



CoDeSys Soft-PLC



Copyright Note

Keep documentation for future use!

This documentation is the intellectual property of **Grossenbacher Systeme AG**, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation, as well as any distribution to third parties can only be made with the express permission of **Grossenbacher Systeme AG**.

Grossenbacher Systeme AG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Grossenbacher Systeme AG reserves the right to make complete or partial modifications to this document.

All brand and product names are trademarks or registered trademarks of the owner concerned.

Proper use

Hardware, software, operating systems and drivers must only be used for the applications specified and only in conjunction with the components recommended by Grossebacher Systeme AG.

Warning

No warranty claims will be recognized for faults arising from the improper handling of the device.

This device should not be used for the implementation of any safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect in the device.

All data specified in this document does not represent warranted properties in the legal sense.

Safety instructions for the user

This system description contains the information required for the proper use of the products described therein. It has been written for technically qualified personnel

Qualified personnel in the sense of these safety instructions in this system description or on the product are those persons:

- who are either familiar with the safety concepts of automation as project designers,
- or are instructed as operating personnel in the use of automation equipment and are familiar with the content of this system description related to operation,
- or have acquired the relevant training as commissioning or service personnel for the repair of such automation equipment, and are authorized to commission current circuits and devices/systems in compliance with standard safety engineering principles.

Contents

1	EXPLANATION OF SYMBOLS	4
2	INTRODUCTION	5
3	INSTALLATION	6
3.1	CODESYS TARGET.....	6
3.1.1	<i>CoDeSys SoftMotion</i>	7
3.1.2	<i>CoDeSys Target Visualization</i>	7
3.1.3	<i>CoDeSys web visualization</i>	7
3.2	RUNTIME SYSTEM.....	8
3.2.1	<i>Configuration file rts3S.cfg</i>	8
3.2.2	<i>Hardware</i>	9
3.3	CONFIGURATION OF PROFIBUS DP-MASTER/MPI OR SLAVE ON THE TARGET SYSTEM.....	10
3.4	RUNTIME SYSTEM UPDATE.....	10
4	SOFT PLC	11
4.1	FEATURES.....	11
4.2	STARTING SOFT-PLC.....	11
4.3	OPERATING STATES OF THE PLC.....	12
4.4	CHANGING OPERATING MODE.....	13
4.5	STARTUP BEHAVIOR/STARTUP.....	14
4.6	PLC BEHAVIOR IN THE EVENT OF A FAULT.....	14
4.7	FUNCTIONS OF THE PLC LIBRARIES.....	15
4.8	PLC BROWSER COMMANDS.....	16
4.9	RETAIN-DATA.....	16
5	PROGRAMMING THE PLC	17
5.1	CONNECTING THE PROGRAMMING PC WITH THE PLC.....	17
5.2	SETTING THE PROGRAMMING PC FOR ETHERNET COMMUNICATION.....	17
5.3	SETTING THE ETHERNET COMMUNICATION IN THE CODESYS PROGRAMMING INTERFACE.....	17
5.4	SETTING THE IP ADDRESS ON THE TARGET SYSTEM.....	17
5.5	SERIAL COMMUNICATION SETTINGS.....	17
6	FIELDBUS: CAN	18
6.1	CHARACTERISTICS OF THE CAN PROTOCOL.....	18
6.2	CANOPEN.....	19
6.2.1	<i>CANopen communication profile</i>	19
6.2.2	<i>CANopen device profiles</i>	20
6.3	WIRING.....	20
7	VERSION HISTORY	22
8	INDEX	23

1 Explanation of Symbols

Danger warnings

The following information is for your personal safety and the prevention of damage to the product described or connected devices.

Safety instructions and warnings for the prevention of danger to the life and health of users or service personnel, and for the prevention of damage are highlighted in this document by the following pictograms. "Warning" and "Information" pictograms are shown in this document.

Warnings indicate the following:

Death, serious injury or substantial material damage may occur if the related safety measures are not implemented.

The individual "Warning" pictograms have the following meaning:



Attention! General!

Is an instruction that must be observed in order to ensure protection in time from dangers during and at the time of the operation of the device. The correct procedure must be observed.



Attention! Electric shock!

Persons may be exposed to dangerous voltages in electrical installations. There is a danger of electric shock if a live part is touched.



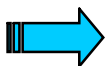
Attention! Observe ESD measures!

Electrostatic discharge may destroy electronic components.

Information pictograms indicate the following:

Important information about the product or the relevant section of the document, requiring the particular attention of the reader.

The "Information" pictogram has the following meaning:



Indicates important and instructional information.

2 Introduction

This documentation is supplementary to the 3S documentation of the CoDeSys runtime system and primarily describes adaptations and special features for the Grossenbacher panels.

More extensive documentation on the 3S runtime system and the CoDeSys programming system can be obtained from: Program Files\Codesys Vx.x\Documents and in the CoDeSys Online Help.

Overview of the most important documents:

Subject	Documentation	Location
First steps	First steps with CoDeSys V23.pdf System description of CoDeSys Soft-PLC.pdf (this document)	...\Codesys V2.3\Documents and Online Help Grossenbacher target file (SetupCodesysTarget: 5056831)
CoDeSys	Codesys_V23_x.pdf	...\Codesys V2.3\Documents and Online Help
CoDeSys Visualization	CoDeSys_Visu_V23_x.pdf	...\Codesys V2.3\Documents and Online Help
SoftMotion	CoDeSys_SoftMotion_V23_x.pdf	...\Codesys V2.3\Documents and Online Help
CANopen	CanOpen for 3S RTS.pdf	Grossenbacher target file (SetupCodesysTarget: 5056831)
System libraries	Syslibxxx.pdf	...\Codesys V2.3\Documents and Online Help
CanUser.lib CanUser_Master.lib	h1554D/G.pdf (function compatible with XC200)	H1554x.pdf Grossenbacher Targetfile (SetupCodesysTarget: 5056831)
EPAM	EPAM manual.pdf	...\EPAM\Docs
WindowsCE	System description WindowsCE 48650	www.gesys.ch
EthernetPanel	Device description 48960	www.gesys.ch
Compact-Panel-PC/Compact-PC	Device description 38465	www.gesys.ch
CIO-CAN	Device description Compact I/O Modul CAN 51324	www.gesys.ch

3 Installation

In order for Grossenbacher panels to be programmed with CoDeSys, the CoDeSys V2.3 (SP5 or newer) programming tool and the corresponding target must be installed beforehand.

