ifm electronic

CE

Software manual
Downloader 32

ecomat100®

ecomat mobile

ecolog V2.1
CoDeSys® V2.3

English



7390659 / 00 2011-04-14

Contents

1 About this manual 5

1.1 What do the symbols and formats mean? 5
1.2 How is this manual structured? 6

2 Safety instructions 7

2.1 Important! 7
2.2 What previous knowledge is required? 8

3 System description 9

4 Configure interfaces 10

4.1 Setting up the interface on the PC 10
4.2 Finding out the valid COM port 10
4.3 Installing and configuring the CANview driver 11

5 Menus 13

5.1 Menu [File] 13
5.1.1 Window [File System] 15
5.2 Menu [Interface] 17
5.2.1 Window [RS232] 18
5.2.2 Window [USB] 19
5.2.3 Window [CAN] 20
5.3 Menu [Protocol] 22
5.4 Menu [Security] 23
5.5 Menu [Options] 24
5.5.1 Window [File Options] 25
5.5.2 Window [Memory Settings] 27
5.5.3 Window [Password] 29
5.6 Menu [Specials] 30
5.6.1 Extended menu [Commands] 32
5.6.2 Window [Memory] 34
5.7 Menu [?] 35
5.8 Buttons 36

6 Programming command lines 37

6.1 Command line arguments of DOWNLOAD.EXE 38
6.2 Executable commands (only in combination with /DEBUG) 39
6.3 Example of a command list 40

7 Using the downloader (examples) 41

7.1 Updating the operating system in the device....41
7.2 Writing the application program to the device....42
7.3 Reading the application program from the device....43
7.4 Creating and transferring the data table to the device....44

8 Glossary of Terms 47

9 Index 63

1 About this manual

Contents

What do the symbols and formats mean? 5
How is this manual structured? 6

202

In the additional "Programming Manual for CoDeSys V2.3" you will obtain more details about the use of the programming system "CoDeSys for Automation Alliance". This manual can be downloaded free of charge from ifm's website:
a) → www.ifm.com > select your country > [Service] > [Download] > [Control systems]
b) → ecomatmobile DVD "Software, tools and documentation"

Nobody is perfect. Send us your suggestions for improvements to this manual and you will receive a little gift from us to thank you.

© All rights reserved by ifm electronic gmbh. No part of this manual may be reproduced and used without the consent of ifm electronic gmbh.

All product names, pictures, companies or other brands used on our pages are the property of the respective rights owners.

1.1 What do the symbols and formats mean?

203

The following symbols or pictograms depict different kinds of remarks in our manuals:

⚠ DANGER

Death or serious irreversible injuries are to be expected.

⚠ WARNING

Death or serious irreversible injuries are possible.

⚠ CAUTION

Slight reversible injuries are possible.

NOTICE

Property damage is to be expected or possible.

NOTE

Important notes on faults and errors.

Info

Further hints.

► ...	Required action
> ...	Response, effect
→ ...	"see"
abc	Cross references (links)
[...]	Designations of keys, buttons or display

1.2 How is this manual structured?

204

This documentation is a combination of different types of manuals. It is for beginners and also a reference for advanced users.

How to use this documentation:

- Refer to the table of contents to select a specific subject.
- The print version of the manual contains a search index in the annex.
- At the beginning of a chapter we will give you a brief overview of its contents.
- Abbreviations and technical terms are listed in the glossary.

In case of malfunctions or uncertainties please contact the manufacturer at:
→ www.ifm.com > select your country > [Contact].

We want to become even better! Each separate section has an identification number in the top right corner. If you want to inform us about any inconsistencies, please indicate this number with the title and the language of this documentation. Thank you for your support.

We reserve the right to make alterations which can result in a change of contents of the documentation. You can find the current version on ifm's website at:
→ www.ifm.com > select your country > [Service] > [Download] > [Control systems]

2 Safety instructions

Contents

Important! 7
What previous knowledge is required? 8

213

2.1 Important!

6091

No characteristics are warranted with the information, notes and examples provided in this manual. The drawings, representations and examples imply no responsibility for the system and no application-specific particularities.

The manufacturer of the machine/equipment is responsible for ensuring the safety of the machine/equipment.

⚠ WARNING

Property damage or bodily injury possible when the notes in this manual are not adhered to!
ifm electronic gmbh does not assume any liability in this regard.

- ► The acting person must have read and understood the safety instructions and the corresponding chapters in this manual before working on and with this device.
- ► The acting person must be authorised to work on the machine/equipment.
- ► Adhere to the technical data of the devices!
  You can find the current data sheet on **ifm's** homepage at:
  → www.ifm.com > select your country > [Data sheet search] > (article no.) > [Technical data in PDF format]
- ► Note the installation and wiring information as well as the functions and features of the devices!
  → supplied installation instructions or on **ifm's** homepage:
  → www.ifm.com > select your country > [Data sheet search] > (article no.) > [Operating instructions]

Start-up behaviour of the controller

The manufacturer of the machine/equipment must ensure with his application program that no dangerous movements can be triggered when the controller starts or restarts.

A restart can, for example, be caused by:

- voltage restoration after power failure
- reset after watchdog response because of too long a cycle time

2.2 What previous knowledge is required?

215

This document is intended for people with knowledge of control technology and PLC programming with IEC 61131-3.

If this device contains a PLC, in addition these persons should know the CoDeSys® software.

The document is intended for specialists. These specialists are people who are qualified by their training and their experience to see risks and to avoid possible hazards that may be caused during operation or maintenance of a product. The document contains information about the correct handling of the product.

Read this document before use to familiarise yourself with operating conditions, installation and operation. Keep the document during the entire duration of use of the device.

Adhere to the safety instructions.

3 System description

6122

This document describes the software tool "Downloader 32" of **ifm electronic gmbh**.

The downloader has the following functions:

- Writing the bootloader, operating system (runtime system), application program or data lists and message texts to the controller or I/O modules (only **ifm** devices).
  Defined formats:
  - Bootloader: `H86 file`
  - Operating system: `H86 file`
  - Application program: `H86 file` or `PRG file`
  - Data / message text lists: `H86 file` or `CLV file`
- Reading the application program (`H86 file`) from a controller and writing it as a 1:1 copy to other controllers (for series production). For safety applications this is the only permitted method to duplicate certified programs.
- Setting and checking the CRC checksum (for safety applications).
- Writing and reading parameter files (e.g. texts or values in `CSV files`).
- Indicating the current runtime system and application program.
- Indicating the current memory allocation (hex dump).
- Starting and stopping the application program in the device.
- The tool can be processed in a batch. This enables to create order lists as a `TXT file` which successively process several downloader commands with just 1 mouse click.
  → Programming command lines (→ page 37)
- Supported interfaces:
  - serial interface RS232
  - USB interface
  - CAN interface (only CAN adapter with VCI interface possible)

4 Configure interfaces

Contents

Setting up the interface on the PC 10
Finding out the valid COM port 10
Installing and configuring the CANview driver 11

6203

Before the downloader can be used sensibly, the user must have set up the interfaces to be used on the PC and must have correctly configured the drivers.

4.1 Setting up the interface on the PC

6148

If the USB or CANview interface is used on this PC for the first time for ifm devices, set up the interface (administrator rights necessary!):

> The window [New hardware found] opens.

► The window "Can Windows connect to Windows Update?" opens.

► Activate the option filed [No, not this time].
Switch to the nex window with [Next].

► Insert the ecomatmobile DVD "Software, tools and documentation" into the DVD drive of the PC.

► Activate the option field [Install the software automatically].
Switch to the nex window with [Next].

> The installation wizard is searching for the suitable interface driver on the DVD.

> A warning appears that the driver software has not passed the Windows logo test.

► Continue the installation with [Continue anyway].

► The installation wizard is installing the driver.

> The message "Installation is completed" appears.

► Switch to the nex window with [Finish].

The same process is now repeated for the port installation.

> The interface is now ready for operation with the connected device.

4.2 Finding out the valid COM port

6205

► Start the control panel under Windows.

► Select: [System] > [Hardware] > [Device Manager].

► Open the branch [Ports (COM & LPT)].

► The installed driver (e.g. ifm CANview-USB) is followed by the related COM-Port, e.g. COM6.

4.3 Installing and configuring the CANview driver

6159

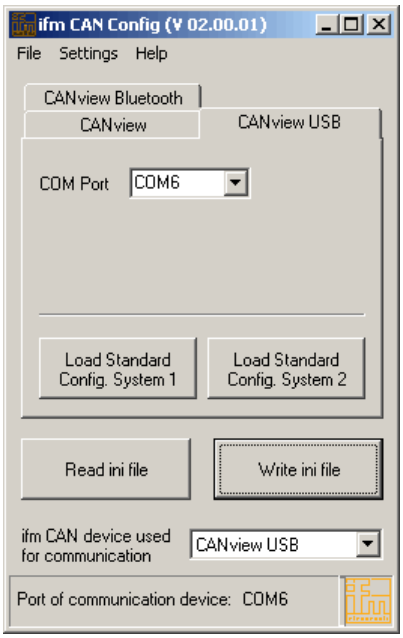
ifm CANview is a VCI emulation of the driver IXXAT.

► To install the CAN modem driver the downloader must not be active.

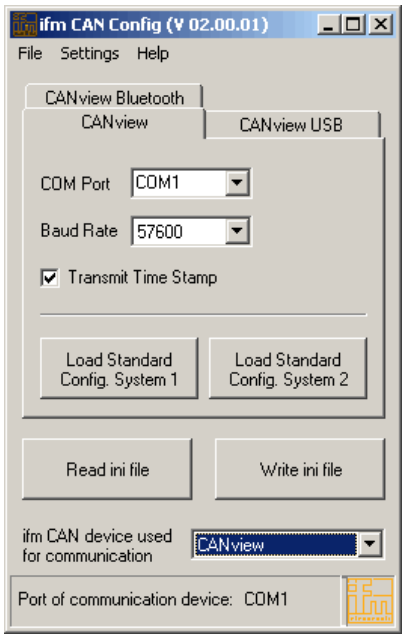
► Connect the CANview modem to the PC.

► Install the CAN modem driver on the PC (→ Operating instructions of the modem).
For ifm CANview insert the ecomatmobile DVD "Software, tools and documentation" into the DVD drive.
NOTE: Administrator rights are necessary for the installation!

► Select the suitable modem driver:



ifm CAN Config for CANview on the USB interface



ifm CAN Config for CANview on the serial interface

► Select the CANview belonging to the modem: EC2071 = USB ⇒ [CANview USB] as follows... a) the respective tab at the top b) as [ifm CAN device used for communication] at the bottom. ► Select the COM port found in the Windows control panel.	► Select the CANview belonging to the modem: EC2070 = serial ⇒ [CANview] as follows... a) the respective tab at the top b) as [ifm CAN device used for communication] at the bottom. ► Select the COM port found in the Windows control panel (usually: COM1). ► Select the baud rate suitable for the device (→ data sheet). ► Activate [Transmit Time Stamp].

> Result → screenshot at the top.

► Save the settings made for the interface with [Write INI File].

► Open the VCI settings of the driver with [Settings] > [VCI] (→ screenshot on the right):

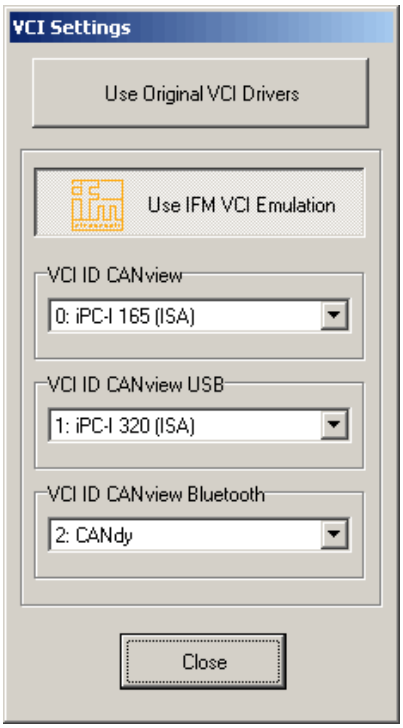
► Select [Use IFM VCI Emulation].

► Depending on the interface: Set the requested driver *) in the field [VCI ID CANview] or [VCI ID CANview USB].

