC either
   - via the programming interface with an RS232 cable (see section 2.4) or
   - via the Ethernet interface with a patch cable (connection via hub/switch) or with a crossover cable (direct connection) (see section 2.3).

4. Configure the CPX terminal with the FST software package: see chapter 3.

5. Connect the field bus cable to the field bus node of the CPX terminal: See the manual for the field bus node.

6. Configure your system with the appropriate control software of your field bus node: see manuals for the field bus node and for your controller. The field bus node appears as an I/O module with 8 bytes of inputs and 8 bytes of outputs. Data exchange is carried out via these inputs and outputs.
5. Remote Controller mode

5.2.2 Configuration example and communication sequence

The following diagram shows as an example a configuration in FST with a field bus node 13.

Field bus node in the configuration table (example: CPX-FB13 PROFIBUS)

Fig. 5/4: Configuration of the CPX terminal in the operating mode Remote Controller Field bus

The default addressing applies for addressing the field bus node (see section 3.2).
5. Remote Controller mode

<table>
<thead>
<tr>
<th>Location</th>
<th>Module</th>
<th>I address</th>
<th>O address</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CPX-FB13</td>
<td>128</td>
<td>128</td>
<td>The inputs of the field bus node are the outputs of the CPX-FEC. The outputs of the field bus node are the inputs of the CPX-FEC.</td>
</tr>
<tr>
<td>1</td>
<td>CPX-FEC</td>
<td>132</td>
<td>132</td>
<td>Inputs: Rotary switch position Outputs: unused</td>
</tr>
<tr>
<td>2</td>
<td>Digital 8-input module (8DI)</td>
<td>0</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Tab. 5/8: Configuration example with CPX field bus node 13 and CPX-FEC
5. Remote Controller mode

Communication sequence

8 bytes of inputs and 8 bytes of outputs are available for data exchange between the CPX-FEC and the field bus node.

Fig. 5/5: Example of the communication sequence of a CPX terminal with field bus node and CPX-FEC as Remote Controller
5. Remote Controller mode

The following applies:

- 8 bytes of outputs of the CPX-FEC are mapped onto 8 bytes of inputs of the field bus node.
- 8 bytes of outputs of the field bus node are mapped onto 8 bytes of inputs of the CPX-FEC.

**Example of application**

Let us assume that the CPX-FEC is to process a program in which a cylinder is to retract and extend several times. The number of positioning cycles depends on the processing and should be transmitted by the higher-order controller (see Tab. 5/9).

1. The higher-order controller writes the number of positioning cycles into the output byte 0 of the CPX field bus node.

2. The value appears also in input word 128 of the CPX-FEC. The PLC of the CPX-FEC uses this information and carries out the specified number of positioning cycles. The higher-order controller is therefore relieved.

3. If the CPX-FEC has carried out its task, it sends a message to confirm this in output word 128.

4. The value appears also in input byte 0 of the field bus node. The higher-order controller evaluates this information and can now introduce the next work step.
5. Remote Controller mode

<table>
<thead>
<tr>
<th>CPX-FEC CPX field bus node</th>
<th>Remote I/O (8 bytes I, 8 bytes O)</th>
</tr>
</thead>
</table>

### Outputs of the CPX-FEC

<table>
<thead>
<tr>
<th>Bit</th>
<th>15</th>
<th>12</th>
<th>11</th>
<th>8</th>
<th>7</th>
<th>4</th>
<th>3</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OW 128</td>
<td>Byte 1</td>
<td>Byte 0</td>
<td>Byte 1</td>
<td>Byte 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OW 129</td>
<td>Byte 3</td>
<td>Byte 2</td>
<td>Byte 3</td>
<td>Byte 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OW 130</td>
<td>Byte 5</td>
<td>Byte 4</td>
<td>Byte 5</td>
<td>Byte 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OW 131</td>
<td>Byte 7</td>
<td>Byte 6</td>
<td>Byte 7</td>
<td>Byte 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Inputs of the CPX-FEC

<table>
<thead>
<tr>
<th>Bit</th>
<th>15</th>
<th>12</th>
<th>11</th>
<th>8</th>
<th>7</th>
<th>4</th>
<th>3</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW 128</td>
<td>Byte 1</td>
<td>Byte 0</td>
<td>Byte 1</td>
<td>Byte 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IW 129</td>
<td>Byte 3</td>
<td>Byte 2</td>
<td>Byte 3</td>
<td>Byte 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IW 130</td>
<td>Byte 5</td>
<td>Byte 4</td>
<td>Byte 5</td>
<td>Byte 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IW 131</td>
<td>Byte 7</td>
<td>Byte 6</td>
<td>Byte 7</td>
<td>Byte 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. 5/9: Communication between the CPX-FEC and the field bus node via I/O bytes (grey markings for the application example above)
5. Remote Controller mode

5.2.3 Parametrizing

The information in section 3.3 applies for parametrizing. It is not possible to parametrize via the field bus.

5.2.4 Diagnosis

Diagnosis with the control LEDs RUN, STOP, ERROR

Information on the general CPX LEDs PS, PL, SF and M can be found in section 1.4.1.

In the operating mode Remote Controller the control and Ethernet LEDs have the following meanings:

<table>
<thead>
<tr>
<th>RUN LED (green)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>OFF</td>
<td>PLC program will be started</td>
</tr>
<tr>
<td>LED lights up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED is out</td>
<td>ON</td>
<td>PLC program will be stopped</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LED</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>OFF</td>
<td>PLC program will be started</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>PLC program will be stopped</td>
</tr>
</tbody>
</table>
5. Remote Controller mode

<table>
<thead>
<tr>
<th>STOP</th>
<th>LED (yellow)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>PLC program started</td>
</tr>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>PLC program stopped</td>
</tr>
</tbody>
</table>

Tab. 5/10: Status displays of the LEDs RUN and STOP in the operating mode Remote Controller Field bus

<table>
<thead>
<tr>
<th>ERROR</th>
<th>LED (red)</th>
<th>Sequence</th>
<th>Status</th>
<th>Fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>No fault</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>PLC program fault I/O fault</td>
<td>Read out fault code with FST or handheld (see section 5.2)</td>
</tr>
</tbody>
</table>

Tab. 5/11: Status displays of the ERROR LEDs

**Diagnosis via the field bus**

The device-specific diagnostic functions of the relevant field bus cannot be used.

As communication with the field bus in the operating mode Remote Controller Field bus takes place via 8 I/O bytes, these bytes must also be used, if necessary, for diagnostic purposes.
Remote I/O Ethernet

Chapter 6
6. Remote I/O Ethernet

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>Remote I/O Ethernet</td>
<td>6-1</td>
</tr>
<tr>
<td>6.1</td>
<td>General information</td>
<td>6-4</td>
</tr>
<tr>
<td>6.2</td>
<td>Modbus/TCP: Commands and addressing</td>
<td>6-7</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Modbus commands and address assignment</td>
<td>6-7</td>
</tr>
<tr>
<td>6.2.2</td>
<td>CPX status information (group A)</td>
<td>6-8</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Processing data (groups B and D)</td>
<td>6-10</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Diagnostic memory (groups C and E)</td>
<td>6-24</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Objects (group F)</td>
<td>6-26</td>
</tr>
<tr>
<td>6.2.6</td>
<td>Addressing examples for Modbus/TCP</td>
<td>6-26</td>
</tr>
<tr>
<td>6.3</td>
<td>EasyIP</td>
<td>6-32</td>
</tr>
<tr>
<td>6.4</td>
<td>Diagnosis</td>
<td>6-33</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Diagnosis with the control and Ethernet LEDs</td>
<td>6-33</td>
</tr>
</tbody>
</table>
6. Remote I/O Ethernet

Contents of this chapter
This chapter describes the functions of the CPX terminal in the operating mode Remote I/O.
The CPX-FEC behaves here like a field bus slave on the Ethernet. It can communicate via the protocols Modbus/TCP or EasyIP.

Modbus is an open communication protocol based on the master-slave architecture. Modbus/TCP is an established standard for communication via Ethernet-TCP/IP in automation technology.

EasyIP is a simple protocol for communication between Festo controllers.

Further information
Configuration takes place via your controller. Information on this can be found in the documentation for your controller. The software package Schneider Unity is used in this chapter as an example.
6. Remote I/O Ethernet

6.1 General information

The CPX-FEC behaves in the operating mode Remote I/O like a field bus slave on the Ethernet: The CPX terminal is controlled by an external controller via Modbus/TCP. With the Webserver of the CPX-FEC the CPX terminal can be monitored and controlled by a Web-Browser. The PLC of the CPX-FEC is not used. You can use the programming interface if required for setting the IP address.

Fig. 6/1: CPX terminal as field bus slave on the Ethernet

Please note
Make sure that the DIL switches are set correctly for the operating mode Remote I/O Ethernet (see chapter 2 Installation).
6. Remote I/O Ethernet

The most important commissioning steps

1. Set the CPX-FEC with the DIL switches to the operating mode Remote I/O Ethernet or check the setting: see section 2.2.2

2. If required, switch off the power supply to the CPX terminal and then switch on again. The modified operating mode is not recognized until after Power OFF/ON.

3. IP addressing (see section 4.3.2):
   - address the CPX-FEC with the handheld CPX-MMI or
   - address it via the controller (manual or dynamic address specification).

4. Connect the CPX terminal with the CPX-FEC to the Ethernet (see chapter 2).

5. Configure your system, as usual, with the appropriate control software.

The diagram below shows as an example what a configuration of the CPX terminal in the software package Schneider Unity looks like.
6. Remote I/O Ethernet

1. IP address of a CPX-FEC
2. Modbus start address (inputs)
3. Modbus start address (outputs)

Fig. 6/2: The Modbus start addresses of the CPX terminal for inputs and outputs are entered in the configuration software (example: Schneider Unity)
6. Remote I/O Ethernet

6.2 Modbus/TCP: Commands and addressing

In order to configure the CPX terminal for Modbus/TCP, you will require the Modbus addresses of the data and of the I/Os of the CPX terminal. Addressing examples can be found in section 6.2.6.

6.2.1 Modbus commands and address assignment

The table below shows the correlation between the Modbus address and data or parameters of the CPX terminal. The data are assigned to various groups.

<table>
<thead>
<tr>
<th>Modbus command</th>
<th>Funct. code</th>
<th>Modbus address</th>
<th>Meaning</th>
<th>Remote I/O 16-bit access</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>read 4 x registers</td>
<td>3</td>
<td>45357...45391, 45392...45647, 45648...45655, 45656...46055</td>
<td>CPX status information, Processing data inputs, Diagnostic memory parameters, Diagnostic memory data</td>
<td>read, read, read, read</td>
<td>A, B, C, C</td>
</tr>
<tr>
<td>write 4 x registers</td>
<td>6, 16</td>
<td>40001...40256, 40257...40264</td>
<td>Processing data outputs, Diagnostic memory parameters</td>
<td>write, write</td>
<td>D, E</td>
</tr>
<tr>
<td>read/write 4 x registers</td>
<td>23</td>
<td>45357...45391, 45392...45647, 45648...45655, 45656...46055, 40001...40256, 40257...40264</td>
<td>CPX status information, Processing data inputs, Diagnostic memory parameters, Diagnostic memory data, Processing data outputs, Diagnostic memory parameters</td>
<td>read, read, read, write, write</td>
<td>A, B, C, C, D, E</td>
</tr>
<tr>
<td>read device identification</td>
<td>43</td>
<td>Objects</td>
<td>Objects ID 0, 1, 2, 3, 4, 5</td>
<td>read</td>
<td>F</td>
</tr>
</tbody>
</table>

Tab. 6/1: Overview of the Modbus function codes for the CPX-FEC in the operating mode Remote I/O
6. Remote I/O Ethernet

6.2.2 CPX status information (group A)

The status information provides information on the configuration and the fault status of the CPX terminal. It lies in the Modbus address range 45367 to 45391.