The CoDeSys programming tool can be downloaded from the 3S website (Support & Download – Customer Download):

<http://www.3s-software.com>

or from the Grossenbacher Systeme website:

<http://www.gesys.ch>



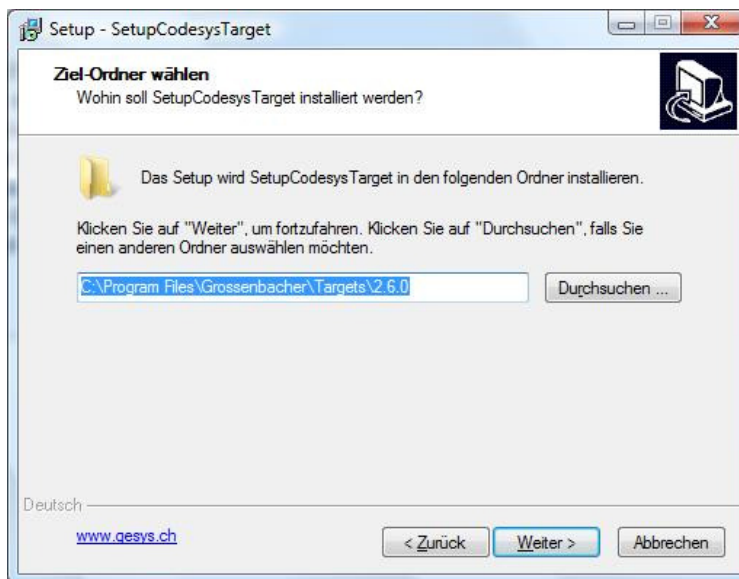
Grossenbacher panels are tested with the CoDeSys version 2.3 SP5 (2.3.5.8). Normally any newer version can be used without any problems. But functions cannot be guaranteed if other CoDeSys versions are used!

3.1 CODESYS TARGET

The target installation can be done with SetupCodesysTarget-Vxxxx.exe from Grossenbacher Systeme. This file is available via: http://www.easypagemachine.com/dnld_codesys.php

Proceed as follows to install the target:

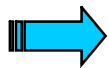
1. Install the Grossenbacher target file “SetupCodesysTarget-Vxxxx.exe” (installs all required files for “InstallTarget” in the selected directory; Default: C:\Program Files\Grossenbacher\Targets\x.x.x)



2. The Setup-program installs automatically the Grossenbacher Targets by the help of the program Install Target from 3S (this program will be installed by Codesys installation). With Install Target it is possible to show all installed targets for Codesys.
3. The appropriate target can then be selected in CoDeSys under “Target Settings-Configuration”

3.1.1 CoDeSys SoftMotion

The target file for Grossenbacher panels contains support for CoDeSys SoftMotion. SoftMotion can be activated in CoDeSys: “Target Settings” □ “General” □ “SoftMotion”

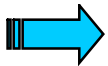


CoDeSys Soft-Motion requires a valid runtime license!

3.1.2 CoDeSys Target Visualization

The target file for Grossenbacher panels contains support for the CoDeSys target visualization. The target visualization can be activated in CoDeSys: “Target Settings” □ “Visualization” □ “Target Visualization”

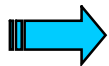
The target visualization requires the library: SysLibTargetVisu.lib incorporated in the project.



CoDeSys target visualization requires a valid runtime license!

3.1.3 CoDeSys web visualization

The target file for Grossenbacher panels contains support for the CoDeSys web visualization. The web visualization can be activated in CoDeSys: “Target Settings” □ “Visualization” □ “Web Visualization”



CoDeSys web visualization requires a valid runtime license!

The Codesys web server must be installed on the panel for CoDeSys web visualization. The Codesys web server is located in the directory StorageCard\PLCRts\webserver.exe

To start the web server, the PLC.BAT must be modified (remove REM in front of the line webserver.exe).

Example PLC.BAT:

```
REM Start the PLC-RTS  
CALL \StorageCard\PlcRts\PLCWinCEX86.exe %1 %2
```

```
REM Start the WebServer for WebVisu  
START \PlcRts\webserver.exe
```

After a Codesys project with web visualization is downloaded, the project can be run in the Internet Browser as follows (see also Codesys_Visu_Vx.x_D.pdf: Sect. 4):

<http://<IP address of the device>:8080/webvisu.htm>

Example:

<http://192.168.0.99:8080/webvisu.htm>

3.2 RUNTIME SYSTEM

The runtime system is based on the WindowsCE RTS of 3S V2.4.4 and consists of the following files on the CompactFlash in the directory **\StorageCard\PLCRts**:

PLCWinCEx86.exe	...Soft-PLC runtime system
CPC650.dll	...System adaption
HiIDPM3S.dll	...Hilscher fieldbus driver
3SSJA1000.dll	...CAN bus driver
SercosAutomata_1KW	...Sercos driver
SymArtilib.dll	...SymArti driver
SymArtiClient.dll	
ArtiClient.dll	
Rts3s.cfg	...System configuration file
Shortcut to PLCWinCEx86 (RAM).lnk	...Link to \PLCRts\PLCWinCEx86.exe
Webserver.exe	...CoDeSys web server
<u>Webserver_conf.xml</u>	...Configuration CoDeSys web server

The directory **\StorageCard\PLCPrg** contains the PLC boot project

The Soft-PLC is automatically started via the Autoexec.cmd when the panel is switched on. (see also WindowsCE system description) This copies all the required files to the directory **\PLCRts** in the RAM, so that the runtime system can be updated during operation (see also [Runtime System Update](#))

3.2.1 Configuration file rts3S.cfg

The configuration file rts3S.cfg contains device specific settings and configurations. This file is factory configured and does not have to be changed later.