*) A different driver must be indicated for every interface.

*) The [CAN Board Type] declared in the downloader must be the same as the [VCI ID CANview] driver declared in the VCI settings for the respective interface.

► Close the window [VCI-Settings] with [Close].



VCI settings recommended by ifm

► Save the settings for the interface with [Write INI File].

► Load the setting to the CANview module (overwrite it, if necessary) with [Load Standard Config. System 2].

► A progress bar in the status line shows the progress of the data transmission.

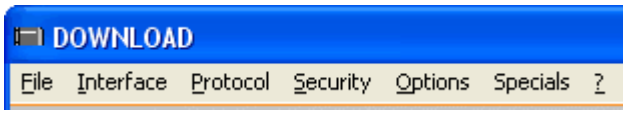
► Close the modem configurator with [File] > [Exit] or with [Alt] + [F4].

5 Menus

Contents

Menu [File] 13
Menu [Interface] 17
Menu [Protocol] 22
Menu [Security] 23
Menu [Options] 24
Menu [Specials] 30
Menu [?] 35
Buttons 36

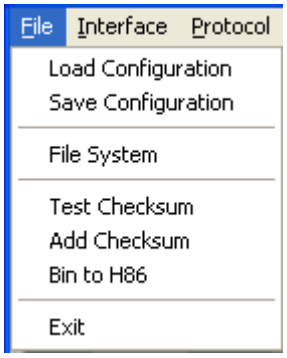
6124



Screenshot: menu bar

5.1 Menu [File]

6126



Screenshot: overview of the menu [File]

The menu opens...
- with a mouse click on [File]
- with the key combination [Alt]+[F]

Menu	Description
[File] > [Load Configuration]	Loads an existing configuration > The Windows Explorer opens the window [Open]. ► Select the drive, directory and `CFG file`. ► Load the selected configuration with [Open]. **NOTE:** When the downloader is started the last saved configuration `Download.cfg` is automatically loaded.

Menu	Description
[File] > [Save Configuration]	Saves the current configuration > The Windows Explorer opens the window [Save as]. ► Select the drive and directory. ► Enter the file name (`.CFG`). ► Save the configuration with [Save]. **NOTE:** When the downloader is exited the current configuration is automatically saved as `Download.cfg` in the directory of `Download.exe`.
[File] > [File System]	(target V05 or higher) Symbol files, for example, can be stored here. Opens the window [File System] (→ page 15).
[File] > [Test Checksum]	Compares the CRC checksum of the application program in the PC with the checksum of the application program in the controller. The checksum must be added to the file with [Add Checksum].
[File] > [Add Checksum]	Adds a checksum to the application program on the hard disk of the PC or updates the available checksum. The controller type defines the length of the checksum (2 or 4 bytes).
[File] > [Bin to H86]	Converts a `BIN file` (application program in the PC) into an `H86 file`. The format `H86` is an ASCII readable representation of a binary file. CoDeSys V2.3 or higher generates `BIN files` on request. To be able to write application programs (as done up to V2.1) to the controller without using a programming system (but with the downloader) the application program must be converted using this command. For the CRnn32 controllers CoDeSys automatically generates a PRG file for [Create boot project] the downloader can use for downloading.
[File] > [Exit]	Exits the downloader. As an alternative: [Alt]+[F4] or mouse click on ☒ or mouse click on [Exit]

5.1.1 Window [File System]

6223

The user saves individual files under predefined names in the device. Under these names the files are available (for CoDeSys and the downloader).

File types: any. The files are application-specific, e.g. symbol file. The symbol file from CoDeSys is saved in the controller. A diagnostic system can load the symbol to obtain the addresses of variables and then read the values of the variables from the controller via 3S communication. The file locations cannot be accessed from the controller.

If the file is larger than the allocated memory area or if this memory area is already full, an error message is provided. For every application download all these files are erased in the file system.

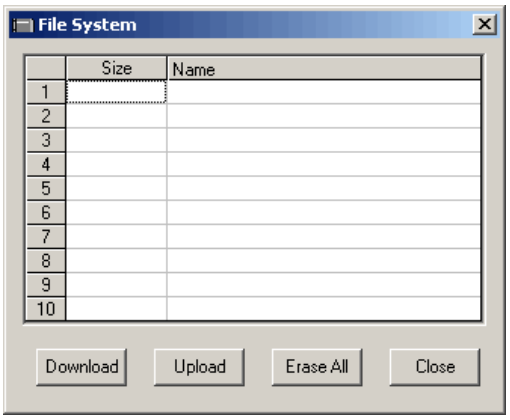
Files cannot be changed.
Individual files cannot be erased.
Only the complete file system can be erased.

► Click [File] > [File System] in the downloader system.

> The window [File System] opens (here: default setting):

> The screenshot shows an empty file system. When files are in the file system, they are indicated.

> New files (even files having the same name) are appended at the end.



The following settings are possible:

Field	Description
Size	For CR0020, CR0200, CR0505, CR7nnn, BU0020, CR9025: size of the file system = 128 kB max. individual file size 64 kB For CC0241, CR03nn, CR1070, CR1071, CR25nn, CS0015: size of the file system = 64 kB max. individual file size 64 kB For CRnn32: size of the file system = 128 kB max. individual file size 128 kB
Name	Everything is converted into lower-case letters. Only ASCII standard characters are allowed (also spaces). 16-bit systems: name length max. 24 characters 32-bit systems: name length max. 256 characters

Button	Description
Download	Write individual files to the device: > The Windows Explorer [Open file] opens. ► Select the directory and file. ► Click [Open] to transfer the selected file to the file system of the device. > The file name including the file size is shown in the window [File System] at the last position. ► Repeat this process with more files, as needed.
Upload	Copy the files in the file system to the PC: > The Windows Explorer [Save as] opens. ► Select the target directory and enter the requested file name. ► With [Save] the downloader copies the file to the hard disk of the PC.
Erase All	Erase all entries in this list. > All files in the device are erased.
Close	Close this window

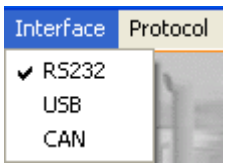
5.2 Menu [Interface]

6127

Activates the connection of the downloader to the device via a specified interface.

► Connect the device to the intended interface.

► If this interface (USB, CAN) is used on this PC for the first time for ifm devices, set up the interface before the next steps are carried out (→ chapter Setting up the interface on the PC (→ page 10)).



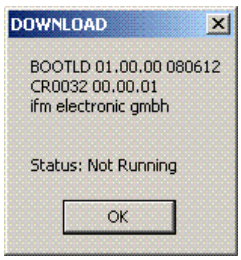
Screenshot: overview of the)menu [Interface]

The menu opens...
- with a mouse click on [Interface]
- with the key combination [Alt]+[I]

Menu	Description
[Interface] > [RS232]	Connection via the serial interface (default setting). Opens the window [RS232] (→ page 18) to configure the serial interface.
[Interface] > [USB]	Connection via the USB interface. Opens the window [USB] (→ page 19) to configure the USB interface.
[Interface] > [CAN]	Connection via the CAN interface. Opens the window [CAN] (→ page 20) to configure the CAN interface.

► After changing the interface click on [Identity].

> The downloader reads the data of the device connected to the current interface and displays it in a new window. Examples:



> Display bootloader version.

> Display device data.

> Display status: PLC program not running: (not yet installed).



> Display version of the operating system.

> Display device data.

> Display status: PLC program running.

> Display cycle time of the PLC program.

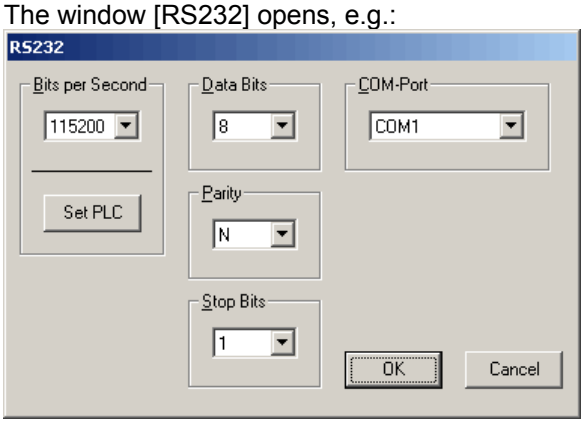
► Use [OK] to close the window.

5.2.1 Window [RS232]

6144

► Click [Interface] > [RS232] in the downloader menu.

> The window [RS232] opens, e.g.:



The following settings are possible:

Field	Description
Bits per second	Baud rate [bits per second]. Possible values = 9600, 19200, 28800, 38400, 57600, 115200.
Data Bits	Number of data bits. Possible values = 7, 8
Parity	Parity: N = no parity E = even parity O = odd parity
Stop Bits	Number of stop bits. Possible values = 1, 2, 50, 100, 125, 250, 500, 800, 1000
COM port	Selection of the COM port on the PC. (Depends on the device and the configuration of the operating system)

► Confirm values with [OK] and close the window.

OR:

► Close the window with [Cancel].

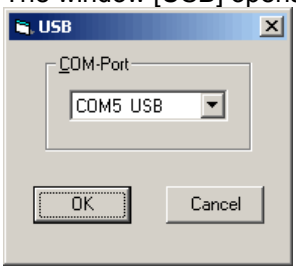
> The values remain unchanged.

5.2.2 Window [USB]

6146

► Click [Interface] > [USB] in the downloader menu.

> The window [USB] opens (here: default setting):



The following settings are possible:

Field	Description
COM port	Selection of the COM port on the PC. (Depends on the device and the configuration of the operating system)

► Confirm values with [OK] and close the window.

OR:

► Close the window with [Cancel].

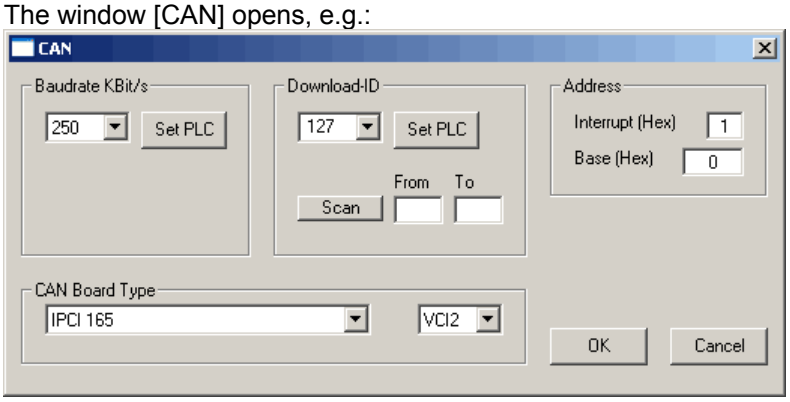
> The values remain unchanged.

5.2.3 Window [CAN]

6149

► Click [Interface] > [CAN] in the downloader menu.

> The window [CAN] opens, e.g.:



The following settings are possible:

Field	Description
Baudrate KBit/s	Number of bits per second. Possible values = 10, 20, 50, 100, 125, 250, 500, 800, 1000 ► Write the selected value to the controller with [Set PLC] *).
Download ID	► Search for a suitable download ID in the area [From]...[To] with [Scan]. The default value for ifm controllers is 127. > Indication of found value (here: 32). OR: ► Overwrite the indicated value with the requested value. ► Write the selected value to the controller with [Set PLC] *).
Address	Interrupt [Hex] = value is assigned when the driver is installed. Do NOT change! Base [Hex] = 0 (do NOT change!)
CAN Board Type	Selection of the connected CAN modem **) → Device description of the manufacturer → Install the CANview driver (→ page 11)

*) The value can only be written to the controller if the communication is ok.

**) The [CAN Board Type] declared in the downloader must be the same as the [VCI ID CANview] driver declared in the VCI settings for the respective interface.

NOTE

The CAN download ID of the device must match the CAN download ID set in CoDeSys!

In the CAN network the CAN download IDs must be unique!

NOTE

The CAN modem driver indicated in the downloader must be identical with the driver specified for the installation.

► Confirm values with [OK] and close the window.

OR:

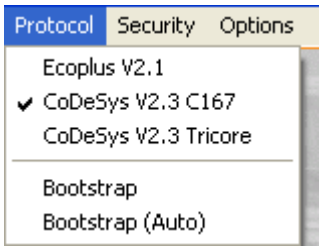
► Close the window with [Cancel].