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>CPX terminal configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>45367</td>
<td>Module 0 ... 15</td>
</tr>
<tr>
<td>45368</td>
<td>Module 16 ... 31</td>
</tr>
<tr>
<td>45369</td>
<td>Module 32 ... 47</td>
</tr>
</tbody>
</table>
| Bit n = 0: Module does not exist  
Bit n = 1: Module exists |

Tab. 6/2: Configuration information on the modules of the CPX terminal

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Fault recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>45383</td>
<td>Module 0 ... 15</td>
</tr>
<tr>
<td>45384</td>
<td>Module 16 ... 31</td>
</tr>
<tr>
<td>45385</td>
<td>Module 32 ... 47</td>
</tr>
</tbody>
</table>
| Bit n = 0: No fault  
Bit n = 1: Fault in module n |

Tab. 6/3: Recognition, which module registers a fault
## 6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Status register</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit 15</td>
<td>14</td>
</tr>
<tr>
<td>45391</td>
<td>Bit 4 = 1: Handheld connected; 0: Not connected</td>
</tr>
<tr>
<td></td>
<td>Bit 11 = 1: Parameter write-protected; 0: No write protection</td>
</tr>
<tr>
<td></td>
<td>Bit 15 = 1: Force active; 0: Force inactive</td>
</tr>
</tbody>
</table>

Tab. 6/4: Further status information (extract from system data)
6. Remote I/O Ethernet

6.2.3 Processing data (groups B and D)

The processing image of the input data (group B) and of the output data (group D) is composed without gaps of the data of the modules as they are fitted from left to right in the CPX terminal. The Modbus addresses depend accordingly on the modules fitted on your CPX terminal. They are arranged in ascending order without gaps.

Module CPX-FEC

* = The Modbus addresses are assigned in ascending order without gaps and depend on the modules on the CPX terminal (addressing example in section 6.2.6).

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>CPX-FEC Remote I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15</td>
</tr>
<tr>
<td>*</td>
<td>Result of access to the I/O diagnostic interface 1)</td>
</tr>
<tr>
<td>*</td>
<td>Data from the system table (read access)</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data (see Tab. 6/24)</td>
</tr>
</tbody>
</table>

1) Composition of the I/O diagnostic interface [Tab. 6/25, Tab. 6/26]

Tab. 6/5: CPX-FEC

Festo PBE-CPX-FEC-EN en 0404NH
6. Remote I/O Ethernet

Electric modules

* = The Modbus addresses are assigned in ascending order without gaps and depend on the modules on the CPX terminal (addressing example in section 6.2.6).

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Digital 4-input module (4DI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15  12  11  8  7  4  3  0</td>
</tr>
<tr>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/6: 4DI module

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Digital 8-input module (4DI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15  12  11  8  7  4  3  0</td>
</tr>
<tr>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/7: 8DI module
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Digital 4-output module (4DO)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Echo outputs</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/8: 4-DO module

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Digital 8-input/output module (8DI/8DO)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>*</td>
<td>Inputs</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/9: 8DI/8DO module
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Analogue 2-input module (2AI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>Analogue inputs channel 0</td>
</tr>
<tr>
<td>*</td>
<td>Analogue inputs channel 1</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/10: 2AI module

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Analogue 2-output module (2AO)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>Echo analogue outputs channel 0</td>
</tr>
<tr>
<td>*</td>
<td>Echo analogue outputs channel 1</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/11: 2AO module
6. Remote I/O Ethernet

Pneumatic modules

* = The Modbus addresses are assigned in ascending order without gaps and depend on the modules on the CPX terminal (addressing example in section 6.2.6).

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic MPA1 module type 32 (1 ... 8 valves)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15  12  11  8  7  4  3  0</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic MPA2 module type 32 (1 ... 4 valves)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15  12  11  8  7  4  3  0</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/12: MPA modules

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for CPA pneumatic type 12 set to 1 ... 8 valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15  12  11  8  7  4  3  0</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/13: CPA pneumatic interface
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for CPA pneumatic type 12 set to 1 ... 16 valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>Echo outputs</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/14: CPA pneumatic interface

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for CPA pneumatic type 12 set to 1 ... 22 valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>Echo outputs 0 ... 15</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/15: CPA pneumatic interface

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for pneumatic type Midi/Maxi (type 03) set to 1 ... 8 valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/16: Midi/Maxi pneumatic interface
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for pneumatic type Midi/Maxi (type 03) set to 1 ... 16 valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>Echo outputs</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/17: Midi/Maxi pneumatic interface

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for pneumatic type Midi/Maxi (type 03) set to 1 ... 24 valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8 7 4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>Echo outputs 0 ... 15</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/18: Midi/Maxi pneumatic interface
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Pneumatic interface for pneumatic type Midi/Maxi (type 03) set to 1 ... 32*) valves</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15</td>
</tr>
<tr>
<td>*</td>
<td>Echo outputs 0 ... 15</td>
</tr>
<tr>
<td>*</td>
<td>x</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>*</td>
<td>*) Only 26 usable</td>
</tr>
</tbody>
</table>

Tab. 6/19: Midi/Maxi pneumatic interface

* = The Modbus addresses are assigned in ascending order without gaps and depend on the modules on the CPX terminal (addressing example in section 6.2.6).
Technology module CP interface

The CP interface has connections for 4 strings to each of which maximum 4 CP modules can be connected.

The last used string is decisive for the number of assigned bytes, even if numerically lower strings are not physically assigned.

* = The Modbus addresses are assigned in ascending order without gaps and depend on the modules on the CPX terminal (addressing example in section 6.2.6).

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>CPX-CP interface used string: 1 (line 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 1</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 3</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 1</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 3</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/20: CPX-CP interface
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>CPX-CP interface used strings: 1, 2 (line 1 ... 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 1</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 3</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 5</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 7</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 1</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 3</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 5</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 7</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/21: CPX-CP interface
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>CPX-CP interface used strings: 1, 2, 3 (line 1 ... 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15 12 11 8  7  4 3 0</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 1 I-data byte 0 O-data byte 1 O-data byte 0</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 3 I-data byte 2 O-data byte 3 O-data byte 2</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 5 I-data byte 4 O-data byte 5 O-data byte 4</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 7 I-data byte 6 O-data byte 7 O-data byte 6</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 9 I-data byte 8 O-data byte 9 O-data byte 8</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 11 I-data byte 10 O-data byte 11 O-data byte 10</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 1 Echo O-data 0 –</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 3 Echo O-data 2 –</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 5 Echo O-data 4 –</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 7 Echo O-data 6 –</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 9 Echo O-data 8 –</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 11 Echo O-data 10 –</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data –</td>
</tr>
</tbody>
</table>

Tab. 6/22: CPX-CP interface
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>CPX-CP interface used strings 1, 2, 3, 4 (line 1 ... 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Processing data inputs</td>
</tr>
<tr>
<td>Bit</td>
<td>15</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 1</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 3</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 5</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 7</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 9</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 11</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 13</td>
</tr>
<tr>
<td>*</td>
<td>I-data byte 15</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 1</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 3</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 5</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 7</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 9</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 11</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 13</td>
</tr>
<tr>
<td>*</td>
<td>Echo O-data 15</td>
</tr>
<tr>
<td>*</td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/23: CPX-CP interface
6. Remote I/O Ethernet

**Composition of diagnostic data (diagnostic word)**

<table>
<thead>
<tr>
<th>Module diagnostic data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input data</td>
</tr>
<tr>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>0/1 0/1 Channel number 0 ... 63) Fault number (0 ... 255)</td>
</tr>
</tbody>
</table>

Bit 15 and 14:
- 0 0: Number of the first faulty O-channel
- 0 1: Number of the first faulty I-channel
- 1 0: There is a module fault
- 1 1: Reserved

Tab. 6/24: Module diagnostic data
6. Remote I/O Ethernet

Composition of the data of the I/O diagnostic interface

<table>
<thead>
<tr>
<th>I/O diagnostic interface</th>
<th>Write access</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
<td></td>
</tr>
<tr>
<td>x  x  x</td>
<td>CPX function number</td>
</tr>
</tbody>
</table>

Data from the system table

Bit 15 (control bit): with positive edge (0 → 1) write access is made to I/O diagnostic interface
Bit 14 = 0: Byte value, 1: Word value
Bit 13 = 0: Read request, 1: Write request

Tab. 6/25: I/O diagnostic interface

<table>
<thead>
<tr>
<th>I/O diagnostic interface</th>
<th>Read access</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Result of last request</td>
</tr>
<tr>
<td></td>
<td>Data from the system table</td>
</tr>
</tbody>
</table>

Result of last request:
- 0: Wait
- 8000h: Request successful
- > 8000h: Fault
  - 8001h: Write protection or handheld has write access
  - 8002h: Writing not permitted, reserved range
  - 8003h: Internal fault

Tab. 6/26: I/O diagnostic interface

Information on the I/O diagnostic interface and examples of its use can be found in the CPX system manual in the chapter “Diagnosis and fault treatment”.

Festo RBE-CPX-FEC-EN en 0404NH 6-23
6. Remote I/O Ethernet

6.2.4 Diagnostic memory (groups C and E)

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Diagnostic memory parameters and data Read access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input data</td>
</tr>
<tr>
<td></td>
<td>Bit</td>
</tr>
<tr>
<td>45648</td>
<td>Run/Stop filter 1 (CPX function number 3480)</td>
</tr>
<tr>
<td>45652</td>
<td>Run/Stop filter 2 (CPX function number 3484)</td>
</tr>
<tr>
<td>45652</td>
<td>Fault end filter (CPX function number 3484)</td>
</tr>
<tr>
<td>45652</td>
<td>Fault number filter (CPX function number 3484)</td>
</tr>
<tr>
<td>45652</td>
<td>Module/Channel filter (CPX function number 3484)</td>
</tr>
<tr>
<td>45653</td>
<td>Module number MN (CPX function number 3485)</td>
</tr>
<tr>
<td>45654</td>
<td>Channel number CN (CPX function number 3486)</td>
</tr>
<tr>
<td>45655</td>
<td>Fault number FN (CPX function number 3487)</td>
</tr>
<tr>
<td>45650</td>
<td>Diagnostic memory data – number of entries (CPX function number 3482)</td>
</tr>
<tr>
<td>45651</td>
<td>Diagnostic memory data – overrun (CPX function number 3483)</td>
</tr>
<tr>
<td>45651</td>
<td>Diagnostic memory data – status (CPX function number 3483)</td>
</tr>
<tr>
<td>45656</td>
<td>Diagnostic memory data (CPX function number 3488 + n)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

1) See CPX system manual

Tab. 6/27: Read access to diagnostic memory parameters and data
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Modbus address</th>
<th>Diagnostic memory parameters</th>
<th>Write access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Output data – write and modify</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>40257</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40261</td>
<td>Run/Stop filter 1 (CPX function number 3480)</td>
<td></td>
</tr>
<tr>
<td>40261</td>
<td>Run/Stop filter 2 (CPX function number 3484)</td>
<td></td>
</tr>
<tr>
<td>40261</td>
<td>Fault end filter (CPX function number 3484)</td>
<td></td>
</tr>
<tr>
<td>40261</td>
<td>Fault number filter (CPX function number 3484)</td>
<td></td>
</tr>
<tr>
<td>40261</td>
<td>Module/Channel filter (CPX function number 3484)</td>
<td></td>
</tr>
<tr>
<td>40262</td>
<td>Module number MN (CPX function number 3485)</td>
<td></td>
</tr>
<tr>
<td>40263</td>
<td>Channel number CN (CPX function number 3486)</td>
<td></td>
</tr>
<tr>
<td>40264</td>
<td>Fault number FN (CPX function number 3487)</td>
<td></td>
</tr>
</tbody>
</table>

Tab. 6/28: Write access to diagnostic memory parameters

Information on the diagnostic memory can be found in the CPX system manual in the appendix “Parameters and data of the CPX terminal”.