The following default configurations will be delivered:

- rts3s_Standard.cfg ...Standard-configuration (CAN1 and optional ProfibusDP)
- rts3s_2xCAN.cfg ...configuration for Option 2.CAN and additional 24kByte Retainmemory
- rts3s_DeviceNet.cfg ...configuration for Option Devicenet

Example standard configuration with CAN1

```

rts3s_Standard.cfg - Editor
Datei Bearbeiten Format Ansicht ?
[IODRIVERLIST]
MODULE0=CPC650
MODULE1=3SSJA1000
MODULE2=HiIDPM3S
MODULE3=SercosAutomata_1KW
MODULE4=SymArtilib
INITFCT0=IODrvInit
INITFCT1=IODrvInit
INITFCT2=IODrvInit
INITFCT3=IODrvInit
INITFCT4=IODrvInit
[PLC]
SETTINGS_FROM_CFG=Yes
USE_SYSINTR_TIMING_AS_TICK=Yes
Files=\StorageCard\PLCPrg
BootMode=Run
DisableSerialProgramming=No
ResetwarmonStop=Yes
HwType=20
UseIRQ11ForExternalEventTask=Yes
ObjectStoreMinPages=3072
[RETAIN]
TYPE=Direct
SIZE=0x8000
ADDRESS=0x800D4000
[3SSJA1000]
SJAInt0=5
SJAMemAddr0=0x800D0000
SJAPCIScan=0
SJAOutputControlMode=0x5e
SJAEndTestMsgAfterInit=0
SJAWaitScnf1g=1
[HiIDPM3S]
DISABLEISASCAN=1
DISABLEPCISCAN=1
DEVICE0_DPM_ADDRESS=0x800D2000
DEVICE0_DPM_SIZE=0x2000
_WATCHDOG=1000
[WATCHDOG]
Enable=0
[COMMUNICATION]
Port=1
Baudrate=115200
[SERCOSAUTOMATA]
IRQ=11
ADDRESS=0x000ce000
    
```

Use with the addresses (RETAIN, 3SSJA1000, HILDPM3S) absolute values (0x800D2000 instead of 0xD2000).

Additional entries for CAN2:

SJAInt1=11 Interrupt vector (see also device description; for EP-300: SJAInt1=7)
SJAMemAddr1=0x800D1000 Address

3.2.2 Hardware

The Soft-PLC can be used with the industrial PC series of the Grossenbacher panels with WindowsCE. The available interfaces are device specific (see also device description). The following interfaces and functions are supported by the software:

Configuration:

RetainMemory:

32KByte ...Configurable address and size (Default: 32KB on address 0x800D4000)
24kByte ...additional 24kByte available with option 2.CAN (s.a. rts3s_2xcan.cfg)

HW Watchdog:

None

CAN (optional)

- Number of ports: max. 2
- Address: Freely definable, Memory mode, Default: CAN1=0x800D0000, CAN2=0x800D1000
- Interrupt vector: Freely definable, Default: CAN1 = IRQ 5, CAN2 = IRQ 11 (CPC) or 7 (EP)

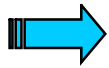
ProfibusDP-Master/Slave/MPI (PC104 Multi-I/O Interface) (optional)

- Address: Freely definable, Default: 0x800D2000, only one module
- Dual-Port RAM size: 8KByte

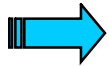
3.3 CONFIGURATION OF PROFIBUS DP-MASTER/MPI OR SLAVE ON THE TARGET SYSTEM

The DP master/MPI and DP slave should be configured by loading the appropriate firmware on the Profibus module (device version with Profibus-DPM). When the PLC-RTS is started, any PB.G07 file present is loaded.

The utility IoCfg.exe (StorageCard\Tools) can be used to configure the Profibus-card as Master/MPI or Slave. The required FW files PBM.G07 and PBS.G07 are also located within the directory StorageCard\Tools. (details s.a. system description WindowsCE)



**The “Profibus-DPM” option enables the device to be run as a DP master/slave or with the MPI protocol.
The Profibus-DPS/MPI option only enables the device to be run as a DP slave or with the MPI protocol.**



The Profibus-DP configuration is device specific and is not available with all devices (see also device description).

When configured as DP master, the device can also be run with the MPI protocol. For this the MPI_lib.lib is required.

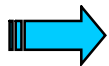
With a DP slave configuration the GSD file HIL_06e7.gsd must be used in the DP master configuration.

3.4 RUNTIME SYSTEM UPDATE

The entire runtime system of the controllers, including the WindowsCE operating system is stored on the Compact Flash memory card in addition to the user program and the EPAM project.

The runtime system update can be carried out using the SetupTargetFirmwareWinCE86-Vxxx.exe program via CompactFlash adapter and PC or via FTP (see also WindowsCE system description)

SetupTargetFirmware is used to install the WindowsCE operating system and tools, the PLC and the EPAM Runtime System.



Retentive PLC data (retain data) is not automatically stored on the CF card and is therefore not copied!