> The values remain unchanged.

5.3 Menu [Protocol]

6128

The selection defines how (with which protocol) the downloader is to communicate with the device.



Screenshot: overview of the menu [Protocol]

The menu opens...
- with a mouse click on [Protocol]
- with the key combination [Alt]+[I]

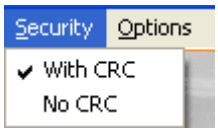
Menu	Description
[Protocol] > [Ecoplus Vx2.1]	Communication via Ecoplus V2.1 This is the predecessor of CoDeSys V2.3.
[Protocol] > [CoDeSys V2.3 C167]	Communication via CoDeSys V2.3 C167 (default setting) This applies to the following devices: - CabinetController: CR030n - ClassicController: CR0020, CR0505 (RTS < V05) - ExtendedController: CR0200 (RTS < V05) - PCB controller: CS0015 - SafetyController: CR7020, CR7200, CR7505 (RTS < V05) - SmartController: CR25nn - PDM360smart: CR1070, CR1071
[Protocol] > [CoDeSys V2.3 Tricore]	Communication via CoDeSys V2.3 Tricore This applies to the following devices: - ClassicController: CR0020, CR0505 (RTS > V05) - ExtendedController: CR0200 (RTS > V05) - Controller CRnn32 - SafetyController: CR7020, CR7200, CR7505 (RTS > V05) - SafetyController: CR7021, CR7201, CR7506
[Protocol] > [Bootstrap] *)	The bootstrap loads the bootloader (firmware) into the device. The loading mode must be forced by the user.
[Protocol] > [Bootstrap (Auto)]	As above, but for the following devices: - ClassicController: CR0020, CR0505 (RTS > V05) - ExtendedController: CR0200 (RTS > V05) - Controller CRnn32 - SafetyController: CR7020, CR7200, CR7505 (RTS > V05) - SafetyController: CR7021, CR7201, CR7506 The loading mode is automatically activated.

*) **NOTE:** The started data transmission must not be interrupted. If the data transmission is interrupted before having been completed, the device can no longer be used und must be returned to the after-sales service for repair.

5.4 Menu [Security]

6129

Security setting whether the data transmission is to be secure or non secure, i.e. with or without generating the CRC checksum.



Screenshot: Overview of the menu [Security]

The menu opens...
- with a mouse click on [Security]
- with the key combination [Alt]+[S]

Menu	Description
[Security] > [With CRC]	Data transmission with generating the checksum (default setting, mandatory for safety applications)
[Security] > [No CRC]	Data transmission without generating the checksum (mainly for files for which generating the checksum is not possible)

5.5 Menu [Options]

6130

Provides detailed setting options.



Screenshot: overview of the menu [Options]

The menu opens...
- with a mouse click on [Options]
- with the key combination [Alt]+[IO]

Menu	Description
[Options] > [File Options]	Opens a new window [File Options] (→ page 25)
[Options] > [Memory Settings]	Opens a new window [Memory Settings] (→ page 27)
[Options] > [Password]	Opens a new window [Password] (→ page 29)
[Options] > [Ask before Download]	Enabled: Before executing the requested download the confirmation prompt must be confirmed with [Yes]. Disabled: The requested download is done immediately without confirmation prompt.
[Options] > [Tool Tips]	Enabled: Shows a brief description of the related function when the mouse cursor hovers over a button. Disabled: Shows no brief description of the related function when the mouse cursor hovers over a button.
[Options] > [Welcome / Good Bye]	Enabled: Shows the version window when starting the downloader and the ifm window when exiting it for 1 second. Disabled: Starts and exits the downloader without these windows.

5.5.1 Window [File Options]

6135

The [File Options] indicate memory areas for the upload or download. The downloader considers these memory areas as the parts of the `H86 file` to be used. Indicate the addresses and sizes of these memory areas in the window [Memory Settings] (→ page 27).

Example: Select 3 memory areas for reading from the device (= upload). During the upload these memory areas are written to 1 common `H86 file`. Before downloading this `H86 file` to another device in the [File Options] select, for example, only 1 area or 2 areas or all 3 to write only these areas to the other device.

NOTE

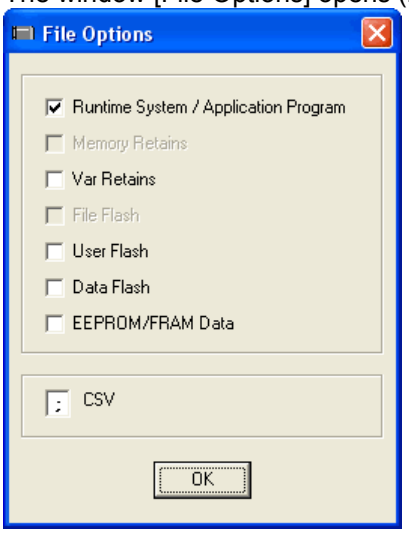
The runtime system must not be combined with other options.

The application program can be combined with one or several other options.

When writing to the PLC (= download) do not write several options together to one file, otherwise problems may occur when the upload is made later on.

► Click [Options] > [File Options] in the downloader menu.

> The window [File Options] opens (here: default setting):



The following settings are possible:

Option	Description
Runtime System / Application Program	Defines the `H86 file` as runtime system or as application program. When writing the downloader automatically recognises the selected file type. For reading only the application program (combined with or without other options) is possible.

Option	Description
Memory Retains	Available for the following devices: - ClassicController: CR0020, CR0505 (RTS > V05) - ExtendedController: CR0200 (RTS > V05) - Controller CRnn32 - SafetyController: CR7020, CR7200, CR7505 (RTS > V05) - SafetyController: CR7021, CR7201, CR7506 - devices with the runtime system V2.1 Memory Retains = Memory protected against power failure, address-oriented Enables the upload or download of the memory retains.
Var Retains	Variables protected against power failure, variable-oriented
File Flash	Available for the following devices: - ClassicController: CR0020, CR0505 (RTS > V05) - ExtendedController: CR0200 (RTS > V05) - Controller CRnn32 - SafetyController: CR7020, CR7200, CR7505 (RTS > V05) - SafetyController: CR7021, CR7201, CR7506 Enables the upload or download of the complete file system.
User Flash	Area for the data of the application program, e.g.: - message text in the PDM - data tables
Data Flash	Area for FLASH_READ and FLASH_WRITE
EEPROM / FRAM Data	Data in the EEPROM / FRAM [1])
CSV	Selection of the separator in the data tables: CSV = ";" (USA): ⇒ CR/LF = ";" ⇒ separation of columns = "," CSV = "," (Europe): ⇒ CR/LF = "," ⇒ separation of columns = ";"

[1]) FRAM indicates here all kinds of non-volatile and fast memories.

5.5.2 Window [Memory Settings]

6136

The menu is used to indicate and set the memory areas in the device. Normally no changes are useful for the customer.

NOTE: Note the memory tables in the system description of the respective device!

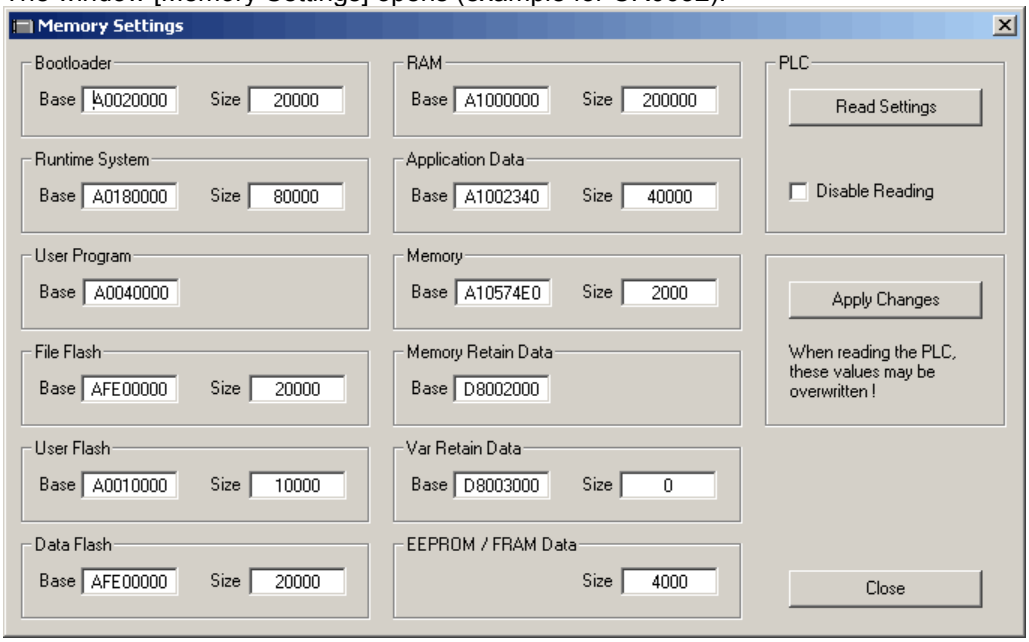
ATTENTION

Risk of data loss and malfunction in case of wrong settings.

► Displayed values are to be changed only with great care and know-how!

► Click [Options] > [Memory Settings] in the downloader menu.

> The window [Memory Settings] opens (example for CR0032):



Base = start address [byte]
Size = memory size [bytes]

The following settings are possible:

Menu	Description
Boot loader	Memory area for the bootloader
Runtime System	Memory area for the runtime system
User Program	Memory area for the application program
File Flash	Available for the following devices: - ClassicController: CR0020, CR0505 (RTS > V05) - ExtendedController: CR0200 (RTS > V05) - Controller CRnn32 - SafetyController: CR7020, CR7200, CR7505 (RTS > V05) - SafetyController: CR7021, CR7201, CR7506 Memory area for the complete file system

Menu	Description
User Flash	Memory area for the data of the application program, e.g.: - message texts in the PDM - data tables - symbol name lists for variables
Data Flash	Memory area for FLASH_READ and FLASH_WRITE
RAM	Memory area for the volatile memory
Application Data	Memory area for the application data
Memory	Memory area for the IEC flag addresses %Mxyz
Memory Retain Data	Available for the following devices: - ClassicController: CR0020, CR0505 (RTS > V05) - ExtendedController: CR0200 (RTS > V05) - Controller CRnn32 - SafetyController: CR7020, CR7200, CR7505 (RTS > V05) - SafetyController: CR7021, CR7201, CR7506 - devices with the runtime system V2.1 Memory Retains = memory area protected against power failure, address-oriented
Var Retain Data	Memory area for variables protected against power failure, variable-oriented
EEPROM / FRAM Data	Memory size for data in the EEPROM or FRAM [1])

Button	Description
Read Settings	Reads the current memory allocation from the device and updates the display
Disable Reading	Enabled = no reading of the memory table from the device possible Disabled = reading the memory table with [Read Settings] from the device possible. The display is updated accordingly.
Apply Changes	Save the changed values
Close	Close this window **INFO:** If changes have not yet been saved, a confirmation prompt appears.

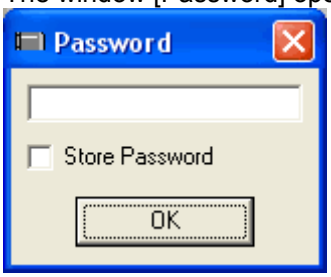
[1]) FRAM indicates here all kinds of non-volatile and fast memories.

5.5.3 Window [Password]

The application program can be protected against unauthorised access with the function SET_PASSWORD. This password must also be indicated in the downloader to be able to read the program from the device and duplicate it.

► Click [Options] > [Password] in the downloader menu.

> The window [Password] opens:



► Enter the password.

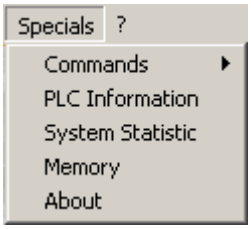
> The application program can be read from the device and duplicated.

► Activate the control field [Store Password] to save the password.

> The password is permanently saved until the next change so that it does not have to be reentered each time the program is accessed.