---

Festo RBE-CPX-FEC-EN en 0404NH 6-25
6. Remote I/O Ethernet

6.2.5 Objects (group F)

<table>
<thead>
<tr>
<th>Object ID</th>
<th>Object name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Manufacturer name</td>
<td>“Festo AG &amp; Co. KG”</td>
</tr>
<tr>
<td>1</td>
<td>Product code</td>
<td>“CPX-FEC”</td>
</tr>
<tr>
<td>2</td>
<td>MajorMinorRevision</td>
<td>“x.y” 1)</td>
</tr>
<tr>
<td>3</td>
<td>VendorURL</td>
<td>“<a href="http://www.festo.com%E2%80%9D">http://www.festo.com”</a></td>
</tr>
<tr>
<td>4</td>
<td>Product name</td>
<td>“Modbus TCP”</td>
</tr>
<tr>
<td>5</td>
<td>Model name</td>
<td>“CPX terminal”</td>
</tr>
</tbody>
</table>

1) X: Version Modbus driver, y: Revision code CPX terminal

Tab. 6/29: Objects

6.2.6 Addressing examples for Modbus/TCP

Example 1: CPX terminal with digital I/O modules

Fig. 6/3: CPX terminal with digital I/O modules and MPA pneumatics
### Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Module</th>
<th>Location</th>
<th>Modbus address</th>
<th>Input data</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC Remote I/O</td>
<td>0</td>
<td>45392</td>
<td>Result of access to the I/O diagnostic interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45393</td>
<td>Data from the system table (read access)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45394</td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>Digital 8-input module (8DI)</td>
<td>1</td>
<td>45395</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45396</td>
<td>Input data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>Digital 4-output module (4DO)</td>
<td>2</td>
<td>45397</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45398</td>
<td>Diagnostic data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Echo O-data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>3</td>
<td>45399</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45400</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>4</td>
<td>45401</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45402</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>5</td>
<td>45403</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45404</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>6</td>
<td>45405</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45406</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/30: Input data addressing example 1 (CPX terminal from Fig. 6/3)
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Module</th>
<th>Location</th>
<th>Modbus address</th>
<th>Output data</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC Remote I/O</td>
<td>0</td>
<td>40001</td>
<td>Result of access to the I/O diagnostic interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40002</td>
<td>Data for the system table (write access)</td>
</tr>
<tr>
<td>Digital 4-output module (4DO)</td>
<td>2</td>
<td>40003</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>O-data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>3</td>
<td>40004</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>4</td>
<td>40005</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>5</td>
<td>40006</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>6</td>
<td>40007</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
</tbody>
</table>

Tab. 6/31: Output data addressing example 1 (CPX terminal from Fig. 6/3)
6. Remote I/O Ethernet

Example 2:
CPX terminal with digital and analogue I/O modules

1 CPX-FEC
2 Digital I/O modules
3 Analogue I/O modules
4 MPA pneumatics

Fig. 6/4: CPX terminal with digital and analogue I/O modules as well as MPA pneumatics
### 6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Module</th>
<th>Location</th>
<th>Modbus address</th>
<th>Input data</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC Remote I/O</td>
<td>0</td>
<td>45392</td>
<td>Result of access to the I/O diagnostic interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45393</td>
<td>Data from the system table (read access)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45394</td>
<td>Diagnostic data</td>
</tr>
<tr>
<td>Digital 8-input module (8DI)</td>
<td>1</td>
<td>45395</td>
<td>Input data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45396</td>
<td>Module diagnostic data</td>
</tr>
<tr>
<td>Analogue 2-input module (2AI)</td>
<td>2</td>
<td>45397</td>
<td>Analogue inputs channel 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45398</td>
<td>Analogue inputs channel 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45399</td>
<td>Module diagnostic data</td>
</tr>
<tr>
<td>Analogue 2-output module (2AO)</td>
<td>3</td>
<td>45400</td>
<td>Echo analogue outputs channel 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45401</td>
<td>Echo analogue outputs channel 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45402</td>
<td>Module diagnostic data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>4</td>
<td>45403</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45404</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Module diagnostic data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>5</td>
<td>45405</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45406</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Module diagnostic data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>6</td>
<td>45407</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45408</td>
<td>Echo O-data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Module diagnostic data</td>
</tr>
</tbody>
</table>

Tab. 6/32: Input data addressing example 2 (CPX terminal from Fig. 6/4)
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th>Module</th>
<th>Location</th>
<th>Modbus address</th>
<th>Output data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>–</td>
<td>–</td>
<td>15 8 7 4 3 0</td>
</tr>
<tr>
<td>CPX-FEC Remote I/O</td>
<td>0</td>
<td>40001</td>
<td>Result of access to the I/O diagnostic interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40002</td>
<td>Data for the system table (write access)</td>
</tr>
<tr>
<td>Analogue 2-output module (2AO)</td>
<td>3</td>
<td>40003</td>
<td>Analogue outputs channel 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40004</td>
<td>Analogue outputs channel 1</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>4</td>
<td>40005</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>5</td>
<td>40006</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
<tr>
<td>MPA pneumatic module (8DO)</td>
<td>6</td>
<td>40007</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Output data</td>
</tr>
</tbody>
</table>

Tab. 6/33: Output data addressing example 2 (CPX terminal from Fig. 6/4)
6. Remote I/O Ethernet

6.3 EasyIP

This protocol is used for the fast exchange of operands between Festo controllers (e.g. CPX-FEC, FEC Standard, PS1, etc.).

In the operating mode Remote I/O, the CPX-FEC behaves like an EasyIP server.

The following EasyIP operand types are supported as server:

<table>
<thead>
<tr>
<th>Operand</th>
<th>Operand number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1, flag word</td>
<td>0 ... 9999</td>
</tr>
<tr>
<td>Type 2, input word</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 3, output word</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 4, register</td>
<td>0 ... 255</td>
</tr>
<tr>
<td>Type 5, timer preselect</td>
<td>0 ... 255</td>
</tr>
</tbody>
</table>

Tab. 6/34: EasyIP operand types supported by the CPX-FEC

The processing data are mapped as follows (Mapping):

<table>
<thead>
<tr>
<th>Processing data</th>
<th>EasyIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs (group B)</td>
<td>EasyIP input data</td>
</tr>
<tr>
<td>Outputs (group D)</td>
<td>EasyIP output data</td>
</tr>
</tbody>
</table>

Tab. 6/35: Data exchange with EasyIP
6. Remote I/O Ethernet

### 6.4 Diagnosis

#### 6.4.1 Diagnosis with the control and Ethernet LEDs

Information on the general CPX LEDs PS, PL, SF and M can be found in section 1.4.1.

In the operating mode Remote I/O the control and Ethernet LEDs have the following meanings:

<table>
<thead>
<tr>
<th>RUN</th>
<th>LED (green)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>Modbus connected</td>
</tr>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>No Modbus connection</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STOP</th>
<th>LED (yellow)</th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED is out</td>
<td>ON OFF</td>
<td>Modbus connected</td>
</tr>
<tr>
<td></td>
<td>LED lights up</td>
<td>ON OFF</td>
<td>No Modbus connection</td>
</tr>
</tbody>
</table>

Tab. 6/36: Modbus status displays of the LEDs RUN and STOP in the operating mode Remote I/O Ethernet
6. Remote I/O Ethernet

<table>
<thead>
<tr>
<th><strong>ERROR</strong></th>
<th>Sequence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED (red)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED is out</td>
<td>ON/OFF</td>
<td>The fault LED has no function in the operating mode Remote I/O.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TP (Link/Traffic)</strong></th>
<th>Sequence</th>
<th>Status</th>
<th>Fault treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED (green)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED lights up</td>
<td>ON/OFF</td>
<td>Ethernet connection OK (Link)</td>
<td>–</td>
</tr>
<tr>
<td>LED flashes</td>
<td>ON/OFF</td>
<td>Data traffic (Traffic)</td>
<td>–</td>
</tr>
<tr>
<td>LED is out</td>
<td>ON/OFF</td>
<td>No Ethernet connection or Ethernet cable not connected</td>
<td>If necessary check Ethernet connection</td>
</tr>
</tbody>
</table>

Tab. 6/37: Status displays of the LEDs ERROR and TP
Technical appendix

Appendix A
A. Technical appendix

Contents

A. Technical appendix ................................................................. A-1
A.1 Technical specifications of the Front End Controller CPX-FEC .......... A-2
## A.1 Technical specifications of the Front End Controller CPX-FEC

<table>
<thead>
<tr>
<th>General technical specifications</th>
<th>See CPX system manual:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protection class as per EN 60529</td>
<td>IP65/IP67: CPX-FEC fitted completely, plug connector protection class IP65/IP67 inserted, unused connections fitted with protective cap. IP20 when used with Ethernet patch cables of other manufacturers.</td>
</tr>
<tr>
<td>Protection against electric shock (protection against direct and indirect contact as per EN 60204-1 /IEC 204)</td>
<td>By means of PELV power units (Protected Extra Low Voltage)</td>
</tr>
<tr>
<td>Module code (CPX-specific)</td>
<td>Operating mode:</td>
</tr>
<tr>
<td></td>
<td>– Stand Alone and Remote Controller: 208\textsubscript{d}, D0\textsubscript{h}</td>
</tr>
<tr>
<td></td>
<td>– Remote I/O: 210\textsubscript{d}, D2\textsubscript{h}</td>
</tr>
<tr>
<td>Module identifier (in the handheld)</td>
<td>– FEC Controller (operating mode Stand Alone, Remote Controller)</td>
</tr>
<tr>
<td></td>
<td>– FEC Modbus TCP (operating mode Remote I/O)</td>
</tr>
</tbody>
</table>

### Power supply

<table>
<thead>
<tr>
<th>Operating voltage / load voltage</th>
<th>See CPX system manual:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current consumption of the CPX-FEC</td>
<td>Max. 500 mA at 24 V (only CPX-FEC)</td>
</tr>
</tbody>
</table>

– of operating voltage supply for electronics/sensors ($V_{el/sen}$)
### PLC

<table>
<thead>
<tr>
<th>Programming language</th>
<th>Statement List (STL) or Ladder Diagram (LDR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program memory</td>
<td>250 kB</td>
</tr>
<tr>
<td>Baud rate</td>
<td>1 ... 2 ms per 1 k instructions</td>
</tr>
<tr>
<td>Programming interface</td>
<td>RS232 Sub-D</td>
</tr>
<tr>
<td></td>
<td>9600 ... 57600 Baud</td>
</tr>
</tbody>
</table>