To back up and restore retentive data, see also Codesys PLC Browser commands:

Saveretain - Saves retentive data in file [filename]

Restoreretain- Restores retentive data from file [Filename]

4 Soft PLC

4.1 FEATURES

Features	
User memory PLC program complete Variables complete Marker range Retain variables I/O process image	Size adjustable (default: 2MB), boot project on CompactFlash (StorageCard\PLCPrg) Size adjustable, DRAM (default: 256KB) Max. 64KByte, SRAM retentive, can be used as Retain ¹ Max. 64KByte, SRAM retentive 4KByte each
PLC functionality Data types Software timer Program structure	BIT, BYTE, WORD, DWORD, REAL, STRING, TIME, DATE Number unlimited, 1 ms Multi tasking (up to 32 tasks)
System library CoDeSys System libraries	All (see also Functions of the PLC Libraries)
Grossenbacher libraries	(see also Functions of the PLC Libraries) SysLib_Add.lib SysLibRtc_Add.lib SysLibCan.lib CanUser.lib CanUser_Master.lib SymArti.lib
I/O interface	CANopen, integrated CAN controller (CAN1) PROFIBUS-DP Master/Slave Optional CAN2
SoftMotion Drive interface	Sercos CAN, CANopen (I/Os and drives cannot be operated on the same CAN network, in this case, a CAN expansion module (CAN2) is required)
PLC programming Interface Programming system Online functions Variables monitoring	Ethernet or serial (COM1) CoDeSys from V2.3 (recommended V2.3 SP5) Load/start/stop program, break point, single step, single cycle I/Os, variables, trace recording etc.
Performance	PLC test program: 11 us / 1000 commands (CPC-650) Commands: LD, AND, ST
CoDeSys Automation Alliance (CAA)	The runtime system for the EP-300, CPC-300 and CPC-650 devices is certified by 3S for "CAA-Compliance". (CAA registration: CAA:8A57EBC4-CF57C8A3-C836AAA3)

4.2 STARTING SOFT-PLC

The Soft-PLC (StorageCard\PLCRts\PLCWinCEx86.exe) is started automatically when the device is powered up (see also Autoexec.cmd and PLC.BAT). The following options are possible when PLCWinCEx86.exe is called (see also StorageCard\PLCRts\PLC.BAT):

- /i Soft-PLC (Folder \StorageCard\PLCRts) is copied first of all to the RAM file system (\PLCRts) and started there. A link to the runtime system is also created on the desktop. (Default)
This option allows the RTS to be updated during the runtime, e.g. via FTP.
- /h The window of Soft-PLC is minimized in the taskbar after the start (default) and can be maximized by clicking it (/minimize causes the same thing).

¹ The size of the retentive memory is device specific (see also device description)

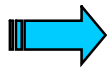
4.3 OPERATING STATES OF THE PLC

The operating states are displayed in the PLC window, in the taskbar (LED icon at bottom right) and via the status LEDs². When the PLC is in RUN status, the RUN LED (green LED) is lit² and a green LED is also shown in the information area of the taskbar. A red LED is displayed in the taskbar when the PLC is in the other states (STOP, STOP-BREAKPOINT, ...). In the event of a fault, the FAIL-LED (red LED) is lit².

Power on

In the “Power on” state, there is no user program on the PLC. In this state a new program can be loaded onto the PLC. “Power on” status is indicated in the programming software.

Stop (Stop RESET)



**The “Stop” mode does NOT comply with the 3S Standard! In “Stop” mode a Stop Reset is carried out on the Grossenbacher panels!
(set ResetWarmOnStop=No in RTS3s.cfg file to configure the standard handling)**

The “Stop” status has the following characteristics:

- A user program is located in the PLC
- The user program is not run

“Stop” resets all tag variable values to their initial values. However, this does not apply to the retentive variables (RETAIN, PERSISTANT). Variables that are not explicitly assigned an initialization value are set to the standard initial values (integer values such as 0). All outputs are set to 0.

“Stop” status is selected:

- After the power supply is switched on if the mode selector switch is set to “Stop-RESET”
- Via the programming software in the PC (Online Stop and Reset)
- After a cycle timeout

Run

In RUN status, the user program is processed cyclically.

RUN status is selected:

- After the power supply is switched on if the mode selector switch is set to “RUN”
- Via the programming software in the PC

System Fault SF

The following system faults may occur:

- Hardware faults
- System faults of the firmware

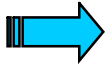
4.4 CHANGING OPERATING MODE

The operating mode is changed via:

Operating mode switch²

Position: Stop-RESET ...PLC starts in STOP mode

Position: RUN ...PLC starts in RUN mode



**If the operating mode switch is set to Stop-Reset, the PLC can neither be started from Codesys nor via the Start button in the PLC window!
If the PLC has no operating mode switch, it always starts in RUN mode.**

Programming software

Reset, Stop

Reset or Stop resets all variable values to their initial values. However, this does not apply to the retentive variables (RETAIN, PERSISTANT). Variables that are not explicitly assigned an initialization value are set to the standard initial values (integer values such as 0). All outputs are set to 0. CodDeSys requests confirmation before all variables are overwritten. The situation is similar to that of a power failure or when the PLC is switched off and on (warm start).

Use the 'Online' 'Start' command to restart the PLC and thus also program processing.

Reset-Cold

This command resets all variables, also RETAIN variables to their initial values. Only persistent variables retain the value they had before the reset. All outputs are set to 0.

Reset original

This command resets all variables, including the retentive variables (RETAIN and PERSISTANT) to their initial value, and deletes the user program in the PLC. The PLC is reset to its original state. All outputs are set to 0.

Start

This command starts the processing of the user program in the PLC.

The command can be executed, directly after the command 'Online' 'Load' or after the user program in the PLC was stopped with the command 'Online' 'Stop', or when the user program is on a break point, or if the command 'Online' 'Single Cycle' was executed.

A detailed description of the online functions is provided in the Codesys manual.