5.6 Menu [Specials]

6131

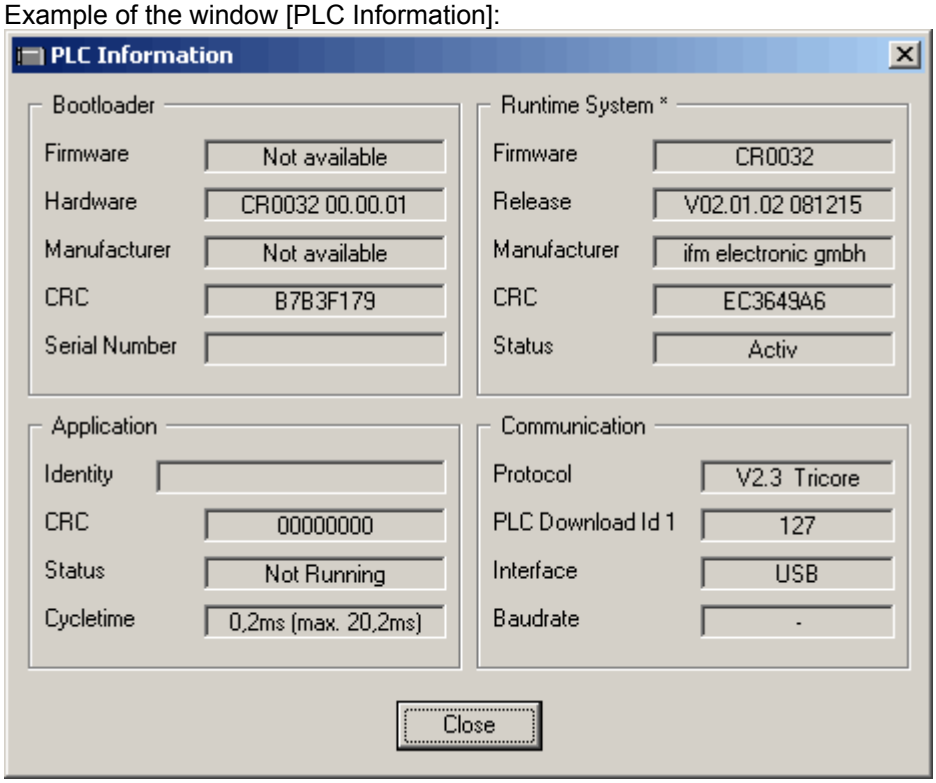


Screenshot: overview of the menu [Specials]

The menu opens...
- with a mouse click on [Specials]

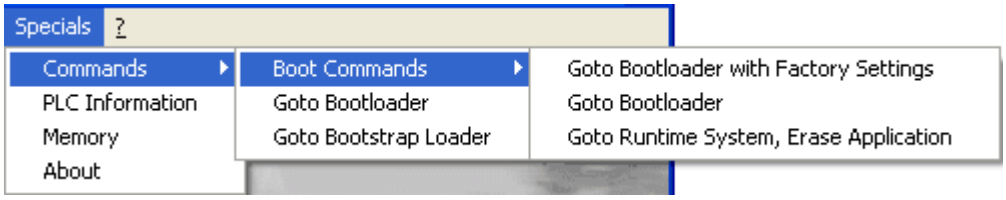
Menu	Description
[Specials] > [Commands]	Adds a level to the menu (→ Extended menu [Commands] (→ page 32)): - Boot Commands - Goto Bootloader - Goto Bootstrap Loader
[Specials] > [PLC Information]	> Opens a new window [PLC Information] showing the current PLC information of the device. (Example → below) ► Close the window with [Close].
[Specials] > [System Statistic]	Only for the following devices: Controller CRnn32 > Reads the statistics data from the device. The generated file `Statistic.LG` contains the contents of the error memory of the controller. ► If the file `Statistic.LG` already exists in the downloader directory, click [Yes] to overwrite the file.
[Specials] > [Memory]	> Opens a new window [Memory] (→ page 34) with the current memory map of the device. ► Close the window with [Close].
[Specials] > [About]	> Opens a new window with the program information of the downloader. ► Close the window with [Close].

Example of the window [PLC Information]:



5.6.1 Extended menu [Commands]

6193



[Specials] > [Commands] > [Boot Commands]

> More submenus possible:
- Goto Bootloader with Factory Settings
- Goto Bootloader
- Goto Runtime System, Erase Application

[Specials] > [Commands] > [Boot Commands] > [Goto Bootloader with Factory Settings]

> You are asked to switch the device off and on again.

► Power the device off and on again.

> If the downloader does not find the device with the current settings, it browses all defined interfaces and protocols.

> The operating system and the application are erased in the device.

> The factory settings are set in the device.

[Specials] > [Commands] > [Boot Commands] > [Goto Bootloader]

> You are asked to switch the device off and on again.

► Power the device off and on again.

> If the downloader does not find the device with the current settings, it browses all defined interfaces and protocols.

> The operating system and the application are erased in the device.

> The settings in the device remain unchanged.

[Specials] > [Commands] > [Boot Commands] > [Goto Runtime System, Erase Application]

> You are asked to switch the device off and on again.

► Power the device off and on again.

> If the downloader does not find the device with the current settings, it browses all defined interfaces and protocols.

> The application is erased in the device.

[Specials] > [Commands] > [Goto Bootloader]

IMPORTANT: The downloader takes the settings for the interface and protocol into account. If the downloader does not find the device, an error message appears and the command is not executed.

> The operating system and the application are erased in the device.

> The settings in the device remain unchanged.

[Specials] > [Commands] > [Goto Runtime System]

(Only for the following devices: Controller CRnn32)

> You are asked to switch the device off and on again.

► Power the device off and on again.

> If the downloader does not find the device with the current settings, it browses all defined interfaces and protocols.

> The application is erased in the device.

[Specials] > [Commands] > [Goto Bootstrap Loader]

(Not for the following devices: Controller CRnn32)

NOTE: After this command the device can no longer be programmed by the user and must be returned to the ifm after-sales service.

> You are asked to switch the device off and on again.

► Power the device off and on again.

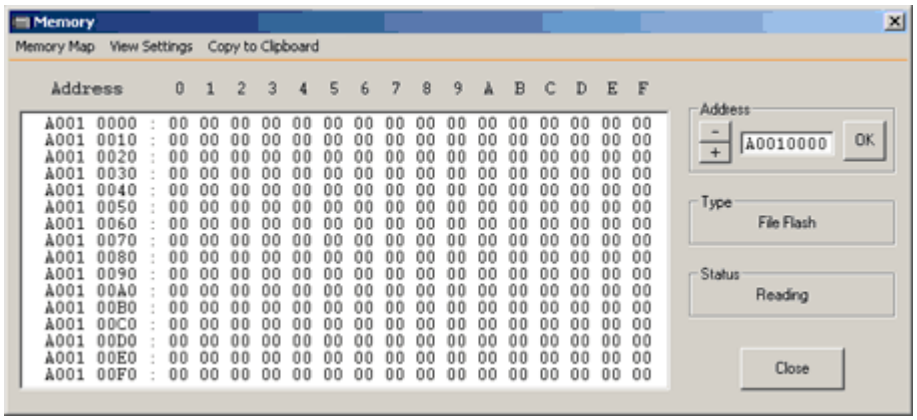
> If the downloader does not find the device with the current settings, it browses all defined interfaces and protocols.

> The bootloader, the operating system and the application are erased in the device.

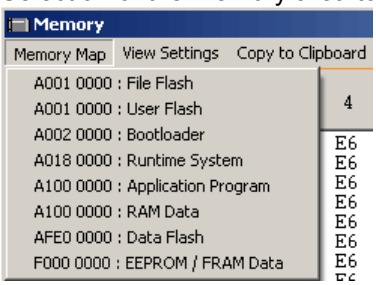
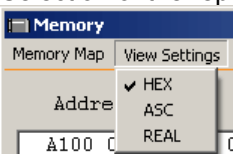
5.6.2 Window [Memory]

6195

The menu [Specials] > [Memory] opens a new window with the memory map of the device, e.g.:



The following settings are possible:

Menu	Description
Memory Map	► Selection of the memory area to be indicated: 
View Settings	► Selection of the representation of values: 
Copy to Clipboard	> Copies the contents of the window to the Windows clipboard ► Copy the contents to an ASCII file (e.g. .TXT). The entries can be edited and modified in the file.

Field	Description
Address	► Use [+] or [-] to scroll 100 bytes up or down in the memory map. ► After a memory address has been entered, confirm this value as the new start address with [OK].
Type	> Display of the current name of the memory area.
Status	> Display whether the memory map is read or written.

► Close the window with [Close].

5.7 Menu [?]

6132

The menu opens...
- with a mouse click on [?]
- with the key combination [Alt]+[⇧]+[?]

► Mouse click on [?] or key combination [Alt]+[⇧]+[?]

> Opens a new window with the program information of the downloader, e.g.:



► Close the window with [OK].

5.8 Buttons

6197



Meaning of the buttons of the downloader:

Button	Description
[Download]	Transfers `H86 files` from the PC to the device: ► Select and mark the `H86 file` in the PC with the Windows Explorer. ► Load the file with [Open]. > The file is written to the device. > Confirm the message "Download successful" with [OK].
[Upload]	Transfers the `H86 files` from the device to the PC: ► Reads the file elements defined under [Options] > [File Options] as one file in the device. ► Select the directory and file name with the Windows Explorer. > Save the `H86 file` in the PC with [Save]. > Confirm the message "Upload successful" with [OK].
[Verify]	Compares the file elements of the device and PC defined under [Options] > [File Options]: ► Select and mark the `H86 file` in the PC with the Windows Explorer. ► Load the file with [Open]. > Confirm the message "Verify successful" with [OK].
[Run]	Start the controller in the device. > The status message (as for [Identity]) appears.
[Stop]	Stop the device in the controller. > The status message (as for [Identity]) appears.
[Identity]	> Checks whether a device is connected to the selected interface and readable with the selected protocol. > Reads the following data in the device: - file options - memory map - device data > The status message appears.
[Exit]	Exits the downloader tool.

6 Programming command lines

Contents

Command line arguments of DOWNLOAD.EXE 38
Executable commands (only in combination with /DEBUG) 39
Example of a command list 40

6199

In batch programs the downloader can process several commands one after the other. The command lines in the command lists must meet the following criteria:

- according to the Windows specification,
- downloader parameters according to the following tables.

a) Enter individual commands in a command line

► Create a link to the downloader on the desktop with the Windows Explorer and name it accordingly.

► Right-click the link and click [Properties] to open the properties dialogue.

► In the line [Target] add the requested commands to the downloader start command, e.g.:
`"C:\...\DOWNLOAD.exe" /run /identity`

> The executable commands are processed in the same order as indicated in the command line, here: start of the PLC program and indication of the status information from the device.

b) Process a list of commands using the downloader

We recommend to write a command list as `TXT file` for every series production giving it a meaningful name. This considerably reduces the risk of wrong settings or data transmissions.

► Create a command list with the Windows Editor and save it, for example, as `CmdLst4711.TXT` in a subdirectory of the downloader, e.g.:
`C:\Programme\ifm electronic\downloader32\Batch`
→ Example of a command list (→ page 40)

► Create a link to the downloader on the desktop with the Windows Explorer and name it accordingly.

► Right-click the link and click [Properties] to open the properties dialogue.

► In the line [Target] add the following command to the downloader start command:
`"C:\...\DOWNLOAD.exe" /cfBatch\CmdLst4711.txt` (without spaces following "/cf"!)

> The downloader processes the commands listed in `CmdLst4711.TXT` one after the other.

> Executable commands are carried out without waiting for user inputs.
Exception 1: Error messages must be acknowledged.
Exception 2: The result of [Identity] must be acknowledged.