### Ethernet

<table>
<thead>
<tr>
<th>Design</th>
<th>10BaseT/100BaseTX as per IEEE 802.3/802.3u</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>RJ45 in IP20 and IP65/IP67</td>
</tr>
<tr>
<td>Baud rate</td>
<td>10/100 MBaud</td>
</tr>
<tr>
<td>Protocols</td>
<td>– Modbus/TCP</td>
</tr>
<tr>
<td></td>
<td>– EasyIP</td>
</tr>
<tr>
<td>Webserver</td>
<td>Standard Web pages for displaying operands in the controller. Memory space for some Web pages:</td>
</tr>
<tr>
<td></td>
<td>– Operating mode Remote Controller: approx. 550 kB</td>
</tr>
<tr>
<td></td>
<td>– Operating mode Remote I/O: 800 kB</td>
</tr>
</tbody>
</table>
Drivers

Appendix B
## B. Drivers

### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Drivers</td>
<td>B-3</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Drivers for the CPX-FEC (operating mode Remote Controller)</td>
<td>B-3</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Drivers for the CPX-FEC (operating mode Remote I/O Ethernet)</td>
<td>B-5</td>
</tr>
</tbody>
</table>
B. Drivers

B.1 Drivers

B.1.1 Drivers for the CPX-FEC (operating mode Remote Controller)

Standard drivers

The following drivers are contained as standard in the CPX-FEC firmware:

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Name in FST</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC</td>
<td>FECCPX</td>
<td>Standard driver for the CPX-FEC</td>
<td>–</td>
</tr>
<tr>
<td>Modbus/TCP</td>
<td>MODBUSTCP</td>
<td>Drivers for the Ethernet communication with the protocol Modbus/TCP.</td>
<td>Flag word for communication with the master</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>TCPIPCPX</td>
<td>General drivers for the Ethernet communication with the protocol TCP/IP.</td>
<td>IP address, net mask, gateway address</td>
</tr>
<tr>
<td>Webserver</td>
<td>WEB_SRVR</td>
<td>This driver regulates access to the CPX-FEC via a Web browser.</td>
<td>Root directory for the Web pages in the FEC. Standard: B:\WEB</td>
</tr>
</tbody>
</table>

Tab. B/1: Drivers for the CPX-FEC which are loaded automatically

Configuration of the MODBUSTCP driver

Communication

The input and output data are exchanged between the CPX-FEC and the MODBUS/TCP master via a data field of up to 256 flag words. When configuring the driver, enter the number of the starting flag word in the CPX-FEC.
B. Drivers

Installing additional drivers
Further drivers are necessary for some special control tasks.

Loading drivers
1. Open the driver configuration in the FST with a double click on “Driver Configuration” in the project window.
2. Select [Insert][Driver...].
3. Select the desired driver and confirm with OK.

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Name in FST</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM extern</td>
<td>COMEXT</td>
<td>This driver enables characters to be sent to and received from serial interfaces as per RS232.</td>
<td>–</td>
</tr>
<tr>
<td>FpMathDriver</td>
<td>FPMATHDR</td>
<td>Driver for floating comma operations</td>
<td>–</td>
</tr>
<tr>
<td>PID driver</td>
<td>PID</td>
<td>Driver for 16 PID controller</td>
<td>–</td>
</tr>
<tr>
<td>SMTP</td>
<td>SMTPDRV</td>
<td>E-mail driver for sending e-mails from the CPX-FEC.</td>
<td>–</td>
</tr>
<tr>
<td>Data type String</td>
<td>STRINGS</td>
<td>Provides a new data type “String” for strings of characters.</td>
<td>Reserved memory, number of strings, file with pre-assignment</td>
</tr>
</tbody>
</table>

Tab. B/2: Driver of the FST for the CPX-FEC

Detailed information on the drivers can be found in volume 2 of the FST manual.
B. Drivers

B.1.2 Drivers for the CPX-FEC (operating mode Remote I/O Ethernet)

The following drivers are contained as standard in the CPX-FEC firmware:

<table>
<thead>
<tr>
<th>Driver</th>
<th>Name in FST</th>
<th>Description</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX-FEC</td>
<td>FECCPX</td>
<td>Standard driver for the CPX-FEC</td>
<td>–</td>
</tr>
<tr>
<td>Modbus/TCP</td>
<td>MODBUSTCP</td>
<td>Driver for the Ethernet communication with the protocol Modbus/TCP.</td>
<td>–</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>TCPIPCPX</td>
<td>General driver for the Ethernet communication with the protocol TCP/IP.</td>
<td>IP address, net mask, gateway address</td>
</tr>
<tr>
<td>Webserver</td>
<td>WEB_SRVR</td>
<td>This driver regulates access to the CPX-FEC via a Webbrowser.</td>
<td>Root directory for the Web pages in the FEC. Standard: B:\WEB</td>
</tr>
</tbody>
</table>

Tab. B/3: Driver for the CPX-FEC in the operating mode Remote I/O Ethernet

The driver MODBUSTCP is activated automatically in the operating mode Remote I/O Ethernet.

Additional CI commands

The MODBUSTCP driver extends the scope of the command interpreter with the following CI commands:

<table>
<thead>
<tr>
<th>CI command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!35</td>
<td>Display version number and driver information. This display is also shown if an unknown command is entered (e.g. !35?).</td>
</tr>
<tr>
<td>!35TS</td>
<td>Status display of the connections (see following table)</td>
</tr>
</tbody>
</table>

Tab. B/4: CI commands of the Modbus/TCP driver
### B. Drivers

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LISTEN</td>
<td>waiting for tcp_open request from remote</td>
</tr>
<tr>
<td>1</td>
<td>SYNSENT</td>
<td>tcp_open send, waiting for remote</td>
</tr>
<tr>
<td>2</td>
<td>SYNRCVD</td>
<td>tcp_open received, acknowledge send, waiting for remote</td>
</tr>
<tr>
<td>3</td>
<td>ESTABLISHED</td>
<td>connection open, data can be transferred</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>FINWAIT1</td>
<td>tcp_close send, waiting for remote</td>
</tr>
<tr>
<td>6</td>
<td>FINWAIT2</td>
<td>close acknowledged</td>
</tr>
<tr>
<td>7</td>
<td>CLOSEWAIT</td>
<td>not used</td>
</tr>
<tr>
<td>8</td>
<td>CLOSING</td>
<td>our close acknowledged and remote close received</td>
</tr>
<tr>
<td>9</td>
<td>LASTACK</td>
<td>close received, close send, waiting for acknowledge</td>
</tr>
<tr>
<td>10</td>
<td>TIMEOUT</td>
<td>after closing, timer is started after that ( \rightarrow ) CLOSED</td>
</tr>
<tr>
<td>11</td>
<td>CLOSED</td>
<td>connection closed waiting for TCP_RES</td>
</tr>
</tbody>
</table>

Tab. B/5: Possible status values with CI command !35TS

**Communication via Modbus/TCP**

Detailed information on using Modbus/TCP can be found in section 6.2.
Function modules

Appendix C
C. Function modules

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.</td>
<td>Function modules</td>
<td>C-2</td>
</tr>
<tr>
<td>C.1</td>
<td>General function modules for the CPX-FEC</td>
<td>C-3</td>
</tr>
<tr>
<td>C.1.1</td>
<td>Fault message</td>
<td>C-13</td>
</tr>
<tr>
<td>C.2</td>
<td>Modules for special functions of the CPX-FEC</td>
<td>C-14</td>
</tr>
<tr>
<td>C.2.1</td>
<td>Modules for the SMTP driver (send e-mail)</td>
<td>C-14</td>
</tr>
<tr>
<td>C.2.2</td>
<td>Fault codes</td>
<td>C-17</td>
</tr>
</tbody>
</table>
C. Function modules

C.1 General function modules for the CPX-FEC

Overview of modules

<table>
<thead>
<tr>
<th>Modules</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_ST_rd</td>
<td>Read CPX internal parameters and data</td>
</tr>
<tr>
<td>C_ST_wr</td>
<td>Write CPX internal parameters</td>
</tr>
<tr>
<td>C_STATUS</td>
<td>Interrogate diagnostic status</td>
</tr>
<tr>
<td>C_MD_rd</td>
<td>Read module diagnostic data</td>
</tr>
<tr>
<td>C_TR_rd</td>
<td>Read entries in diagnostic memory</td>
</tr>
<tr>
<td>C_MP_rd</td>
<td>Read general module parameters</td>
</tr>
<tr>
<td>C_MP_wr</td>
<td>Write general module parameters</td>
</tr>
<tr>
<td>C_AP_rd</td>
<td>Read special analogue module parameters</td>
</tr>
<tr>
<td>C_AP_wr</td>
<td>Write special analogue module parameters</td>
</tr>
</tbody>
</table>

The modules C_ST_rd and C_ST_wr offer access to all parameters and data of the CPX terminal via the so-called function numbers. The other modules offer access to certain parameters without the need for the function number to be made known.

The function numbers named in this section will assist you in finding information in other manuals. A detailed description of the individual parameters and data as well as the relevant assigned function numbers can be found in the CPX system manual (P.BE-CPX-SYS-..) as well as in the manual for the relevant module (e.g. P.BE-CPX-EA-..).
C. Function modules

Module status

All modules return the so-called module status. The meaning of the module status is explained in the following table.

<table>
<thead>
<tr>
<th>Return values of module status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>-1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6 ... 10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
</tbody>
</table>
C. Function modules

### C_ST_rd

Read CPX internal parameters and data

Enables all parameters and data of the CPX terminal to be read after specification of the relevant function number.

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>FU33</td>
</tr>
</tbody>
</table>

### C_ST_wr

Write CPX internal parameters

Enables all parameters of the CPX terminal to be written after specification of the relevant function number.

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>FU33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
</tbody>
</table>
C. Function modules

C_STATUS Interrogate diagnostic status

The system diagnostic data of the CPX terminal can be read with this module (FU33 ... FU35). You can then ascertain the module number for which there is diagnostic information (FU36 ... FU38). Each bit stands for the relevant module number (0 ... 47). If the bit supplies a 1-signal, there is diagnostic information for this module.

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
<th>*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32 Module status</td>
<td>–</td>
</tr>
<tr>
<td>FU33 CPX status bits</td>
<td>1936</td>
</tr>
<tr>
<td>FU34 First module with fault</td>
<td>1937</td>
</tr>
<tr>
<td>FU35 Fault message of the first module with a fault</td>
<td>1938</td>
</tr>
<tr>
<td>FU36 Diagnostic information exists module 0 ... 15 **</td>
<td>–</td>
</tr>
<tr>
<td>FU37 Diagnostic information exists module 16 ... 31 **</td>
<td></td>
</tr>
<tr>
<td>FU38 Diagnostic information exists module 32 ... 47 **</td>
<td></td>
</tr>
</tbody>
</table>

*) Parameter corresponds to the function number named
**) 1 = there is diagnostic information;
   0 = no diagnostic information
C. Function modules

**C_MD_rd** Read module diagnostic data (FU33 = 0)

Returns all diagnostic data of the module named.