BREAKPOINT: All outputs remain set

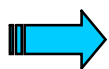
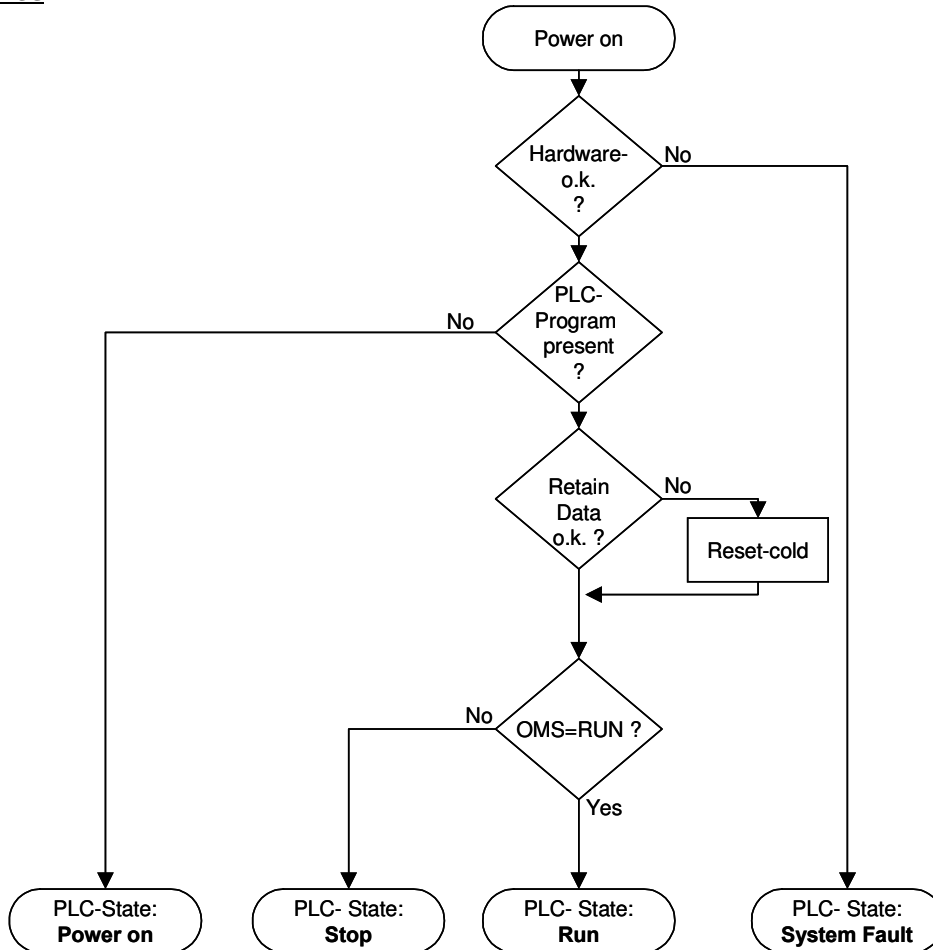
WATCHDOG: Outputs are set to 0; status of the variables is retained, RESET WARM is executed on START.

² device specific (see also device description)

4.5 STARTUP BEHAVIOR/STARTUP

After the power supply is switched on, the user program is loaded from the CompactFlash™ into the RAM and the PLC starts up.

Sequence



If the PLC has no operating mode switch, it always starts in RUN mode.

4.6 PLC BEHAVIOR IN THE EVENT OF A FAULT

Error	Behavior of the PLC	SF-LED ³
Watchdog (cycle timeout)	Outputs are set to 0; status of the variables is retained, RESET WARM must be executed beforehand on START.	On
Integer division by zero (Integer)	As per Watchdog	On
Floating number division by zero (Real)	Incorrect result, PLC continues running must be intercepted in the application!	-
Battery fault	Must be configured in the application (see also SysRtcCheckBattery)	-

³ device specific (see also device description)