6.1 Command line arguments of DOWNLOAD.EXE

6200

One or several arguments can be added to the command line for the downloader to carry out a sensible function. Some of the possible arguments are described in the following table:

Parameters	Description
/CFx	Execute the command file (x = path\name of the command list file)
/LZSVERSIONx	x = number of the protocol (1, 2, 3)
/RS232	Use the RS232 interface (default)
/COMx	x = number of the COM port (1, 2, ...)
/CANx	Use the CAN interface (x = board type, '1' for IPCI 320)
/BAUDRATEx	Examples: x = 9600 for RS232 or x = 125 for CAN
/CANINTx	Interrupt number of the CAN card (default = 5_{hex})
/CANBASEx	Basic address of the CAN card (default = $D000_{hex}$)
/MODULx	Module number of the controller 0...31 (default = 0)
/VCIVERSIONx	x = version of the VCI driver (1 or 2)
/BASEUSERPROGRAMx	Start address of the memory area for the application program (default = 30000_{hex})
/BASEFILEFLASHx	Start address of the memory area for the file system
/FILEFLASHSIZEx	x = size of the memory area for the file system
/BASEUSERFLASHx	Start address of the memory area for the data of the application program
/USERFLASHSIZEx	x = size of the memory area for the data of the application program
/BASEDATAFLASHx	Start address of the memory area for the functions FLASH_READ and FLASH_WRITE
/DATAFLASHSIZEx	x = size of the memory area for the functions FLASH_READ and FLASH_WRITE
/BASERAMx	Start address of the memory area for the volatile memory
/RAMSIZEx	x = size of the memory area for the volatile memory
/BASEDATAx	Start address of the memory area for the application data
/DATASIZEx	x = size of the memory area for the application data
/BASEMEMORYRETAINx	Start address of the memory area protected against power failure, address-oriented
/MEMORYRETAINSIZEx	x = size of the memory area protected against power failure, address-oriented
/BASEVARRETAINx	Start address of the memory area for variables protected against power failure, variable-oriented
/VARRETAINSIZEx	x = size of the memory area for variables protected against power failure, variable-oriented
/BASEMEMORYx	Start address of the memory
/MEMORYSIZEx	x = size of the memory
/EEPROMSIZEx	x = size of the serial EEPROM (default = 2000_{hex})
/BINANFADRx	Start address for BIN files (binary files)
/BINENDADRx	End address for BIN files (binary files)
/BASELZSx	Start address of the runtime system
/LZSSIZEx	Size of the runtime system
/DEBUG	Extended functions accessible (→ Executable commands (→ page 39))

Parameters	Description
/Fx	x = path\name of the file to be loaded
/DLFILE	Upload/download of the operating system or application program (default)
/USERFLASHFILE	Upload/download of the flash data (32K from 70000_{hex})
/FILEFLASHFILE	Upload/download of the flash data
/DATAFLASHFILE	Upload/download of the flash data (16K from $7C000_{hex}$)
/USERRETAINFILE	Upload/download of the retain data
/EEPROMFILE	Upload/download of the data in the serial EEPROM
/NOCRC	Do not verify the checksum of the file
/CRC	Verify the checksum of the file (default)
/AUTOANSWERx	Automatically answer dialogues ((x = YES, NO, OK, IGNORE, CANCEL, RETRY)
/WRITECFG	Save the configuration at the end of the program (default)
/FCFGx	Load CFG file
/NOVISIBLE	No display on screen (exception for error messages)

6.2 Executable commands (only in combination with /DEBUG)

6201

Parameters	Description
/DOWNLOAD	Carry out the download
/UPLOAD	Carry out the upload
/VERIFY	Make the comparison
/RUN	Start the application program
/STOP	Stop the application program
/IDENTITY	Read the identifier
/SETCANBAUDRATEx	Change the CAN baud rate of the controller (x = baud rate)
/SETRS232BAUDRATEx	Change the RS232 baud rate of the controller (x = baud rate)
/DELAYx	Delay x in [ms]
/MESSAGEx	Provide the message x
/EXIT	Abort processing

6.3 Example of a command list

6202

The following command list is as universal as possible:

- By uncommenting a line (i.e. writing "rem" before the line) the line becomes ineffective.
- By removing "rem" before the individual line the line becomes effective.
- Lines with preceding "REM" permanently remain as a comment.

```
REM work with CAN third point from CAN Board type list = ThinCAN (IXXAT)
rem CAN3
REM Set baudrate to 125kB
rem baudrate125
REM download ID = 32
rem modul32

REM work with RS232
rs232
REM Set port to COM 1
com1
REM Set baudrate to 57600
baudrate57600

REM upload userprogram
rem fbatch\test_prg.H86
rem dlfile
rem upload

REM load userprogram
rem fbatch\test_prg.H86
rem dlfile
rem download

REM load CSV list to FRAM
rem fbatch\texte2.csv
rem eepromfile
rem NOCRC
rem download

REM load CSV list to Autosave
fbatch\Flash_texte_test3_safety.csv
REM USERRETAINFILE
DATAFLASHFILE
NOCRC
download

Rem start PLC
run

REM get identity
identity
```

Using the downloader this command list processes the following tasks one after the other:

- Communication with the device via the serial interface on COM1 with 57,600 Bd.
- Load the file Flash_texte_test3_safety.csv from the directory fbatch, define it as DATAFLASHFILE and write it to the device without generating the checksum.
- Start the PLC in the device.
- Read and indicate the status information of the device.

7 Using the downloader (examples)

Contents

Updating the operating system in the device 41
Writing the application program to the device 42
Reading the application program from the device 43
Creating and transferring the data table to the device 44

6207

In this chapter we will describe some actions for which you can use the downloader.

7.1 Updating the operating system in the device

6214

ifm devices are normally supplied without an operating system. The user can immediately load his chosen operating system to the device:

Requirements:
- Device is connected to the PC.
- Interface is selected and correctly set.

► Use [Identity] to check the connection to the device.

► If an error message is displayed:
check and correct connection and settings.

► Click the button [Download].

> The Windows Explorer window opens searching for an `H86 file`.

► Select the directory with the requested operating system, e.g.:
`C:\Programme\ifm electronic\CoDeSys V2.3\Targets\ifm\Library\ifm_CR0032`.

► Select the file, e.g.: `ifm_cr0032_v020102.h86`.

► Start the download to the device with [Open].

> If set:
The confirmation prompt whether to start the download appears. Confirm with [Yes].

> The `H86 file` is written to the device.

> The message "Download successful" appears.

► Confirm the message and close the window with [OK].

► The operating system is saved in the device.

► Read and indicate the status of the device with [Identify].

► Confirm the message and close the window with [OK].

> Exit the downloader with [Exit].

7.2 Writing the application program to the device

6215

Requirement: An operating system is saved in the device.

► Generate a boot project from the application program with CoDeSys:
- Open the application program.
- Generate the following files with [Online] > [Create boot project]:
`Project.PRG`, `Project.HEX`, `Project.CHK`.

The user can write the application program to the device using the downloader without having installed a programming system like CoDeSys:

Requirements:
- Device is connected to the PC.
- Interface is selected and correctly set.

► Use [Identity] to check the connection to the device.

► If an error message is displayed:
check and correct connection and settings.

► Click the button [Download].

> The Windows Explorer window opens searching for an `H86 file`.

► Select the directory with the requested application.

► Select the `H86 file` or `PRG file`.

► Start the download to the device with [Open].

> If set:
The confirmation prompt whether to start the download appears. Confirm with [Yes].

> The downloader generates an `H86 file` of the same name in the directory of the application. A file of the same name which may already exist is overwritten without asking.

> The `H86 file` is written to the device.

> The message "Download successful" appears.

► Confirm the message and close the window with [OK].

> The application program is saved in the device.

► Start the application program in the device with [Run].

► Read and indicate the status of the device with [Identify].

► Confirm the message and close the window with [OK].

► Exit the downloader with [Exit].

7.3 Reading the application program from the device

6216

For safety applications the application program must be certified by a certification body. To do so, the certification body receives a copy of the `H86 file` which was read (together with the checksum) from the tested device. When the body has certified the application program this `H86 file` can then be copied to a number of similar devices using the downloader. This is the only method ensuring that the checksum is not changed, which is a requirement for a certification of safety applications.

How is the application program read from the device?

Requirements:
- Device is connected to the PC.
- Interface is selected and correctly set.

► Use [Identity] to check the connection to the device.

► If an error message is displayed:
check and correct connection and settings.

► Select the menu [Security] > [With CRC].

► Select the menu [Options] > [File Options].

► Only activate the line [Runtime System / Application Program].
Deactivate all other file options.

► Save and close the setting of [File Options] with [OK].

► Click the button [Upload].

> The Windows Explorer window [Save as] opens searching for an `H86 file`.

► Select the directory with the requested application.

► Indicate the requested file name.

► Start the upload from the device with [Save].

> If the file already exists:
The confirmation prompt whether to start the upload appears. Confirm with [Yes].

> The data is read from the device and saved on the hard disk of the PC.

> The message "Upload successful" appears.

► Confirm the message and close the window with [OK].

> The application program is saved in the PC.

► Compare the application program in the device with the `H86 file` saved on the PC with [Verify].

> The message "Verify successful" appears.

► Confirm the message and close the window with [OK].

► Exit the downloader with [Exit].

► For safety applications:
Submit the saved `H86 file` for certification to the certification body.

7.4 Creating and transferring the data table to the device

6228

For the ifm PDM devices message text can be created in several languages in a spreadsheet program, e.g.:

	A	B	C	D
1	**Text to FLASH CR107x**			
2				
3				
4				
5				
6	Start_Adr	Text	typ	input text
7	ifm data at FC000			
8	0	Text 1.....	string	Text 1
9	21	Flashtext2.....	string	Flashtext2
10	42	Flashtext3.....	string	Flashtext3
11	63	Flashtext4.....	string	Flashtext4
12	84	Flashtext5.....	string	Flashtext5
13	105	Flashtext6.....	string	Flashtext6
14	126	Flashtext7.....	string	Flashtext7
15	147	reserve8.....	string	reserve8
16	168	reserve9.....	string	reserve9
17	189	reserve10.....	string	reserve10
18	210	reserve11.....	string	reserve11
19	231	reserve12.....	string	reserve12
20	252	reserve13.....	string	reserve13
21	273	reserve14.....	string	reserve14

In this example the user enters the raw text in column D. The spreadsheet program adopts this text for column B with the text length being automatically adapted to the preset 20 characters.

Important for column A:

- The entry "ifm data at ..." (here in line 7):
  The value specified here ($FC00_{hex}$) is the **memory start address** in the device and thus the offset address for the individual text lines or the offset address for the first language set.
  IMPORTANT: This line MUST be directly followed by the first data line! Headlines which may be required for generating the list (here: line 6) must be above the line with the memory start address!
  IMPORTANT: The memory start address depends on the device and must be indicated correctly!
  (→ Window [Memory Settings] (→ page 27))
- The numerical values below: 0, 21, 42, ...:
  These are the relative start values for the individual text lines. The absolute address results from: start address ($FC00_{hex}$) plus relative start value. The relative start value of a line results from the data length of the current line plus the relative start value of the preceding line (→ Note on column C).
- Create the address areas in ONE continuous block.
  For EEPROMs the memory area reserved for the operating system (e.g. 300_{hex}...$3FF_{hex}$) must be written to. When saving in the EEPROM this area is protected and is not overwritten. (→ Window [Memory Settings] (→ page 27))

► Enter the data to be transmitted (values or text) in column B.

► Enter the respective file type in column C.
IMPORTANT: A STRING type data element always ends with the value "00_{hex}". Therefore the addresses in the above example are indicated in steps of 21 bytes (20 useful bytes plus 1 termination byte).

► Save this table as `CSV file` with a separator:
 - [;] as separator for columns.
 - [CR] as separator for lines.

> During transfer to the device the downloader generates an `H86 file` with the contents of the columns A...C and copies it to the memory area of the device defined in the window [File Options] (→ page 25).

Advantages of this method:
 - much less memory space required,
 - easy access to certain text numbers,
 - easy switching to other language sets in the application program by changing the offset address accordingly.

8 Glossary of Terms

A

Address

This is the "name" of the bus participant. All participants need a unique address so that the signals can be exchanged without problem.

Application software

Software specific to the application, implemented by the machine manufacturer, generally containing logic sequences, limits and expressions that control the appropriate inputs, outputs, calculations and decisions

Necessary to meet the specific (→SRP/CS) requirements.
→ Programming language, safety-related

Architecture

Specific configuration of hardware and software elements in a system.

B

Baud

Baud, abbrev.: Bd = unit for the data transmission speed. Do not confuse baud with "bits per second" (bps, bits/s). Baud indicates the number of changes of state (steps, cycles) per second over a transmission length. But it is not defined how many bits per step are transmitted. The name baud can be traced back to the French inventor J. M. Baudot whose code was used for telex machines.