<table>
<thead>
<tr>
<th>Input parameter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>Module number (0 ... 47)</td>
</tr>
<tr>
<td>FU33</td>
<td>Function</td>
</tr>
<tr>
<td></td>
<td>0 = Read module diagnostic data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
<th>*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>Module status</td>
</tr>
<tr>
<td>FU33</td>
<td>Channel number of the first faulty channel</td>
</tr>
<tr>
<td>FU34</td>
<td>Module fault number</td>
</tr>
<tr>
<td>FU35</td>
<td>Information 2 (reserved)</td>
</tr>
<tr>
<td>FU36</td>
<td>Information 3 (reserved)</td>
</tr>
</tbody>
</table>

*) Parameter corresponds to the function number named m = module number (0 ... 47)
C. Function modules

C_MD_rd Read fault numbers of the channels (FU33 = 1)

Supplies the fault numbers of maximum 6 channels. The starting number of the first channel, as from which the fault numbers are to be read, will be specified in FU34.

Detailed information on the possible fault numbers can be found in the CPX system manual as well as in the manual for the relevant module.

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>Module number</td>
</tr>
<tr>
<td>FU33</td>
</tr>
<tr>
<td>Function</td>
</tr>
<tr>
<td>1: Read fault numbers of</td>
</tr>
<tr>
<td>the channels</td>
</tr>
<tr>
<td>FU34</td>
</tr>
<tr>
<td>Number of the first</td>
</tr>
<tr>
<td>channel x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>Module status</td>
</tr>
<tr>
<td>FU33</td>
</tr>
<tr>
<td>Fault number of channel</td>
</tr>
<tr>
<td>x</td>
</tr>
<tr>
<td>FU34</td>
</tr>
<tr>
<td>Fault number of channel</td>
</tr>
<tr>
<td>x + 1</td>
</tr>
<tr>
<td>FU35</td>
</tr>
<tr>
<td>Fault number of channel</td>
</tr>
<tr>
<td>x + 2</td>
</tr>
<tr>
<td>FU36</td>
</tr>
<tr>
<td>Fault number of channel</td>
</tr>
<tr>
<td>x + 3</td>
</tr>
<tr>
<td>FU37</td>
</tr>
<tr>
<td>Fault number of channel</td>
</tr>
<tr>
<td>x + 4</td>
</tr>
<tr>
<td>FU38</td>
</tr>
<tr>
<td>Fault number of channel</td>
</tr>
<tr>
<td>x + 5</td>
</tr>
</tbody>
</table>
C. Function modules

**C_TR_rd**

**Read entries in diagnostic memory**

Enables the diagnostic memory to be read out. The diagnostic memory contains up to 40 diagnostic entries. A diagnostic entry consists of 10 bytes. The first five bytes contain information on the time of the fault. The last five bytes contain information on the fault.

More about the composition of the diagnostic entries can be found in the CPX system manual.

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FU32</strong> Number of the first flag word in which the data are to be saved (0 ... 9999)</td>
</tr>
<tr>
<td><strong>FU33</strong> Number of the first entry in the diagnostic memory as from which reading is to start (0 ... 39)</td>
</tr>
<tr>
<td><strong>FU34</strong> Number of entries (0 ... 40) *)</td>
</tr>
</tbody>
</table>

*) With 0, no diagnostic entries are read, only the information in the return parameters FU33 and FU34 is supplied.

<table>
<thead>
<tr>
<th>Return parameter</th>
<th>*)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FU32</strong> Module status</td>
<td></td>
</tr>
<tr>
<td><strong>FU33</strong> Number of existing entries</td>
<td>3482</td>
</tr>
<tr>
<td><strong>FU34</strong> Overrun and status</td>
<td></td>
</tr>
<tr>
<td>– Bit 0: Overrun (more than 40 entries)</td>
<td></td>
</tr>
<tr>
<td>– Bit 1: Registering inactive</td>
<td></td>
</tr>
<tr>
<td>3483</td>
<td></td>
</tr>
</tbody>
</table>

*) Parameter corresponds to the function number named...
C. Function modules

**C_MP_rd**  
Read general module parameters

Returns the general module parameters of the module named.

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
<th>*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>Module status</td>
</tr>
<tr>
<td>FU33</td>
<td>Parameter byte 0</td>
</tr>
<tr>
<td>FU34</td>
<td>Parameter byte 1</td>
</tr>
<tr>
<td>FU35</td>
<td>Parameter byte 2</td>
</tr>
<tr>
<td>FU36</td>
<td>Parameter byte 3</td>
</tr>
<tr>
<td>FU37</td>
<td>Parameter byte 4</td>
</tr>
<tr>
<td>FU38</td>
<td>Parameter byte 5</td>
</tr>
</tbody>
</table>

*) Parameter corresponds to the function number named  
m = module number (0...47)

Special parameters of analogue modules can be read with module C_AP_rd.
C. Function modules

**C_MP_wr**

Write general module parameters

Enables the general module parameters of the module named to be written.

Detailed information on the module parameters of the module you are using can be found in the manual for the relevant module. There you will also find information on possible parameter values and their presets.

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>Module number (0 ... 47)</td>
</tr>
<tr>
<td>FU33</td>
<td>Parameter byte 0</td>
</tr>
<tr>
<td>FU34</td>
<td>Parameter byte 1</td>
</tr>
<tr>
<td>FU35</td>
<td>Parameter byte 2</td>
</tr>
<tr>
<td>FU36</td>
<td>Parameter byte 3</td>
</tr>
<tr>
<td>FU37</td>
<td>Parameter byte 4</td>
</tr>
<tr>
<td>FU38</td>
<td>Parameter byte 5</td>
</tr>
</tbody>
</table>

*) Parameter corresponds to the function number named m = module number (0 ... 47)

<table>
<thead>
<tr>
<th>Return parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
</tbody>
</table>

Special parameters of analogue modules can be written with module C_AP_wr.
C. Function modules

### C_AP_rd Read analogue module parameters

<table>
<thead>
<tr>
<th>Input parameter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>Module number (0 ... 47)</td>
</tr>
<tr>
<td>FU33</td>
<td>Channel number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
<th>*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>Module status</td>
</tr>
<tr>
<td>FU33</td>
<td>Reserved</td>
</tr>
<tr>
<td>FU34</td>
<td>Monitoring channel 0, 1</td>
</tr>
<tr>
<td>FU35</td>
<td>Lower limit value</td>
</tr>
<tr>
<td>FU36</td>
<td>Upper limit value</td>
</tr>
<tr>
<td>FU37</td>
<td>Measured value smoothing (with input modules)</td>
</tr>
</tbody>
</table>

*) Parameter corresponds to the function number named
m = module number (0 ... 47)

**) Function number depends on module type
(see manual for module)
C. Function modules

C.1. Write analogue module parameters

C.1.1 Fault message

If a fault occurs, the driver will enter the following fault message in the fault word of the CPX-FEC:

<table>
<thead>
<tr>
<th>Fault message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>42,&lt;CPX fault no.&gt;,&lt;Module no.&gt;</td>
<td>CPX fault number 1) and module number of the CPX module on which the fault occurred.</td>
</tr>
</tbody>
</table>

1) See CPX system manual P.BE-CPX-SYS-...
C. Function modules

C.2 Modules for special functions of the CPX-FEC

C.2.1 Modules for the SMTP driver (send e-mail)

Overview of modules

<table>
<thead>
<tr>
<th>Function module</th>
<th>Description</th>
</tr>
</thead>
</table>
| SMTPCFM         | – Status interrogation (FU32 = 0)  
                  – Define sender address and host name (FU32 = 1)  
                  – Send e-mail (FU32 = 2) |

Module SMTPCFM returns a fault code in return parameter FU32 (see section C.2.2).

SMTPCFM

Status interrogation

<table>
<thead>
<tr>
<th>Input parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>FU33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
</tr>
<tr>
<td>FU33</td>
</tr>
<tr>
<td>FU34</td>
</tr>
<tr>
<td>FU35</td>
</tr>
</tbody>
</table>

With the status interrogation you can ascertain whether sending is completed (see following table).
C. Function modules

### SMTPCFM

**Define sender address and mail host**

<table>
<thead>
<tr>
<th>Status</th>
<th>Return values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sending runs</td>
<td>FU32 = 0 and FU33 ≠ 0</td>
</tr>
<tr>
<td>Sending completed successfully</td>
<td>FU32 = 0 and FU33 = 0</td>
</tr>
<tr>
<td>Sending not completed successfully (fault)</td>
<td>FU32 ≠ 0</td>
</tr>
</tbody>
</table>

#### Input parameter

| FU32 | 1 = function: Define sender address and mail host |
| FU33 | Number of the string with e-mail address of the sender |
| FU34 | Number of the string with name or IP address of the mail host |

#### Return parameter

| FU32 | 0 if successful, otherwise fault code |
| FU33 | Status code, 0 if sending is completed |
| FU34 | Fault code |
| FU35 | Extended fault code |
### SMTPCFM: Send e-mail

#### Input parameter

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>2 = function: Send e-mail</td>
</tr>
<tr>
<td>FU33</td>
<td>Number of the string with E-mail address of the receiver</td>
</tr>
<tr>
<td>FU34</td>
<td>Number of the string with e-mail reference</td>
</tr>
<tr>
<td>FU35</td>
<td>Number of the string with which the message began</td>
</tr>
<tr>
<td>FU36</td>
<td>Number of the string with the message contents</td>
</tr>
</tbody>
</table>

#### Return parameter

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FU32</td>
<td>0 if successful, otherwise fault code</td>
</tr>
<tr>
<td>FU33</td>
<td>Status code, 0 if sending is completed</td>
</tr>
<tr>
<td>FU34</td>
<td>Fault code</td>
</tr>
<tr>
<td>FU35</td>
<td>Extended fault code</td>
</tr>
</tbody>
</table>
C. Function modules

C.2.2 Fault codes

If return parameter FU32 supplies ≠ 0, then a fault has occurred:

<table>
<thead>
<tr>
<th>Fault code in FU32</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SMTP driver not in Idle mode</td>
</tr>
<tr>
<td>2</td>
<td>Invalid string number for the sender address</td>
</tr>
<tr>
<td>3</td>
<td>Invalid string length for the sender address</td>
</tr>
<tr>
<td>4</td>
<td>Invalid string number for mail host</td>
</tr>
<tr>
<td>5</td>
<td>Invalid string length for mail host</td>
</tr>
<tr>
<td>6</td>
<td>Invalid string number for the receiver address</td>
</tr>
<tr>
<td>7</td>
<td>Invalid string length for the receiver address</td>
</tr>
<tr>
<td>8</td>
<td>Invalid string number for reference</td>
</tr>
<tr>
<td>9</td>
<td>Invalid string length for reference</td>
</tr>
<tr>
<td>10</td>
<td>Invalid string number(s) for message</td>
</tr>
<tr>
<td>99</td>
<td>Invalid parameters</td>
</tr>
<tr>
<td>100</td>
<td>SMTP driver not loaded</td>
</tr>
<tr>
<td>101</td>
<td>TCP/IP driver not loaded</td>
</tr>
<tr>
<td>102</td>
<td>STRING driver not loaded</td>
</tr>
</tbody>
</table>
C. Function modules

The following fault codes can appear in FU33:

<table>
<thead>
<tr>
<th>Fault code in FU33</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>Invalid parameters</td>
</tr>
<tr>
<td>100</td>
<td>SMTP driver not loaded</td>
</tr>
<tr>
<td>101</td>
<td>TCPIP driver not loaded</td>
</tr>
<tr>
<td>102</td>
<td>STRING driver not loaded</td>
</tr>
<tr>
<td>103</td>
<td>Fault in deleting the mail host (DNS)</td>
</tr>
<tr>
<td>104</td>
<td>Timeout in deleting the mail host (DNS)</td>
</tr>
<tr>
<td>105</td>
<td>Timeout in connection to the mail host</td>
</tr>
<tr>
<td>106</td>
<td>Timeout, no (more) replies received from mail host</td>
</tr>
<tr>
<td>107</td>
<td>TCP connection to mail host lost</td>
</tr>
<tr>
<td>255</td>
<td>Mail host has registered a fault. Check the fault code in FU35.</td>
</tr>
</tbody>
</table>
The Command Interpreter

Appendix D
D. The Command Interpreter

Contents

D. The Command Interpreter .......................................................... D-1
  D.1 The Command Interpreter (CI) .............................................. D-3
  D.1.1 Connection to a dialog device ........................................... D-3
  D.1.2 Selecting the command interpreter (Login) ......................... D-4
  D.1.3 Exiting the command interpreter ...................................... D-6
  D.1.4 CI command ................................................................. D-6
  D.1.5 Displaying operands and statuses with Display (D) ............ D-10
  D.1.6 Changing operands with Modify (M) ................................. D-14
  D.1.7 Commands for program controller ................................. D-19
  D.1.8 Commands for forcing inputs and outputs ....................... D-19
  D.1.9 Initialising user memory .............................................. D-22
  D.1.10 Password .................................................................. D-22
  D.1.11 Driver-specific commands ............................................. D-22
  D.1.12 Linking CI commands .................................................. D-22
D. The Command Interpreter

D.1 The Command Interpreter (CI)

The Command Interpreter, “CI” for short, enables the controllers to be operated externally by a Terminal or Terminal Emulator and represents the interface for online operation of FST.

**Note**
The FST contains a Terminal Emulator (see Volume 1 of the FST manual) that enables you to send manual CI commands to the Command Interpreter. The term “Command” is used below to mean both command and instruction.

D.1.1 Connection to a dialog device

To operate the command interpreter, it must be connected to a suitable dialog device. The following options are available:

- PC with RS232 or TCP/IP port and Terminal Emulator (e.g. CI Terminal of FST, see Volume 1 of the FST manual)
- Terminal with RS232 or TCP/IP port.

**Serial port**
The serial port on the controller to be used by the CI for communication can usually be set under “Controller Settings” in the project.

**TCP/IP**
If the correct FST drivers are installed, the CI can also be accessed via additional COM ports or TCP/IP.

**Note**
Please note that some functions of the additional CI ports are limited.
D. The Command Interpreter

D.1.2 Selecting the command interpreter (Login)

**Caution**
The command interpreter (CI) contains commands that reorganise or delete parts of the memory. This destroys existing data.
- Only use CI commands if you know their effects!

**Login with FST**
All online functions of the FST use the CI. You can also send manual CI commands via the CI terminal integrated into FST (see Volume 1 of the FST manual).

**Login with Terminal or Terminal Emulator**
The CI is registered on a connected terminal after either DC4 (Control T) has been entered or a hardware break has been transferred. Any command currently being processed is cancelled.

```
DC4 (Ctrl T)
```

The controller responds by displaying the version number of the runtime main program and its standard prompt “>” in the next line.

```
FESTO IPC V2.nn
>
```
D. The Command Interpreter

Transfer of BREAK is accompanied by setting of transfer speed on the controller to 9600 and/or 2400 baud. There are 4 distinct methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When BREAK is received, the speed is changed in cycles twice to 9600 baud and once to 2400 baud. This is the default method.</td>
</tr>
<tr>
<td>2</td>
<td>9600 baud is always set (old method).</td>
</tr>
<tr>
<td>3</td>
<td>2400 baud is always set (advisable with slow modem connections and when using Field PC Net alias MpRAM).</td>
</tr>
<tr>
<td>4</td>
<td>When BREAK is received, the speed is changed in cycles three times to 2400 baud and twice to 2400 baud.</td>
</tr>
</tbody>
</table>

Tab. D/6: Methods for speed transfer

All methods enable the baudrate to be set to any speed after login (see CI command MV). After booting, Method 1 is set. Methods 1, 2 and 4 also enable use of a previous FST host software. However, login may occasionally fail. If this is the case, try again. Follow the instructions for FST.

FST knows these new login methods and will try to modify them. The login methods can also be set with function module COM1METH (see Volume 2 of the FST manual).
D. The Command Interpreter

D.1.3 Exiting the command interpreter

The X command frees the serial interface used by CI. This command will only function if it has been entered via the serial interface.

X<CR>

No message is sent by the Command Interpreter.

D.1.4 CI command

The sections below describe the valid CI commands. Drivers can receive their own CI commands (see also Section D.1.11). These are indicated in Volume 2.

Command structure Each CI command has a defined input format. These include:

- a command letter
- a parameter (letter or number, depending on parameter)
- a value (not always required).

<table>
<thead>
<tr>
<th>Input format</th>
<th>Message from CI</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Command letter&gt; [ &lt;Parameter&gt; ] [ =Value ]</code></td>
<td>Dependent on command</td>
</tr>
</tbody>
</table>

Tab. D/7: Input format and message from CI

Both upper and lower case characters can be entered. Conclude entries with <CR>.

Incorrect entries can be changed using the backspace key (Ctrl H) before they are concluded with Enter.
D. The Command Interpreter

Command letter The table below shows valid command letters:

<table>
<thead>
<tr>
<th>Command letter</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>BREAK Program run interrupted</td>
</tr>
<tr>
<td>DC4 (Ctrl T)</td>
<td>LOGIN Login</td>
</tr>
<tr>
<td>D</td>
<td>DISPLAY Displays operands</td>
</tr>
<tr>
<td>LC</td>
<td>PASSWORD Enter/change password</td>
</tr>
<tr>
<td>LX</td>
<td>PASSWORD Password protection on/off</td>
</tr>
<tr>
<td>M</td>
<td>MODIFY Modify: Changes operands</td>
</tr>
<tr>
<td>R</td>
<td>RUN Starts/continues program</td>
</tr>
<tr>
<td>S</td>
<td>STOP Stops program</td>
</tr>
<tr>
<td>X</td>
<td>LOGOUT Enables serial port</td>
</tr>
<tr>
<td>Y</td>
<td>INIT Deletes user memory</td>
</tr>
</tbody>
</table>

Tab. D/8: Command letters

Parameters The table below shows the possible parameters.
D. The Command Interpreter

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Meaning</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A[&lt;YN&gt;.]&lt;wN&gt;.&lt;BN&gt;</td>
<td>Output bit</td>
<td>Instead of the abbreviation, enter the valid value. The value range is dependent on operand type.</td>
</tr>
<tr>
<td>AW[&lt;YN&gt;.]&lt;WN&gt;</td>
<td>Output word</td>
<td></td>
</tr>
<tr>
<td>B&lt;BN&gt;</td>
<td>Program module</td>
<td></td>
</tr>
<tr>
<td>BF&lt;BN&gt;</td>
<td>Function module</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Display format</td>
<td></td>
</tr>
<tr>
<td>E[&lt;YN&gt;.]&lt;WN&gt;.&lt;BN&gt;</td>
<td>Input bit</td>
<td></td>
</tr>
<tr>
<td>EW[&lt;YN&gt;.]&lt;WN&gt;</td>
<td>Input word</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Error word</td>
<td></td>
</tr>
<tr>
<td>M&lt;wN&gt;.&lt;BN&gt;</td>
<td>Flag bit</td>
<td></td>
</tr>
<tr>
<td>MW&lt;wN&gt;</td>
<td>Flag word</td>
<td></td>
</tr>
<tr>
<td>O&lt;wN&gt;</td>
<td>Global function units FU0 ... FU31 and FU39 ... FU255</td>
<td></td>
</tr>
<tr>
<td>O&lt;PN&gt;.&lt;WN&gt;</td>
<td>Local function units FU32... FU38</td>
<td></td>
</tr>
<tr>
<td>P&lt;PN&gt;</td>
<td>Program status</td>
<td></td>
</tr>
<tr>
<td>R&lt;RN&gt;</td>
<td>Index</td>
<td></td>
</tr>
<tr>
<td>S&lt;PN&gt;</td>
<td>Program initialisation flag</td>
<td></td>
</tr>
<tr>
<td>T&lt;TN&gt;</td>
<td>Pulse timer</td>
<td></td>
</tr>
<tr>
<td>TA&lt;TN&gt;</td>
<td>Switch-off delay timer</td>
<td></td>
</tr>
<tr>
<td>TE&lt;TN&gt;</td>
<td>Switch-on delay timer</td>
<td></td>
</tr>
<tr>
<td>TV&lt;TN&gt;</td>
<td>Timer pre-setting</td>
<td></td>
</tr>
<tr>
<td>TW&lt;TN&gt;</td>
<td>Timer word</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Baud rate</td>
<td></td>
</tr>
<tr>
<td>Z&lt;CN&gt;</td>
<td>Counters</td>
<td></td>
</tr>
<tr>
<td>ZV&lt;CN&gt;</td>
<td>Counter pre-setting</td>
<td></td>
</tr>
<tr>
<td>ZW&lt;CN&gt;</td>
<td>Counter word</td>
<td></td>
</tr>
</tbody>
</table>

Tab. D/9: Parameters
D. The Command Interpreter

Value  The permitted values depend on the respective parameter and/or operands.

CI response  The table below shows the response from the CI to valid and invalid CI commands:

<table>
<thead>
<tr>
<th>CI command</th>
<th>CI response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td><code>&lt;Command&gt;“\r”</code></td>
</tr>
<tr>
<td>Response to valid commands</td>
<td><code>&lt;Command&gt;&lt;Response&gt;“\n\r\n21”</code></td>
</tr>
<tr>
<td>Response to invalid commands</td>
<td><code>&lt;Command&gt;“\b\nACCESS ERROR\n\r\n21”</code></td>
</tr>
</tbody>
</table>

For invalid commands, either “ACCESS ERROR” or (rarely) its abbreviation “ERR” appears. A beep then also sounds in the speaker.
D. The Command Interpreter

D.1.5 Displaying operands and statuses with Display (D)

Display

Display enables you to show the statuses and contents of the operands and also the current status of the programs.

Example: Display Output O0.1

Input

>DA0.1

Output (example)

>DA0.1=0

The response from the command interpreter is always displayed in the input line. The entered characters “D”, “A0.1” and “CR” (Enter) are sent to the controller immediately. The controller returns “D”, “A0.1” and the response “=0”. The response is concluded with “CR”, “LF” and “>”.

Display commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA[YN].&lt;WN&gt;.&lt;BN&gt;</td>
<td>Displays output bit</td>
</tr>
<tr>
<td>DAW[YN].&lt;WN&gt;</td>
<td>Displays output word</td>
</tr>
<tr>
<td>DB&lt;BN&gt;</td>
<td>Displays program module</td>
</tr>
</tbody>
</table>

Response: “={Type},0, {Status}, {Step}”.

- The first value is the module type, STL=0, LDR/FUP=1 or C=2.
- The second value, memory area, is always 0.
D. The Command Interpreter

- The third value indicates the status of the selecting program.
- The final value is the current step number within the module.