4.7 FUNCTIONS OF THE PLC LIBRARIES

Library	Functionality	Documentation	3S
3S_CanDrv.lib	Firmware interface for CAN driver	Standard system library 3S ⁴	<input type="checkbox"/>
3S_CANOpenManager.lib	CANopen Manager	3S standard system library	<input type="checkbox"/>
3S_CANOpenMaster.lib	CANopen Master	3S standard system library	<input type="checkbox"/>
3S_CANOpenNetVar.lib	CANopen network variables	3S standard system library	<input type="checkbox"/>
3S_CanOpenDevice.lib	CANopenDevice	3S standard system library	<input type="checkbox"/>
BUSDIAG.LIB	Diagnostics functions for Hilscher fieldbus cards (Profibus)	FB description in the library manager	<input type="checkbox"/>
CanUser.lib CanUser.exp	CANopen Userlibrary - general functions: Diagnostics, Read/Write CAN object	Moeller AWB2786-1554D/GB (h1554x.pdf; comp. with XC200)	
CanUser_Master.lib CanUser_Master.exp	CANopen Userlibrary - Master functions: SDO	Moeller AWB2786-1554D/GB.pdf (comp. with XC200)	
NetVarUdp_LIB_V23.lib	Communication with UDP network variables	3S standard system library	<input type="checkbox"/>
MPI_Lib.lib	MPI communication with the DPM card (in preparation)		
SymArtiLib.lib	Communication via the SymArti interface (symbolic PLC variables communication)	FB description in the library manager	
SysLibAlarmTrend.lib	Alarm handling for TargetVisu	3S standard system library <input type="checkbox"/> Codesys_Visu_V23_D.pdf	<input type="checkbox"/>
SysLibCallback.lib	Activating defined callback functions for runtime events	3S standard system library <input type="checkbox"/> SysLibCallback.pdf	<input type="checkbox"/>
SysLibCan.lib	Grossenbacher specific Systemlib. Basic functions for CanUser.lib and CanUser_Master.lib	Not for the user!	
SysLibCom.lib	Serial communication with a target computer	3S standard system library <input type="checkbox"/> SysLibCom.pdf	<input type="checkbox"/>
SysLibEvent.lib	Event handling	Standard 3S system library <input type="checkbox"/> SysLibEvent.pdf	<input type="checkbox"/>
SysLibFile.lib	Handling a file system on the target PC	3S standard system library <input type="checkbox"/> SysLibFile.pdf	<input type="checkbox"/>
SysLibFileStream	ANSI C compatible file functions	3S standard system library <input type="checkbox"/> SysLibFileStream.pdf	<input type="checkbox"/>
SysLibDir.lib	Handling a file system on the target PC	3S standard system library <input type="checkbox"/> SysLibDir.pdf	<input type="checkbox"/>
SysLibIecTasks.lib	Management of IEC tasks	3S standard system library <input type="checkbox"/> SysLibIecTasks.pdf	<input type="checkbox"/>
SysLibMem.lib	Functions for memory management	3S standard system library <input type="checkbox"/> SysLibMem.pdf	<input type="checkbox"/>
SysLibPlcCtrl.lib	Control of the PLC for handling retain values, as well as for activating the watchdog	3S standard system library <input type="checkbox"/> SysLibPlcCtrl.pdf	<input type="checkbox"/>
SysLibPorts.lib	Functions for accessing I/O ports	3S standard system library <input type="checkbox"/> SysLibPorts.pdf	<input type="checkbox"/>
SysLibProjectInfo.lib	Functions for reading project information	3S standard system library <input type="checkbox"/> SysLibProjectInfo.pdf	<input type="checkbox"/>
SysLibRtc.lib	Reading the PC real-time clock	3S standard system library <input type="checkbox"/> SysLibRtc.pdf	<input type="checkbox"/>
SysLibSem.lib	Generating and using semaphores for task synchronization	3S standard system library <input type="checkbox"/> SysLibSem.pdf	<input type="checkbox"/>
SysLibShm.lib	Functions for accessing shared memory	3S standard system library <input type="checkbox"/> SysLibSem.pdf	<input type="checkbox"/>
SysLibSockets.lib	Access to sockets for communication via TCP/IP and UDP	3S standard system library <input type="checkbox"/> SysLibSockets.pdf	<input type="checkbox"/>
SysLibStr.lib	String functions	3S standard system library <input type="checkbox"/> SysLibStr.pdf	<input type="checkbox"/>
SysLibTasks.lib,	Management of tasks	3S standard system library <input type="checkbox"/> SysLibTasks.pdf	<input type="checkbox"/>
SysTaskInfo.lib	PLC task info	3S standard system library <input type="checkbox"/> FB description in the library manager	<input type="checkbox"/>

⁴ for CAN see also documentation "CANopen for 3S RTS"

SysLibTargetVisu.lib, SysLibVisu.lib	Codesys visualization	3S standard system library <input type="checkbox"/> Codesys_Visu_V23_D.pdf	<input type="checkbox"/>
SysLibTime.lib	Additional functions for reading the real-time clock from the PC (see also SysLibRtc.lib); required for displaying the task time evaluation in the CoDeSys task configuration	3S standard system library <input type="checkbox"/> SysLibTime.pdf	<input type="checkbox"/>
SysLib_Add.lib	Grossenbacher panel additional functions (e.g. read/set IP address, read MAC address etc.)	<input type="checkbox"/> FB description in the library manager	
SysLibRTC_Add.lib	Grossenbacher panel additional functions for date/time conversion	<input type="checkbox"/> FB description in the library manager	

4.8 PLC BROWSER COMMANDS

The following commands are implemented in addition to the commands listed in section 6.13.2, "Command Entry in the PLC Browser" of the Codesys manual:

Command	Function	Note
tskclear	Delete the IEC task info (cycle count, cycletime-max/min/avg)	
reboot	Restart the target system.	CoDeSys is logged out
plcload	Display of the CPU load as a percentage (PLC load of the scheduler, IEC tasks and communication)	The value gives a rough indication of the system load. The value can vary according to the actual processing time required for the PLC project. A very high CPU load (>80%) can impair the functionality of the system (e.g. visualization, FTP server, RemoteServer etc.).
rtsinfo	Display of the runtime system information (Version, IO driver)	
traceschedon	Task trace on	
traceschedstore	Save task trace to file [filename]	
SaveRetain	Store retain data in file on CF card	
RestoreRetain	Load retain data from file	

4.9 RETAIN-DATA

Retain data will be stored within a nonvolatile memory range. The available memory range is hardware specific (s.a. device description). The last 12 Bytes of this memory range are reserved and contain a project specific ID. If a CF is moved from one device to another or if a different project is loaded the retain data will be initialized.

5 Programming the PLC

The PLC is programmed ideally via Ethernet with TCP/IP protocol.

5.1 CONNECTING THE PROGRAMMING PC WITH THE PLC

A “Crosslink” cable must be used for a direct connection without an Ethernet hub or switch. A standard RJ45 socket is provided on the PLC side.

If the Ethernet connection is implemented via an Ethernet hub or switch, patch cables must be used from the hub to the PLC as well as to the programming PC.

5.2 SETTING THE PROGRAMMING PC FOR ETHERNET COMMUNICATION

A network adapter and the TCP/IP protocol must be installed on the programming PC. A suitable IP address, i.e. in the same subnet, must be set on the PLC.

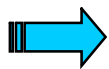
To do this contact your network administrator.

Example of a possible configuration:

Programming PC: 192.168.0.99

PLC: 192.168.0.100

Subnet mask on both devices: 255.255.255.0



The communication between the PC and PLC can be checked simply using the ping “IP address” command line command. If the program indicates ping timeouts, either the cabling or the TCP/IP setting are not correct.