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

Bus

Serial data transmission of several participants on the same cable.

C

CAN

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

CAN is a priority controlled fieldbus system for larger data volumes. It is available in different variants, e.g. "CANopen" or "CAN in Automation" (CiA).

CAN stack

CAN stack = stack of tasks for CAN data communication.

Category (CAT)

Classification of the safety-related parts of a control system in respect of their resistance to faults and their subsequent behaviour in the fault condition. This safety is achieved by the structural arrangement of the parts, fault detection and/or by their reliability.
(→ EN 954).

CCF

Common **C**ause **F**ailure
Failures of different items, resulting from a common event, where these failures are not consequences of each other.

CiA

CiA = CAN in Automation e.V.

User and manufacturer organisation in Germany / Erlangen. Definition and control body for CAN and CAN-based network protocols.

Homepage → http://www.can-cia.org

CiA DS 304

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

CAN device profile CANopen safety for safety-related communication.

CiA DS 401

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

CAN device profile for digital and analogue I/O modules

CiA DS 402

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

CAN device profile for drives

CiA DS 403

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

CAN device profile for HMI

CiA DS 404

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

CAN device profile for measurement and control technology

CiA DS 405

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

Specification for interface to programmable controllers (IEC 61131-3)

CiA DS 406

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

CAN device profile for encoders

CiA DS 407

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

CAN application profile for local public transport

Clamp 15

In vehicles clamp 15 is the plus cable switched by the ignition lock.

COB-ID

COB = **C**ommunication **Ob**ject
ID = **Id**entifier

Via the COB-ID the participants distinguish the different messages to be exchanged.

CoDeSys

CoDeSys® is a registered trademark of 3S – Smart Software Solutions GmbH, Germany.

"CoDeSys for Automation Alliance" associates companies of the automation industry whose hardware devices are all programmed with the widely used IEC 61131-3 development tool CoDeSys®.

Homepage → http://www.3s-software.com

CRC

CRC = **C**yclic **R**edundancy **C**heck

CRC is a method of information technology to determine a test value for data, to detect faults during the transmission or duplication of data.

Prior to the transmission of a block of data, a CRC value is calculated. After the end of the transaction the CRC value is calculated again at the target location. Then, these two test values are compared.

Cycle time

This is the time for a cycle. The PLC program performs one complete run.

Depending on event-controlled branchings in the program this can take longer or shorter.

D

DC

Direct **C**urrent

DC

Diagnostic **C**overage
Diagnostic coverage is the measure of the effectiveness of diagnostics as the ratio between the failure rate of detected dangerous failures and the failure rate of total dangerous failures:

Formula: DC = failure rate detected dangerous failures / total dangerous failures

Designation	Range
none	DC < 60 %
low	60 % < DC < 90 %
medium	90 % < DC < 99 %
high	99 % < DC

Table: Diagnostic coverage DC

An accuracy of 5 % is assumed for the limit values shown in the table.

Diagnostic coverage can be determined for the whole safety-related system or for only parts of the safety-related system.

Demand rate rd

The demand rate r_d is the frequency of demands to a safety-related reaction of an SRP/CS per time unit.

Diagnosis

During the diagnosis, the "state of health" of the device is checked. It is to be found out if and what faults are given in the device.

Depending on the device, the inputs and outputs can also be monitored for their correct function.
- wire break,
- short circuit,
- value outside range.

For diagnosis, configuration and log data can be used, created during the "normal" operation of the device.

The correct start of the system components is monitored during the initialisation and start phase. Errors are recorded in the log file.

For further diagnosis, self-tests can also be carried out.

Diagnostic coverage

Diagnostic **C**overage
Diagnostic coverage is the measure of the effectiveness of diagnostics as the ratio between the failure rate of detected dangerous failures and the failure rate of total dangerous failures:

Formula: DC = failure rate detected dangerous failures / total dangerous failures

Designation	Range
none	DC < 60 %
low	60 % < DC < 90 %
medium	90 % < DC < 99 %
high	99 % < DC

Table: Diagnostic coverage DC

An accuracy of 5 % is assumed for the limit values shown in the table.

Diagnostic coverage can be determined for the whole safety-related system or for only parts of the safety-related system.

Dither

Dither is a component of the PWM signals to control hydraulic valves. It has shown for electromagnetic drives of hydraulic valves that it is much easier for controlling the valves if the control signal (PWM pulse) is superimposed by a certain frequency of the PWM frequency. This dither frequency must be an integer part of the PWM frequency.

Diversity

In technology diversity is a strategy to increase failure safety.

The systems are designed redundantly, however different implementations are used intentionally and not any individual systems of the same design. It is assumed that systems of the same performance, however of different implementation, are sensitive or insensitive to different interference and will therefore not fail simultaneously.

The actual implementation may vary according to the application and the requested safety:

- use of components of several manufacturers,
- use of different protocols to control devices,
- use of totally different technologies, for example an electrical and a pneumatic controller,
- use of different measuring methods (current, voltage),

- two channels with reverse value progression:
  channel A: 0...100 %
  channel B: 100...0 %

DRAM

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

Technology for an electronic memory module with random access (Random Access Memory, RAM). The memory element is a capacitor which is either charged or discharged. It becomes accessible via a switching transistor and is either read or overwritten with new contents. The memory contents are volatile: the stored information is lost in case of lacking operating voltage or too late restart.

DTC

DTC = **D**iagnostic **T**rouble **C**ode = error code
Faults and errors well be managed and reported via assigned numbers – the DTCs.

E

ECU

(1) **E**lectronic **C**ontrol **U**nit = control unit or microcontroller
(2) **E**ngine **C**ontrol **U**nit = control device of a motor

EDS-file

EDS = **E**lectronic **D**ata **S**heet, e.g. for:

- File for the object directory in the master
- CANopen device descriptions

Via EDS devices and programs can exchange their specifications and consider them in a simplified way.

Embedded software

System software, basic program in the device, virtually the operating system.

The firmware establishes the connection between the hardware of the device and the user software. This software is provided by the manufacturer of the controller as a part of the system and cannot be changed by the user.

EMCY

abbreviation for emergency

EMV

EMC = **E**lectro **M**agnetic **C**ompatibility

According to the EC directive (2004/108/EEC) concerning electromagnetic compatibility (in short EMC directive) requirements are made for electrical and electronic apparatus, equipment, systems or components to operate satisfactorily in the existing electromagnetic environment. The devices must not interfere with their environment and must not be adversely influenced by external electromagnetic interference.

Ethernet

Ethernet is a widely used, manufacturer-independent technology which enables data transmission in the network at a speed of 10 or 100 million bits per second (Mbps). Ethernet belongs to the family of so-called "optimum data transmission" on a non exclusive transmission medium. The concept was developed in 1972 and specified as IEEE 802.3 in 1985.

EUC

EUC = "Equipment Under Control"

EUC is equipment, machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities (→ IEC 61508-4, section 3.2.3). Therefore, the EUC is the set of all equipment, machinery, apparatus or plant that gives rise to hazards for which the safety-related system is required.

If any reasonably foreseeable action or inaction leads to hazards with an intolerable risk arising from the EUC, then safety functions are necessary to achieve or maintain a safe state for the EUC. These safety functions are performed by one or more safety-related systems.

F

Failure

Failure is the termination of the ability of an item to perform a required function.

After a failure, the item has a fault. Failure is an event, fault is a state.

The concept as defined does not apply to items consisting of software only.

Failure, dangerous

A dangerous failure has the potential to put the SRP/SC in a hazardous or fail-to-function state. Whether or not the potential is realized can depend on the channel architecture of the system; in redundant systems a dangerous hardware failure is less likely to lead to the overall dangerous or fail-to-function state.

Failure, systematic

A systematic failure is a failure related in a deterministic way (not coincidental) to a certain cause. The systematic failure can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors.

Corrective maintenance without modification of the system will usually not eliminate the failure cause.

Fault

A fault is the state of an item characterized by the inability to perform the requested function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources.

A fault is often the result of a failure of the item itself, but may exist without prior failure.

In ISO 13849-1 "fault" means "random fault".

Fault tolerance time

The max. time it may take between the occurrence of a fault and the establishment of the safe state in the application without having to assume a danger for people.

The max. cycle time of the application program (in the worst case 100 ms) and the possible delay and response times due to switching elements have to be considered.

The resulting total time must be smaller than the fault tolerance time of the application.

FiFo

FiFo (**F**irst **I**n, **F**irst **O**ut) = operation of the stack: the data package which was written into a stack at first will be read at first too. For every identifier there is such one buffer (as a queue) available.

Firmware

System software, basic program in the device, virtually the operating system.

The firmware establishes the connection between the hardware of the device and the user software. This software is provided by the manufacturer of the controller as a part of the system and cannot be changed by the user.

First fault occurrence time

Time until the first failure of a safety element.

The operating system verifies the controller by means of the internal monitoring and test routines within a period of max. 30 s.

This "test cycle time" must be smaller than the statistical first fault occurrence time for the application.

Flash memory

Flash ROM (or flash EPROM or flash memory) combines the advantages of semiconductor memory and hard disks. Just like every other semiconductor memory the flash memory does not require moving parts. And the data is maintained after switch-off, similar to a hard disk.

The flash ROM evolved from the EEPROM (**E**lectrical **E**rasable and **P**rogrammable **R**ead-**O**nly **M**emory). The storage function of data in the flash ROM is identical to the EEPROM. Similar to a hard disk, the data are however written and deleted blockwise in data blocks up to 64, 128, 256, 1024, ... bytes at the same time.

Advantages of flash memories

- The stored data are maintained even if there is no supply voltage.
- Due to the absence of moving parts, flash is noiseless and insensitive to shocks and magnetic fields.
- In comparison to hard disks, flash memories have a very short access time. Read and write speed are virtually constant across the entire memory area.
- The memory size that can be obtained has no upper limit, due to the simple and space-saving arrangement of the storage cells.

Disadvantages of flash memories

- A storage cell can tolerate a limited number of write and delete processes:
  - Multi-level cells: typ. 10 000 cycles
  - Single level cells: typ. 100 000 cycles
- Given that a write process writes memory blocks of between 16 and 128 Kbytes at the same time, memory cells which require no change are used as well.

FMEA

FMEA = **F**ailure **M**ode and **E**ffects **A**nalysis

Method of reliability engineering, to find potential weak points. Within the framework of quality or security management, the FMEA is used preventively to prevent faults and increase the technical reliability.

FRAM

FRAM, or also FeRAM, means **Fe**rroelectric **R**andom **A**ccess **M**emory. The storage operation and erasing operation is carried out by a polarisation change in a ferroelectric layer.

Advantages of FRAM as compared to conventional read-only memories:

- non-volatile,
- compatible with common EEPROMs, but:
- access time approx. 100 ns,
- nearly unlimited access cycles possible.

Functional safety

Part of the overall safety referred to the →EUC and the EUC control system which depends on the correct functioning of the electric or electronic safety-related system, safety-related systems of other technologies and external devices for risk reduction.

H

Harm

Physical injury or damage to health.

Hazard

Hazard is the potential source of harm.

A distinction is made between the source of the hazard, e.g.:
- mechanical hazard,
- electrical hazard,

or the nature of the potential harm, e.g.:
- electric shock hazard,
- cutting hazard,
- toxic hazard.

The hazard envisaged in this definition is either permanently present during the intended use of the machine, e.g.:
- motion of hazardous moving elements,
- electric arc during a welding phase,
- unhealthy posture,
- noise emission,
- high temperature,

or the hazard may appear unexpectedly, e.g.:
- explosion,
- crushing hazard as a consequence of an unintended/unexpected start-up,
- ejection as a consequence of a breakage,
- fall as a consequence of acceleration/deceleration.

Heartbeat

The participants regularly send short signals. In this way the other participants can verify if a participant has failed. No master is necessary.

HMI

HMI = **H**uman **M**achine **I**nterface

I

ID

ID = **Id**entifier

Name to differentiate the devices / participants connected to a system or the message packets transmitted between the participants.

IEC user cycle

IEC user cycle = PLC cycle in the CoDeSys application program.

Instructions

Superordinate word for one of the following terms:
installation instructions, data sheet, user information, operating instructions, device manual, installation information, online help, system manual, programming manual, etc.