<table>
<thead>
<tr>
<th>DBF&lt;BN&gt;</th>
<th>Displays function module</th>
</tr>
</thead>
</table>

Response as for DB<BN>.

<table>
<thead>
<tr>
<th>DD</th>
<th>Shows display format of multi-bit operands</th>
</tr>
</thead>
</table>

Response:
“=D” for displaying decimal without sign
“=S” for displaying decimal with sign
“=H” for displaying hexadecimal

<table>
<thead>
<tr>
<th>DE[&lt;YN&gt;.]&lt;WN&gt;.&lt;BN&gt;</th>
<th>Displays input bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEW[&lt;YN&gt;.]&lt;WN&gt;</td>
<td>Displays input word</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DF</th>
<th>Displays error word</th>
</tr>
</thead>
</table>

Command DE requests the error status from the controller. If no error has occurred, the controller responds with “=0,0,0”.

<table>
<thead>
<tr>
<th>Error type</th>
<th>Set-up of the CI response</th>
</tr>
</thead>
<tbody>
<tr>
<td>General errors</td>
<td>=&lt;Error number&gt;,&lt;Program number&gt;,&lt;Step number&gt; (^1)</td>
</tr>
<tr>
<td>CPX error (42)</td>
<td>=&lt;42&gt;,&lt;CPX error number&gt;,&lt;CPX module number&gt;</td>
</tr>
<tr>
<td>I/O error (11, 12)</td>
<td>=&lt;Error no.&gt;,&lt;255&gt;,&lt;No. of input or output word&gt;</td>
</tr>
</tbody>
</table>

\(^1\) The error number corresponds to the value of the error word (see also volume 1 of the FST manual); program number in which the error occurred; if the program has no steps (e.g. with LDR programs), Step 0 is displayed.

For example: “=42,5,1”.

For example: “=42,5,1”.
D. The Command Interpreter

<table>
<thead>
<tr>
<th>DM&lt;WN&gt; . &lt;BN&gt;</th>
<th>Displays flag bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMW&lt;WN&gt;</td>
<td>Displays flag word</td>
</tr>
</tbody>
</table>

| DO<WN>         | Displays function unit |

Global function units FU0 to FU31 and FU39 to FU255 can be displayed.

| DO<PN> . <WN> | Displays local function unit |

Local function units FU32 to FU38 can be displayed. There are separate function units for each program.

| DP<PN>         | Displays program status |

The response contains 6 values:

The first value denotes the program type:
STL=0, LDR/FUP=1 or C=2.

The second value, memory area, is always 0.

The third value denotes the program status:
0 for inactive, 2 for active but halted, or 3 for active.

The fourth value denotes the step number: unequal to zero for STL step programs and LDR programs with jumps as long as the program is active. If a step program is not active, it is in Step 0.

The final two values denote the numbers and step numbers of the selected module.

| DR<RN>         | Displays register |

| DS<PN>         | Displays program initialisation flag |
D. The Command Interpreter

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT&lt;TN&gt;</td>
<td>Displays status for pulse timer</td>
</tr>
<tr>
<td>DTA&lt;TN&gt;</td>
<td>Displays status for switch-off delayed timer</td>
</tr>
<tr>
<td>DTE&lt;TN&gt;</td>
<td>Displays status for switch-on delayed timer</td>
</tr>
<tr>
<td>DTV&lt;TN&gt;</td>
<td>Displays timer pre-setting</td>
</tr>
<tr>
<td>DTW&lt;TN&gt;</td>
<td>Displays timer word</td>
</tr>
<tr>
<td>DV</td>
<td>Displays baud rate</td>
</tr>
</tbody>
</table>

The DV command indicates the current baudrate. Possible values are “=1200”, “=2400”, “=4800”, “=9600”, “=19200”, “=38400” or “=56000”.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DZ&lt;CN&gt;</td>
<td>Displays counter status</td>
</tr>
<tr>
<td>DZV&lt;CN&gt;</td>
<td>Displays counter pre-setting</td>
</tr>
<tr>
<td>DZW&lt;CN&gt;</td>
<td>Displays counter word</td>
</tr>
</tbody>
</table>
D. The Command Interpreter

D.1.6 Changing operands with Modify (M)

Modify

Modify enables you to change the contents and/or statuses of operands.

- To modify an operand directly, without previous display, enter the required value after the prompt and confirm the entry with Enter <CR>.

Input

```plaintext
>MAW1=255
```

Output

```plaintext
>MAW1=255
```

- Communicating via RS232 enables you to display the contents and/or the status of the operand beforehand. Enter only command letter M and the operand and then press Enter <CR>.

Input

```plaintext
>MAW1
```

Output (example)

```plaintext
>MAW1=255:
```

The CI reports the current value. After the colon, you can enter the new value and confirm by pressing <CR>.

The values can be entered in decimal, hexadecimal and signed decimal notation (see Display format).
Modify commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`MA[&lt;YN&gt;.&lt;WN&gt;.&lt;BN&gt;={0</td>
<td>1}`</td>
</tr>
<tr>
<td><code>MAW[&lt;YN&gt;.&lt;WN&gt;=&lt;Value&gt;</code></td>
<td>Modifies output word</td>
</tr>
<tr>
<td>`MD={D</td>
<td>S</td>
</tr>
<tr>
<td>`ME[&lt;YN&gt;.&lt;WN&gt;.&lt;BN&gt;={0</td>
<td>1}`</td>
</tr>
<tr>
<td><code>MEW[&lt;YN&gt;.&lt;WN&gt;=&lt;Value&gt;</code></td>
<td>Modifies input word</td>
</tr>
<tr>
<td><code>MF=&lt;Value&gt;</code></td>
<td>Modifies error word</td>
</tr>
</tbody>
</table>

The display format can be set to decimal without sign “=D”, decimal with sign “=S” or hexadecimal “=H”.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`MM&lt;WN&gt;.&lt;BN&gt;={0</td>
<td>1}`</td>
</tr>
<tr>
<td><code>MFW&lt;WN&gt;=&lt;Value&gt;</code></td>
<td>Modifies flag word</td>
</tr>
<tr>
<td><code>MO&lt;WN&gt;=&lt;Value&gt;</code></td>
<td>Modifies global function unit</td>
</tr>
</tbody>
</table>

Modifies global function units FU0 to FU31 and FU39 to FU255.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MO&lt;PN&gt;.&lt;WN&gt;=&lt;Value&gt;</code></td>
<td>Modifies local function unit</td>
</tr>
</tbody>
</table>

Modifies local function units FU32 to FU38. There are separate function units for each program.
### The Command Interpreter

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR&lt;RN&gt;=&lt;Value&gt;</td>
<td>Modifies register</td>
</tr>
<tr>
<td>MT&lt;TN&gt;={0</td>
<td>1}</td>
</tr>
<tr>
<td>MTA&lt;TN&gt;={0</td>
<td>1}</td>
</tr>
<tr>
<td>MTE&lt;TN&gt;={0</td>
<td>1}</td>
</tr>
<tr>
<td>MTV&lt;TN&gt;=&lt;Value&gt;</td>
<td>Modifies timer pre-setting</td>
</tr>
<tr>
<td>MTW&lt;TN&gt;=&lt;Value&gt;</td>
<td>Modifies timer word</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV=&lt;Baudrate&gt;</td>
<td>Sets baudrate</td>
</tr>
</tbody>
</table>

The baudrate can be set using commands “MV=1200”, “MV=2400”, “MV=4800”, “MV=9600”, “MV=19200”, “MV=38400” or “MV=56000”. The value can be shortened to 2 characters, for example “MV=96”.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MZ&lt;CN&gt;={0</td>
<td>1}</td>
</tr>
<tr>
<td>MZV&lt;CN&gt;=&lt;Value&gt;</td>
<td>Sets counter pre-setting</td>
</tr>
<tr>
<td>MZW&lt;CN&gt;=&lt;Value&gt;</td>
<td>Sets counter word</td>
</tr>
</tbody>
</table>
D. The Command Interpreter

D.1.7 Commands for program controller

**B** Break

<table>
<thead>
<tr>
<th>B</th>
<th>Interrupts all running programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP&lt;PN&gt;</td>
<td>Interrupts the indicated program</td>
</tr>
</tbody>
</table>

**R** Run

<table>
<thead>
<tr>
<th>R</th>
<th>Starts or continues program</th>
</tr>
</thead>
</table>

If the “Reset programs” option is selected in the PLC settings, Program P0 is started or run on. If the option is not selected, all halted (interrupted programs) are run on.

**RB<Number>[,<FU32>[,,<FU33>[,, ... [,,<FU37>[,,<FU38>]]]]]]

Selects program module

The “RB” command selects a loaded program module (one that is contained in the project file).

**Note**
The command uses the local function units of Program P63, which should be reserved for this purpose.

The call parameters must be indicated. If no parameter is indicated, its last value is used. The response:

`=FU32,FU33,FU34,FU35,FU36,FU37,FU38`. 

Festo RBE-CPX-FEC-EN en 0404NH D-17
Example: Select CMP 7 with FU32=14, FU33=old value, FU34=9. Result FU32=4712, FU33=103, FU34 to 38=0.

\[
\text{RB7,14,,9}=4712,103,0,0,0,0,0
\]

The “RF” command selects a loaded function module (one that is contained in the project file).

**Note**
The command uses the local function units of Program P63, which should be reserved for this purpose.

The call parameters must be indicated. If no parameter is indicated, its last value is used. The response:

\[
=\text{FU32},\text{FU33},\text{FU34},\text{FU35},\text{FU36},\text{FU37},\text{FU38}.
\]

**S** Stop

<table>
<thead>
<tr>
<th><strong>S</strong></th>
<th>Stops all programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SP&lt;PN&gt;</strong></td>
<td>Stops program &lt;PN&gt;</td>
</tr>
</tbody>
</table>
D. The Command Interpreter

D.1.8 Commands for forcing inputs and outputs

All digital inputs and outputs can be forced selectively to 0 or 1. If an input bit is forced to 0 or 1, it can be detected by the programs and the CI. If an output bit is forced to 0 or 1, it cannot be detected by the program and the CI. Further information can be found in Section volume 1 of the FST manual under “Forcing Inputs and Outputs”.

The Force table is not retentive. It is automatically deleted by the Y command or by downloaded a project.

The following CI commands are available for forcing I/Os.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YF</td>
<td>Deletes Force table</td>
</tr>
<tr>
<td>DAF&lt;WN&gt;.&lt;BN&gt;</td>
<td>Displays output bit</td>
</tr>
</tbody>
</table>

Result:

=0: Forced to 0  
=1: Forced to 1  
=N: Not forced
D. The Command Interpreter

**DAWF<WN>** Displays output word

Result: “=xxxxxxxxxxxxxxxx”, bitwise with:

- =0: Forced to 0
- =1: Forced to 1
- =N: Not forced

**DEF<WN>.<BN>** Displays input bit

Result:

- =0: Forced to 0
- =1: Forced to 1
- =N: Not forced

**DEWF<WN>** Displays input word

Result: “=xxxxxxxxxxxxxxxx”, bitwise with:

- =0: Forced to 0
- =1: Forced to 1
- =N: Not forced

**MAF<WN>.<BN>={0 | 1 | N}** Enters output bit into Force table

- =0: Forces to 0
- =1: Forces to 1
- =N: Do not force

**MAWF<WN>={Value | N}** Enters output word into Force table

- =Value: Forces to this value
- =N: Do not force
D. The Command Interpreter

\[ \text{MEF}<WN>.<BN>=\{0 \mid 1 \mid N\} \]

Enters input word into Force table

- =0: Forces to 0
- =1: Forces to 1
- =N: Do not force

\[ \text{MEWF}<WN>=\{\text{Value} \mid N\} \]

Enters input word into Force table

- =Value: Forces to this value
- =N: Do not force

D.1.9 Initialising user memory

Y

Initialising

Caution

The Y! command deletes all project data and drivers from the RAM memory.