5.3 SETTING THE ETHERNET COMMUNICATION IN THE CODESYS PROGRAMMING INTERFACE

The IP address setting for the PLC must be carried out in Codesys via “Online-Communication Parameters”. The TCP/IP driver must be selected for Ethernet communication and the IP address of the device must be entered. (see also Codesys manual)

5.4 SETTING THE IP ADDRESS ON THE TARGET SYSTEM

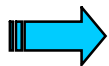
The IP address can be set via the Control Panel on the WindowsCE devices (“Start-Settings-Network and Dial-up Connections” see also WindowsCE system description) or with an EPAM application. (see also EPAM manual)

5.5 SERIAL COMMUNICATION SETTINGS

Programming can also be carried out via the standard RS232 interface (COM1). A standard null modem cable is used for the connection to the programming PC.

For programming via the serial interface the serial (RS232) driver must be configured in Codesys under Communication Parameters.

The default values of the PLC are as follows: COM1: 115200 baud, No parity, 1 Stop bit.



**The serial interface can also be used in the PLC application. In this case, the following entry must be made in the file rts3s.cfg (StorageCard\PLCRts\rts3s.cfg):
DisableSerialProgramming=Yes (default is No)
COM1 cannot be used simultaneously as a programming interface and in the application!**

6 Fieldbus: CAN

6.1 CHARACTERISTICS OF THE CAN PROTOCOL

CAN (Controller Area Network) was originally developed for use in automotive applications. The protocol has been implemented by several semiconductor manufacturers that offer these CAN controllers with different features so that it now covers a wide range of applications. CAN based systems are currently used in industrial, process and building automation, as well as in the vehicles of virtually every automotive manufacturer.

Two of the most important features of CAN are its high level of transmission security, even in harsh electrical environments, and its extremely short reaction times.

CAN operates as a multi-master system, i.e. all nodes can communicate with each other without a special master required. This is achieved by the so-called "Broadcasting" procedure. With this process network stations or nodes are not addressed like in conventional fieldbus systems, but prioritized messages are sent from one transmitter to all other nodes. Each receiver node decides independently on the basis of the message identifier whether it will process the message or not. The identifier also determines the priority of the message and is used to control the bus access of the nodes.

CAN's bus access method is completely unsynchronized and is called non-destructive bitwise bus arbitration. If two or more nodes access the bus at the same time, this arbitration procedure ensures that the message with the highest priority is processed without delay and messages with a lower priority have to be repeated.

Unlike conventional (master-slave) bus systems, a lower bus load is achieved because the data transfer on the CAN network is only 'event-driven', i.e. it only takes place if the input signal of an I/O module is changed. The direct communication described above also reduces the number of messages by half without having to send every message to the master. These two benefits lead to the very short reaction times of CAN-based systems.

CAN therefore offers a high performance, even with relatively low transfer rates, whilst EMC problems and cable costs can also be reduced.

Another important feature of the CAN protocol is the fact that a high level of transmission security (Hamming distance 6) is achieved by using a number of interrelated and complementary error detection mechanisms.

CAN uses a linear topology with shielded twisted-pair cables and bus terminating resistors at both ends of the bus line.

The data transfer speed is specified between 10 Kbit/s to 1 Mbit/s. At 50 Kbit/s, for example, a bus length of 1000 m is possible. The bus length must be reduced with higher transfer speeds, for example, to 25 m at 1 Mbit/s.

6.2 CANOPEN

An open network system requires the use of a standard method of communication as a basis. CANopen therefore makes use of the international CAN Standard ISO 11898 as a basis for communication.

This standard covers layers 1 and 2 (Physical and Data Link Layer) of the ISO/OSI communication model.

The CANopen profile family was developed from this within the 'CAN in Automation' (CiA) international user and manufacturer organization. This profile family consists of the communication profile (CiA DS-301) that represents the ISO/OSI layer 7 (Application Layer) and specifies the communication mechanisms, and from a number of device profiles (CiA DS-40X) that specify the functionality of the devices. The following device profiles currently exist as 'CiA Draft Standard Proposals':

for

- I/O Modules' (CiA DS-401),
- Drives and Motion Control' (CiA DS-402),
- Measuring Devices and Closed-Loop Control' (CiA DS-404) and
- Encoders' (CiA DS-406).

6.2.1 CANopen communication profile

The CANopen communication profile specifies how data should be exchanged between the nodes of a network. Four different types of message were defined for this:

- Network administration messages
- Service Data Objects (SDOs)
- Process Data Objects (PDOs)
- Predefined Messages

Network administration messages are used to control the nodes and their operating states in the network. With this type of message, it is possible, for example, to configure the data transfer mechanisms of a node. Another option of this type of message is 'Node Guarding', the CANopen mechanism by which nodes can be detected that are no longer participating in network communication.

Service Data Objects are used for low-priority acyclical data transfers in CANopen networks. SDOs are typically used for the configuration of CANopen nodes, for setting device parameters and for downloading programs. SDOs allow any data length to be transferred by using the so-called 'segmented transfer'.

Process Data Objects are used for high priority and high-speed data transfers in CANopen networks. PDOs do not contain any protocol overhead, and thus provide a very fast and flexible type of data transfer from one node to any number of other nodes.

PDO data transfers can be carried out in many different ways. This can be 'event controlled', i.e. the data transfer only takes place when a specific event occurs. For example, this event can be the change in status of a digital signal or the change of an analog signal by more than a predefined value.

This type of data transfer keeps the bus load to a minimum and thus enables a high level of communication with a very short reaction time and a relatively low data rate. All relevant process and application data should be transferred with this message type.

Data in these PDOs is restricted to 8 bytes. The content and the structure of each PDO can be configured individually. The PDOs can also be used to read data out of network nodes. For this CAN uses the 'remote transmission request'.

Predefined Messages are specified in CANopen, for example for synchronization tasks. This is particularly important in applications in which drives or I/O modules have to be read out or written simultaneously.