Intended use

Use of a product in accordance with the information provided in the instructions for use.

IP address

IP = **I**nternet **P**rotocol

The IP address is a number which is necessary to clearly identify an internet participant. For the sake of clarity the number is written in 4 decimal values, e.g. 127.215.205.156.

ISO 11898

Standard: "Road vehicles – Controller area network"

Part 1: "Data link layer and physical signalling"

Part 2: "High-speed medium access unit"

Part 3: "Low-speed, fault-tolerant, medium dependent interface"

Part 4: "Time-triggered communication"

Part 5: "High-speed medium access unit with low-power mode"

ISO 11992

Standard: "Interchange of digital information on electrical connections between towing and towed vehicles"

Part 1: "Physical and data-link layers"

Part 2: "Application layer for brakes and running gear"

Part 3: "Application layer for equipment other than brakes and running gear"

Part 4: "Diagnostics"

ISO 16845

Standard: "Road vehicles – Controller area network (CAN) – Conformance test plan"

L

LED

LED = **L**ight **E**mitting **D**iode

Light emitting diode, also called luminescent diode, an electronic element of high coloured luminosity at small volume with negligible power loss.

Life, mean

Mean **T**ime **T**o **F**ailure (MTTF) or: mean life.

The $MTTF_d$ is the expectation of the mean time to dangerous failure.

Designation	Range
low	3 years < $MTTF_d$ < 10 years
medium	10 years < $MTTF_d$ < 30 years
high	30 years < $MTTF_d$ < 100 years

Table: Mean time of each channel to the dangerous failure $MTTF_d$

Link

A link is a cross-reference to another part in the document or to an external document.

LSB

Least **S**ignificant **B**it/Byte

M

MAC-ID

MAC = **M**anufacturer‘s **A**ddress **C**ode
= manufacturer's serial number

→ID = **Id**entifier

Every network card has a MAC address, a clearly defined worldwide unique numerical code, more or less a kind of serial number. Such a MAC address is a sequence of 6 hexadecimal numbers, e.g. "00-0C-6E-D0-02-3F".

Master

Handles the complete organisation on the bus. The master decides on the bus access time and polls the →slaves cyclically.

Mission time TM

Mission time T_M is the period of time covering the intended use of an SRP/CS.

Misuse

The use of a product in a way not intended by the designer.

The manufacturer of the product has to warn against readily predictable misuse in his user information.

MMI

HMI = **H**uman **M**achine **I**nterface
→ HMI (→ page 52)

Monitoring

Safety function which ensures that a protective measure is initiated:

- if the ability of a component or an element to perform its function is diminished.
- if the process conditions are changed in such a way that the resulting risk increases.

MRAM

MRAM means **M**agnetoresistive **R**andom **A**ccess **M**emory. The information is stored by means of magnetic storage elements. The property of certain materials is used to change their electrical resistance when exposed to magnetic fields.

Advantages of MRAM as compared to conventional RAM memories:

- non volatile (like FRAM), but:
- access time only approx. 35 ns,
- unlimited number of access cycles possible.

MSB

Most **S**ignificant **B**it/Byte

MTBF

Mean **T**ime **B**etween **F**ailures (MTBF)
Is the expected value of the operating time between two consecutive failures of items that are maintained.

NOTE: For items that are NOT maintained the mean life →MTTF is the expected value (mean value) of the distribution of lives.

MTTF

Mean **T**ime **T**o **F**ailure (MTTF) or: mean life.

MTTFd

Mean **T**ime **T**o **F**ailure (MTTF) or: mean life.

The $MTTF_d$ is the expectation of the mean time to dangerous failure.

Designation	Range
low	3 years < $MTTF_d$ < 10 years
medium	10 years < $MTTF_d$ < 30 years
high	30 years < $MTTF_d$ < 100 years

Table: Mean time of each channel to the dangerous failure $MTTF_d$

Muting

Muting is the temporary automatic suspension of a safety function(s) by the SRP/CS.

Example: The safety light curtain is bridged, if the closing tools have reached a finger-proof distance to each other. The operator can now approach the machine without any danger and guide the workpiece.

N

NMT

NMT = **N**etwork **M**anagemen**t** = (here: in the CAN bus)

The NMT master controls the operating states of the NMT slaves.

Node

This means a participant in the network.

Node Guarding

Network participant

Configurable cyclic monitoring of each slave configured accordingly. The master verfies if the slaves reply in time. The slaves verify if the master regularly sends requests. In this way failed network participants can be quickly identified and reported.

O

Obj / object

Term for data / messages which can be exchanged in the CANopen network.

Object directory

Contains all CANopen communication parameters of a device as well as device-specific parameters and data.

OBV

Contains all CANopen communication parameters of a device as well as device-specific parameters and data.

Operating system

Basic program in the device, establishes the connection between the hardware of the device and the user software.

Operational

Operating state of a CANopen participant. In this mode SDOs, NMT commands and PDOs can be transferred.

P

PC card

→PCMCIA card

PCMCIA card

PCMCIA = Personal Computer Memory Card International Association, a standard for expansion cards of mobile computers. Since the introduction of the cardbus standard in 1995 PCMCIA cards have also been called PC card.

PDM

PDM = **P**rocess and **D**ialogue **M**odule

Device for communication of the operator with the machine / plant.

PDO

PDO = **P**rocess **D**ata **O**bject

The time-critical process data is transferred by means of the "process data objects" (PDOs). The PDOs can be freely exchanged between the individual nodes (PDO linking). In addition it is defined whether data exchange is to be event-controlled (asynchronous) or synchronised. Depending on the type of data to be transferred the correct selection of the type of transmission can lead to considerable relief for the CAN bus.

These services are not confirmed by the protocol, i.e. it is not checked whether the message reaches the receiver. Exchange of network variables corresponds to a "1 to n connection" (1 transmitter to n receivers).

PDU

PDU = **P**rotocol **D**ata **U**nit

The PDU is an item of the CAN protocol SAE J1939. PDU indicates a part of the destination or source address.

Performance Level

Performance **L**evel
According to ISO 13849-1, a specification (PL a...e) of safety-related parts of control systems to perform a safety function under foreseeable conditions.

PES

Programmable **E**lectronic **S**ystem
A programmable electronic system is a system ...
- for control, protection or monitoring,
- dependent for its operation on one or more programmable electronic devices,
- including all elements of the system such as input and output devices.

PGN

PGN = **P**arameter **G**roup **N**umber
PGN = PDU format (PF) + PDU source (PS)

The parameter group number is an item of the CAN protocol SAE J1939. PGN collects the address parts PF and PS.

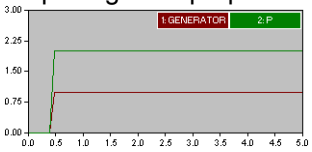
Pictogram

Pictograms are figurative symbols which convey information by a simplified graphic representation.

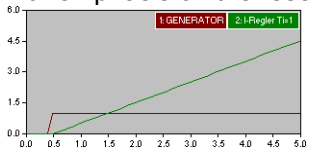
→ Chapter What do the symbols and formats mean? (→ page 5)

PID controller

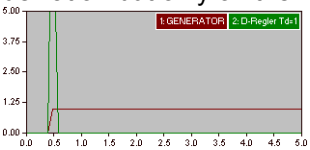
P = proportional part
The P controller exclusive consists of a proportional part of the amplification K_p. The output signal is proportional to the input signal.



I = integral part
An I controller acts to the manipulating variable by phasing integration of the control deviation with emphasis on the reset time T_N.



D = differential part
The D controller doesn't react on the control deviation but only on their speed of change.



PL

Performance **L**evel
According to ISO 13849-1, a specification (PL a...e) of safety-related parts of control systems to perform a safety function under foreseeable conditions.

PLC configuration

Part of the CoDeSys user interface.

► The programmer tells the programming system which hardware is to be programmed.

> CoDeSys loads the corresponding libraries.

> Reading and writing the peripheral states (inputs/outputs) is possible.

PLr

Using the "required performance level" PL_r the risk reduction for each safety function according to ISO 13849 is achieved.

For each selected safety function to be carried out by a SRP/CS, a PL_r shall be determined and documented. The determination of the PL_r is the result of the risk assessment and refers to the amount of the risk reduction.

Pre-Op

Pre-Op = PRE-OPERATIONAL mode

Operating status of a CANopen participant. After application of the supply voltage each participant automatically passes into this state. In the CANopen network only SDOs and NMT commands can be transferred in this mode but no process data.

prepared

Operating status of a CANopen participant. In this mode only NMT commands are transferred.

Process image

Process image is the status of the inputs and outputs the PLC operates with within one cycle.

- At the beginning of the cycle the PLC reads the conditions of all inputs into the process image.
  During the cycle the PLC cannot detect changes to the inputs.
- During the cycle the outputs are only changed virtually (in the process image).
- At the end of the cycle the PLC writes the virtual output states to the real outputs.

Programming language, safety-related

Only the following programming languages shall be used for safety-related applications:

- Limited variability language (LVL) that provides the capability of combining predefined, application-specific library functions.
  In CoDeSys these are LD (ladder diagram) and FBD (function block diagram).
- Full variability language (FVL) provides the capability of implementing a wide variety of functions.
  These include e.g. C, C++, Assembler. In CoDeSys it is ST (structured text).

► Structured text is recommended exclusively in separate, certified functions, usually in embedded software.

► In the "normal" application program only LD and FBD should be used. The following minimum requirements shall be met.

In general the following minimum requirements are made on the safety-related application software (SRASW):

► Modular and clear structure of the program. Consequence: simple testability.

► Functions are represented in a comprehensible manner:
- for the operator on the screen (navigation)
- readability of a subsequent print of the document.

► Use symbolic variables (no IEC addresses).

► Use meaningful variable names and comments.

► Use easy functions (no indirect addressing, no variable fields).

► Defensive programming.

► Easy extension or adaptation of the program possible.

Protective measure

Measure intended to achieve risk reduction, e.g.:
- fault-excluding design,
- safeguarding measures (guards),
- complementary protective measures (user information),
- personal protective equipment (helmet, protective goggles).

PWM

PWM = pulse width modulation

Via PWM a digital output (capability provided by the device) can provide an almost analogue voltage by means of regular fast pulses. The PWM output signal is a pulsed signal between GND and supply voltage.

Within a defined period (PWM frequency) the mark-to-space ratio is varied. Depending on the mark-to-space ratio, the connected load determines the corresponding RMS current.

R

Ratio

Measurements can also be performed ratiometrically. The input signal generates an output signal which is in a defined ratio to the input signal. This means that analogue input signals can be evaluated without additional reference voltage. A fluctuation of the supply voltage has no influence on this measured value.

RAW-CAN

RAW-CAN means the pure CAN protocol which works without an additional communication protocol on the CAN bus (on ISO/OSI layer 2). The CAN protocol is international defined according to ISO 11898-1 and garantees in ISO 16845 the interchangeability of CAN chips in addition.

Redundant

Redundancy is the presence of more than the necessary means so that a function unit performs a requested function or that data can represent information.

Several kinds of redundancy are distinguished:

- Functional redundancy aims at designing safety-related systems in multiple ways in parallel so that in the event of a failure of one component the others ensure the task.
- In addition it is tried to separate redundant systems from each other with regard to space. Thus the risk that they are affected by a common interference is minimised.
- Finally, components from different manufacturers are sometimes used to avoid that a systematic fault causes all redundant systems to fail (diverse redundancy).

The software of redundant systems should differ in the following aspects:

- specification (different teams),
- specification language,
- programming (different teams),
- programming language,
- compiler.

Remanent

Remanent data is protected against data loss in case of power failure.

The operating system for example automatically copies the remanent data to a flash memory as soon as the voltage supply falls below a critical value. If the voltage supply is available again, the operating system loads the remanent data back to the RAM memory.

The data in the RAM memory of a controller, however, is volatile and normally lost in case of power failure.

Reset, manual

4021

The manual reset is an internal function within the SRP/CS used to restore manually one or more safety functions before re-starting a machine.

Residual risk

Risk remaining after protective measures have been taken. The residual risk has to be clearly warned against in operating instructions and on the machine.

Risk

Combination of the probability of occurrence of harm and the severity of that harm.

Risk analysis

Combination of ...