Y

Deletes all project data and drivers from the RAM memory, with call-back

Y!

Deletes all project data and drivers from the RAM memory, without call-back
D.1.10 Password

The following CI commands enable you to enter, change or delete the password online and also activate or deactivate password protection.

A password consists of between 3 and 20 visible ASCII characters. Separators such as commas, spaces, tab, IBM extended characters etc. are not permitted. Further information on password protection can be found in volume 1 of the FST manual.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC&lt;old&gt;,&lt;new&gt;</td>
<td>Enter/change password</td>
</tr>
<tr>
<td></td>
<td>&lt;Old&gt;: Old password</td>
</tr>
<tr>
<td></td>
<td>&lt;New&gt;: New password</td>
</tr>
<tr>
<td>LC,&lt;new&gt;</td>
<td></td>
</tr>
</tbody>
</table>

When a new password is entered, the old password must always be indicated too. LCTEST,FEC changes the password from TEST to FEC, for instance. If no password previously existed, the old password does not need to be entered. The comma, however, must still be entered e.g. LC,FEC.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LX</td>
<td>Password protection on (logout)</td>
</tr>
<tr>
<td>LX&gt;Password</td>
<td>Password protection off (login)</td>
</tr>
</tbody>
</table>

The LX command is also used to login or logout.

For example, if the password is “FEC”, then:

- password protection is de-activated with LXFEC (login).
- command LX or LX with incorrect password activates password protection (logout).
D. The Command Interpreter

D.1.11 Driver-specific commands

The FST PLC operating system enables the drivers to receive their own commands. Driver-specific CI commands begin with an exclamation mark “!” and the driver number with <DN>, the command itself then follows.

\[!\text{<DN>}<\text{Command}>\]

Driver-specific CI commands are indicated in Volume 2 of the FST manual.

A driver does not necessarily have to have its own commands. Many drivers respond to an empty command with status information. Driver-specific commands generally have a similar set-up to the standard CI commands.

For example, the string driver with the number 3 manages display commands for character chains, in which the corresponding string number is used.

Example:

\[!3D12=’\text{Festo}’\]

>
D. The Command Interpreter

D.1.12 Linking CI commands

Almost all commands can be linked. The CI processes the command in sequence and the responses are grouped. The command groups must be separated by a semicolon (see Example 1).

The commands of a command group (e.g. successive Display or Modify commands) can be separated by a comma and the command symbol itself (“D” or “M”) is not repeated (see Example 2).

Example 1

The commands for starting Program P0 and for requesting the program status are:

```
>RP0
>DP0=0,0,3,2,0,0
>RP0;DP0
```

The same command sequence links:

Input

```
>RP0;DP0
```

Output (example)

```
>RP0;DP0=0,0,3,2,0,0
```

Example 2

R0, FW16 and I0.3 are to be displayed. As individual commands:

```
>DR0=432
>DMW16=0
>DE0.3=1
```
The same command sequence links:

Input

>DR0,MW16,E0.3

Output (example)

>DR0,MW16,E0.3=432=0=1

Multi-line commands cannot be linked, e.g. Modify com-
mmands with display of current value. Linking is also impos-
sible with commands that are transferred to a driver.

Mass display

Commands used to display values can be suffixed with a
minus sign. 16 successive values are then shown as a mass
display. This display method is also valid for bit operands.

Example

The “DR1” command displays register 1.

>DR1=0

By contrast, the “DR1” command displays registers 1 to 16.

>DR1==0==0==0==0==0==0==0==0==0==0==0==0==0==0
D. The Command Interpreter
Index

Appendix E
E. Index

Contents

E. Index ........................................................................................................... E-2
# Index

## A

- Abbreviations ............................................. XIV
- Addressing .............................................. 3-1
- Addressing example, General operation .............. 3-2
- Addressing examples, Remote I/O Modbus/TCP ...... 6-26
- Application layer ........................................ 4-1

## C

- Cable
  - Ethernet ............................................... 2-1
  - FED .................................................... 2-1
  - Handheld .............................................. 2-18
  - Programming cable .................................. 2-14
- CFM ....................................................... C-3
- CI .......................................................... 3-6, D-7
- CI Commands ............................................. D-6
  - Break (B) ............................................... D-17
  - Command letter ...................................... D-7
  - Command structure .................................. D-6
  - Display (D) ............................................ D-10
  - Driver-specific commands ......................... D-23
  - Init (Y!) ................................................ D-21
  - Linking ................................................ D-24
  - Modify (M) ............................................. D-14
  - Parameters ............................................ D-7
  - Run (R) ................................................ D-17
  - Stop (S) ............................................... D-18
- CI commands ............................................. 3-6
  - As HTTP query ....................................... 4-25
  - In HTML .............................................. 4-26
- Command Interpreter ................................. 3-6, D-7
- Command interpreter, Login ......................... D-7

## Configuration

- General information (Stand Alone) ................... 3-4
- Remote Controller Ethernet .......................... 5-
E. Index

Remote Controller Field bus ....................... 5-15
Remote I/O Ethernet .......................... 6-7
Connecting and display elements .......... 2-5
Create Web pages ......................... 4-25

D

Default addressing ......................... 3-19
Designated use ................................ VIII
DHCP ........................................ 4-8
Diagnosis
  In the user program ....................... 3-59
  Remote Controller Field bus .......... 5-21
  Remote I/O Ethernet ..................... 6-57
  Stand Alone / Remote Controller Ethernet .................. 3-57
  With the Hardware Configurator ........ 3-57
Diagnosis Trace ........................... 3-59
Diagnostic memory parameters ........ 3-33
DIL switches ................................ 2-9
Dismantling ................................ 2-6
Download Project ......................... 3-46
Drivers ..................................... B-3

E

E-mail ........................................ 4-18
EasyIP
  Remote Controller Ethernet ............. 5-10
  Remote I/O Ethernet .................... 6-3
Electrical connecting and display elements ........ 2-5
Ethernet interface ....................... 2-1, 4-17
### Index

#### F

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FED-50/90</td>
<td>2-1</td>
</tr>
<tr>
<td>Field bus node (Remote Controller)</td>
<td>5-1</td>
</tr>
<tr>
<td>Fitting</td>
<td>2-6</td>
</tr>
<tr>
<td>Forcing</td>
<td>3-38</td>
</tr>
<tr>
<td>Front End Display</td>
<td>2-1</td>
</tr>
<tr>
<td>FST</td>
<td></td>
</tr>
<tr>
<td>Addressing</td>
<td>3-19</td>
</tr>
<tr>
<td>Checking the configuration</td>
<td>3-1</td>
</tr>
<tr>
<td>Configuration</td>
<td>3-10</td>
</tr>
<tr>
<td>Creating a project</td>
<td>3-5</td>
</tr>
<tr>
<td>Programming</td>
<td>3-38</td>
</tr>
<tr>
<td>Function module status</td>
<td>C-4</td>
</tr>
<tr>
<td>Function modules</td>
<td>3-45,C-3</td>
</tr>
<tr>
<td>Module status</td>
<td>C-4</td>
</tr>
</tbody>
</table>

#### G

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>4-8</td>
</tr>
</tbody>
</table>

#### H

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handheld</td>
<td>2-18</td>
</tr>
<tr>
<td>HTTP</td>
<td>4-16</td>
</tr>
</tbody>
</table>

#### I

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Important user instructions</td>
<td>XI</td>
</tr>
<tr>
<td>Installation</td>
<td>2-4</td>
</tr>
<tr>
<td>IP address</td>
<td>4-17</td>
</tr>
<tr>
<td>IP addressing</td>
<td>4-13</td>
</tr>
<tr>
<td>IP65/IP67</td>
<td>2-19</td>
</tr>
</tbody>
</table>
# Index

## L
- Layers 1 to 4 ........................................ 4-9
- Layers 5 to 7 ........................................ 4-1
- LEDs
  - Remote Controller Ethernet .................. 5-1
  - Remote Controller Field bus ................. 5-2
  - Remote I/O Ethernet ........................... 6-3
  - Stand Alone .................................... 3-5
- LEDs ................................................. 1-1

## M
- MAC-ID ................................................ 4-6
- Modbus/TCP
  - Remote Controller Ethernet .................. 5-7
  - Remote I/O Ethernet ........................... 6-7
- Module parameters ............................... 3-3

## N
- Net classes .......................................... 4-7
- Net mask ............................................ 4-7
- Notes on the use of this manual ................

## O
- Operands ............................................ 3-40
- Operating mode ................................... 2-10
  - Remote Controller Field bus ................. 5-1
- Operating modes ................................... 1-5
  - Overview ....................................... 1-5
  - Remote Controller Ethernet ..................
  - Remote I/O Ethernet ...........................
  - Stand Alone ..................................
- Operations ....................................... 3-4

---

E-6

Festo  PBE-CPX-FEC-EN  en 0404NH
### E. Index

#### P
- Parametrizing ............................................. 3-2
- Example of application .................................. 3-3
- From the user program .................................. 3-3
- Remote Controller Field bus .............................. 5-2
- When switching on ........................................ 3-3
- With handheld .............................................. 3-2
- Password, Setting by CI .................................. D-2
- Pictograms ................................................. XII
- Pin assignment
  - Ethernet interface ...................................... 2-1
  - Programming interface ................................ 2-1
- PLC ......................................................... 1-9, 3-39, A-1
- Programming ................................................ 3-3
- Program example ......................................... 3-50
- Sending an e-mail ........................................ 4-1
- Programming cable ....................................... 2-1
- Programming interface .................................. 2-1
- Protection class IP65/IP67 ................................. 2-1
- Protocols and services .................................... 4-12

#### R
- Remote Controller Ethernet (operating mode) ........ 5-1
- Remote Controller Field bus, Configuration example .. 5-16
- Remote Controller Field bus (operating mode) ........ 5-11
- Remote I/O Ethernet (operating mode) .................... 6-4
- Rotary switch .............................................. 2-6
- Run time behaviour of the controller .................. 3-48

#### S
- Serial port .................................................. D-1
- Service ..................................................... IX

---

Festo RBE-CPX-FEC-EN en 0404NH
Service interface for handheld .......................... 2-18
Setting, Operating mode ................................. 2-7
SMTP ......................................................... 4-18, C-14
Software status ............................................. 4-18
Stand Alone (operating mode) ........................... 3-4
Switch ......................................................... 4-10
System parameters ......................................... 3-3

T
Target group .................................................. 11
TCP/IP ......................................................... 4-7, D-3
Technical specifications .................................. A-3
Text markings ............................................... XI
TFTP .......................................................... 4-17
Tutorial ......................................................... 1-15

U
UDP .......................................................... 4-6
URL .......................................................... 4-16

W
Webserver ..................................................... 4-21
Load files ..................................................... 4-28