6.2.2 CANopen device profiles

The CANopen device profiles not only define standard device functions but they also contain options for a structured implementation of manufacturer specific features. These device profiles enable manufacturers to produce standard CANopen devices that can be operated in a network with the devices of other manufacturers. As previously mentioned, a number of device profiles already exist for different devices.

Thanks to all these characteristics, CANopen is an ideal network system for virtually all types of automated devices and machines.

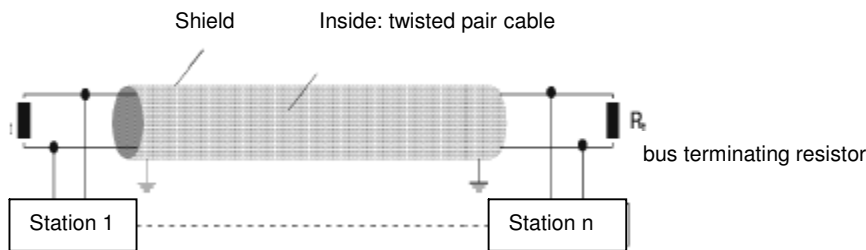
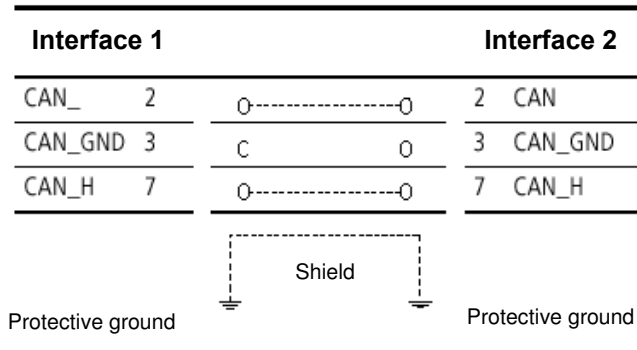
CANopen is thus ideally suited for data exchange on both the control level and the field level with sensors and actuators, with the entire system integrated in one physical network.

6.3 WIRING

The stations on the bus system are interconnected with fieldbus cables in compliance with ISO 11898. The cables must therefore have the following electrical properties:

Parameter	Abbreviation	Unit	Value	Value	Value	Note
			min.	nom.	Max.	
Impedance	Z	Ω	108	120	132	Measured between two signal cables
Spec. resistance		m Ω /m		70		For the receiving module, the voltage difference on the bus cable depends on the cable resistance between itself and the transmitter.
Cable delay		ns/m		5		The minimum delay time between two points on the bus is taken as zero. The maximum delay is determined by the bit timing and the delay of the transmit and receive circuits.

The figure shows the minimum wiring with shielding between two bus nodes using a sub-D connector in the example. A bus terminating resistor (120 Ohms between pins 2 and 7 of the sub-D connector) must be connected at the beginning and end of each CAN bus. The two signal wires must be reversed!



Note on CAN interface of HMI-PLC controllers:

A bus terminating resistor is not provided internally! Pins 3 and 6 (CAN_GND) are both connected internally with the CAN Ground. Pins 4, 5 and 8 must *not* be connected! The CAN bus drivers are fed internally. The internal supply is *not* provided on Pin 9. (see also device description)

Baud rate and cable lengths

Baud rate	Max. length
10Kbit/sec	2500m
20Kbit/sec	2500m
50Kbit/sec	1000m
100Kbit/sec	650m
125Kbit/sec	500m
250Kbit/sec	250m
500Kbit/sec	100m
800Kbit/sec*	20m
1000Kbit/sec*	6m

Note: A baud rate of 800Kbit/sec as well as 1000Kbit/sec is not supported on all devices.

7 Version History

Version	Date / Signed	Modifications:
1.0	06-06 / Fis	Initial version
2.0	05-08 / Fis	Revision for Codesys RTS V2.4.4
3.0	02-10 / CMI	New CI

Grossenbacher Systeme AG
Spinnereistrasse 10
CH-9008 St.Gallen
Switzerland

Tel : +41(0)71/243 29 29
Fax : +41(0)71/243 29 28
Email : info@gesys.ch
Internet : www.gesys.ch

8 Index

B	
Baud rate and cable lengths	21
Bus termination	21
C	
CAA compliance.....	11
CANopen.....	19
CANopen communication profile	19
CANopen device profiles	20
Changing operating mode	13
Characteristics of the CAN protocol	18
CoDeSys Automation Alliance (CAA).....	11
CoDeSys SoftMotion.....	7
CoDeSys Target	6
CoDeSys target visualization	7
CoDeSys web visualization.....	7
Configuration file rts3S.cfg	8
Configuration of Profibus DP master/MPI or slave on the target system	10
Connecting the programming PC with the PLC	17
Contents	3
E	
Explanation of Symbols	4
F	
Features	11
Fieldbus: CAN	18
Functions of the PLC libraries.....	15
G	
GSD file HIL_06e7.gsd	10
I	
Installation	6
Introduction	5
M	
Manufacturer address.....	22
MPI protocol.....	10
O	
Operating states of the PLC.....	12
Power on.....	12
Run.....	12
Stop.....	12
System fault.....	12
P	
PLC behavior in the event of a fault	14
PLC browser commands	16
Programming the PLC.....	17
R	
Reset	13
Reset original.....	13
Reset-Cold	13
Retain-Data.....	16
RS232 interface	17
Runtime system.....	8
Runtime system update	10
S	
Serial communication settings.....	17
Setting the Ethernet communication in the CoDeSys programming interface.....	17
Setting the IP address on the target system ..	17
Setting the programming PC for Ethernet communication	17
Soft PLC	11
Start.....	13
Starting Soft-PLC.....	11
Startup behavior/startup	14
V	
Version history	22
W	
Wiring.....	20