- the specification of the limits of the machine (intended use, time limits),
- hazard identification (intervention of people, operating status of the machine, foreseeable misuse) and
- the risk estimation (degree of injury, extent of damage, frequency and duration of the risk, probability of occurrence, possibility of avoiding the hazard or limiting the harm).

Risk assessment

Overall process comprising risk analysis and risk evaluation.

According to Machinery Directive 2006/42/EU the following applies: "The manufacturer of machinery or his authorised representative must ensure that a risk assessment is carried out in order to determine the health and safety requirements which apply to the machinery. The machinery must then be designed and constructed taking into account the results of the risk assessment." (→ Annex 1, General principles)

Risk evaluation

Judgement, on the basis of the risk analysis, of whether risk reduction objectives have been achieved.

ro

RO = read only for reading only

Unidirectional data transmission: Data can only be read and not changed.

RTC

RTC = **R**eal **T**ime **C**lock

Provides (batter-backed) the current date and time. Frequent use for the storage of error message protocols.

rw

RW = read/ write

Bidirectional data transmission: Data can be read and also changed.

S

SAE J1939

The network protocol SAE J1939 describes the communication on a CAN bus in commercial vehicles for transmission of diagnosis data (e.g.motor speed, temperature) and control information.
→ CiA DS 402

Standard: "Recommended Practice for a Serial Control and Communications Vehicle Network"

Part 2: "Agricultural and Forestry Off-Road Machinery Control and Communication Network"

Part 3: "On Board Diagnostics Implementation Guide"

Part 5: "Marine Stern Drive and Inboard Spark-Ignition Engine On-Board Diagnostics Implementation Guide"

Part 11: "Physical Layer – 250 kBits/s, Shielded Twisted Pair"

Part 13: "Off-Board Diagnostic Connector"

Part 15: "Reduced Physical Layer, 250 kBits/s, Un-Shielded Twisted Pair (UTP)"

Part 21: "Data Link Layer"

Part 31: "Network Layer"

Part 71: "Vehicle Application Layer"

Part 73: "Application Layer – Diagnostics"

Part 81: "Network Management Protocol"

Safety function

Function of the machine whose failure can result in an immediate increase of the risk(s). The designer of such a machine therefore has to:
- safely prevent a failure of the safety function,
- reliably detect a failure of the safety function in time,
- bring the machine into a safe state in time in the event of a failure of the safety function.

Safety-standard types

The safety standards in the field of machines are structured as below:

Type-A standards (basic safety standards) giving basic concepts, principles for design, and general aspects that can be applied to all machinery. Examples: basic terminology, methodology (ISO 12100-1), technical principles (ISO 12100-2), risk assessment (ISO 14121), ...

Type-B standards (generic safety standards) dealing with one safety aspect or one type of safeguard that can be used across a wide range of machinery.

- Type-B1 standards on particular safety aspects. Examples: safety distances (EN 294), hand/arm speeds (EN 999), safety-related parts of control systems (ISO 13849), temperatures, noise, ...

- Type-B2 standards on safeguards. Examples: emergency stop circuits ((ISO 13850), two-hand controls, interlocking devices or electro-sensitive protective equipment (ISO 61496), ...

Type-C standards (machine safety standards) dealing with detailed safety requirements for a particular machine or group of machines.

SCT

In CANopen safety the **S**afeguard **C**ycle **T**ime (SCT) monitors the correct function of the periodic transmission (data refresh) of the SRDOs. The data must have been repeated within the set time to be valid. Otherwise the receiving controller signals a fault and passes into the safe state (= outputs switched off).

SD card

An SD memory card (short for **S**ecure **D**igital Memory Card) is a digital storage medium that operates to the principle of flash storage.

SDO

SDO = **S**ervice **D**ata **O**bject.

SDO is a specification for a manufacturer-dependent data structure for standardised data access. "Clients" ask for the requested data from "servers". The SDOs always consist of 8 bytes. Longer data packages are distributed to several messages.

Examples:

- Automatic configuration of all slaves via SDOs at the system start,
- reading error messages from the object directory.

Every SDO is monitored for a response and repeated if the slave does not respond within the monitoring time.

Self-test

Test program that actively tests components or devices. The program is started by the user and takes a certain time. The result is a test protocol (log file) which shows what was tested and if the result is positive or negative.

SIL

According to IEC 62061 the safety-integrity level SIL is a classification (SIL CL 1...4) of the safety integrity of the safety functions. It is used for the evaluation of electrical/electronic/programmable electronic (E/E/EP) systems with regard to the reliability of safety functions. The safety-related design principles that have to be adhered to so that the risk of a malfunction can be minimised result from the required level.

Slave

Passive participant on the bus, only replies on request of the →master. Slaves have a clearly defined and unique →address in the bus.

SRDO

Safe data is exchanged via SRDOs (**S**afety-**R**elated **D**ata **O**bjects). An SRDO always consists of two CAN messages with different identifiers:

- message 1 contains the original user data,
- message 2 contains the same data which are inverted bit by bit.

SRP/CS

Safety-**R**elated **P**art of a **C**ontrol **S**ystem

Part of a control system that responds to safety-related input signals and generates safety-related output signals. The combined safety-related parts of a control system start at the point where the safety-related input signals are initiated (including, for example, the actuating cam and the roller of the position switch) and end at the output of the power control elements (including, for example, the main contacts of a contactor).

SRVT

The SRVT (**S**afety-**R**elated **O**bject **V**alidation **T**ime) ensures with CANopen safety that the time between the SRDO-message pairs is adhered to.

Only if the redundant, inverted message has been transmitted after the original message within the SRVT set are the transmitted data valid. Otherwise the receiving controller signals a fault and will pass into the safe state (= outputs switched off).

State, safe

The state of a machine is said to be safe when there is no more hazard formed by it. This is usually the case if all possible dangerous movements are switched off and cannot start again unexpectedly.

Symbols

Pictograms are figurative symbols which convey information by a simplified graphic representation.

→ Chapter What do the symbols and formats mean? (→ page 5)

System variable

Variable to which access can be made via IEC address or symbol name from the PLC.

T

Target

The target indicates the target system where the PLC program is to run. The target contains the files (drivers and if available specific help files) required for programming and parameter setting.

TCP

The **T**ransmission **C**ontrol **P**rotocol is part of the TCP/IP protocol family. Each TCP/IP data connection has a transmitter and a receiver. This principle is a connection-oriented data transmission. In the TCP/IP protocol family the TCP as the connection-oriented protocol assumes the task of data protection, data flow control and takes measures in the event of data loss.

(compare: →UDP)

Template

A template can be filled with content.
Here: A structure of pre-configured software elements as basis for an application program.

Test rate rt

The test rate r_t is the frequency of the automatic tests to detect errors in an SRP/CS in time.

U

UDP

UDP (**U**ser **D**atagram **P**rotocol) is a minimal connectionless network protocol which belongs to the transport layer of the internet protocol family. The task of UDP is to ensure that data which is transmitted via the internet is passed to the right application.

At present network variables based on CAN and UDP are implemented. The values of the variables are automatically exchanged on the basis of broadcast messages. In UDP they are implemented as broadcast messages, in CAN as PDOs. These services are not confirmed by the protocol, i.e. it is not checked whether the message is received. Exchange of network variables corresponds to a "1 to n connection" (1 transmitter to n receivers).

Uptime, mean

Mean **T**ime **B**etween **F**ailures (MTBF)
Is the expected value of the operating time between two consecutive failures of items that are maintained.

NOTE: For items that are NOT maintained the mean life →MTTF is the expected value (mean value) of the distribution of lives.

Use, intended

Use of a product in accordance with the information provided in the instructions for use.

W

Watchdog

In general the term watchdog is used for a component of a system which watches the function of other components. If a possible malfunction is detected, this is either signalled or suitable program branchings are activated. The signal or branchings serve as a trigger for other co-operating system components to solve the problem.

WO

WO = write only

Unidirectional data transmission: Data can only be changed and not read.

9 Index

About this manual ... 5
Address ... 47
Application software ... 47
Architecture ... 47
Baud ... 47
Bus ... 47
Buttons ... 36
CAN ... 47
CAN stack ... 47
Category (CAT) ... 47
CCF ... 47
CiA ... 47
CiA DS 304 ... 47
CiA DS 401 ... 47
CiA DS 402 ... 48
CiA DS 403 ... 48
CiA DS 404 ... 48
CiA DS 405 ... 48
CiA DS 406 ... 48
CiA DS 407 ... 48
Clamp 15 ... 48
COB-ID ... 48
CoDeSys ... 48
Command line arguments of DOWNLOAD.EXE ... 38
Configure interfaces ... 10
CRC ... 48
Creating and transferring the data table to the device ... 44
Cycle time ... 48
DC ... 48
Demand rate rd ... 49
Diagnosis ... 49
Diagnostic coverage ... 49
Dither ... 49
Diversity ... 49
DRAM ... 50
DTC ... 50
ECU ... 50
EDS-file ... 50
Embedded software ... 50
EMCY ... 50
EMV ... 50
Ethernet ... 50
EUC ... 50
Example of a command list ... 40
Executable commands (only in combination with /DEBUG) ... 38, 39
Extended menu [Commands] ... 32
Failure ... 51
Failure, dangerous ... 51
Failure, systematic ... 51
Fault ... 51
Fault tolerance time ... 51
FiFo ... 51
Finding out the valid COM port ... 10
Firmware ... 51
First fault occurrence time ... 51
Flash memory ... 51
FMEA ... 52
FRAM ... 52
Functional safety ... 52
Harm ... 52
Hazard ... 52
Heartbeat ... 52
HMI ... 52
How is this manual structured? ... 6
ID ... 53
IEC user cycle ... 53
Important! ... 7
Installing and configuring the CANview driver ... 11
Instructions ... 53
Intended use ... 53
IP address ... 53
ISO 11898 ... 53
ISO 11992 ... 53
ISO 16845 ... 53
LED ... 53
Life, mean ... 53
Link ... 53
LSB ... 53
MAC-ID ... 54
Master ... 54
Menu [?] ... 35
Menu [File] ... 13
Menu [Interface] ... 17
Menu [Options] ... 24
Menu [Protocol] ... 22
Menu [Security] ... 23
Menu [Specials] ... 30
Menus ... 13
Mission time TM ... 54
Misuse ... 54
MMI ... 54
Monitoring ... 54
MRAM ... 54
MSB ... 54
MTBF ... 54
MTTF ... 54
MTTFd ... 54
Muting ... 55
NMT ... 55
Node ... 55
Node Guarding ... 55
Obj / object ... 55
Object directory ... 55
OBV ... 55
Operating system ... 55
Operational ... 55
PC card ... 55
PCMCIA card ... 55
PDM ... 55
PDO ... 55
PDU ... 56
Performance Level ... 56
PES ... 56
PGN ... 56
Pictogram ... 56
PID controller ... 56
PL ... 56
PLC configuration ... 56
PLr ... 56
Pre-Op ... 57
prepared ... 57
Process image ... 57
Programming command lines ... 9, 37
Programming language, safety-related ... 57
Protective measure ... 57
PWM ... 57
Ratio ... 58
RAW-CAN ... 58

Reading the application program from the device 43
Redundant 58
Remanent 58
Reset, manual 58
Residual risk 58
Risk 58
Risk analysis 58
Risk assessment 58
Risk evaluation 59
ro 59
RTC 59
rw 59
SAE J1939 59
Safety function 59
Safety instructions 7
Safety-standard types 59
SCT 60
SD card 60
SDO 60
Self-test 60
Setting up the interface on the PC 10, 17
SIL 60
Slave 60
SRDO 60
SRP/CS 60
SRVT 60
State, safe 61
Symbols 61
System description 9
System variable 61
Target 61
TCP 61
Template 61
Test rate rt 61
UDP 61
Updating the operating system in the device 41
Uptime, mean 61
Use, intended 61
Using the downloader (examples) 41
Watchdog 61
What do the symbols and formats mean? 5
What previous knowledge is required? 8
Window [CAN] 17, 20
Window [File Options] 25
Window [File System] 14, 15
Window [Memory Settings] 27
Window [Memory] 34
Window [Password] 29
Window [RS232] 17, 18
Window [USB] 17, 19
wo 62
Writing the application program to the device 42