

sysWORXX CTR-500

System Manual Version 1.0

Edition January 2022

Document No.: L-2562e_01

SYS TEC electronic AG Am Windrad 2 D-08468 Heinsdorfergrund Phone: +49 (3765) 38600-0 Fax: +49 (3765) 38600-4100 Web: https://www.systec-electronic.com Mail: info@systec-electronic.com

SYS TEC electronic AG - System House for Distributed Automation

Changes

Date/Version	Section	Changes	Author/Editor
04.01.2020	-	Initial version	C. Sc.

Product names used in this manual which are also registered trademarks have not been marked extra. The missing © mark does not imply that the trade name is unregistered. Nor is it possible to determine the existence of any patents or protection of inventions on the basis of the names used.

The information in this manual has been carefully checked and is believed to be accurate. However, it is expressly stated that SYS TEC electronic AG does not assume warranty or legal responsibility or any liability for consequential damages which result from the use or contents of this user manual. The information contained in this manual can be changed without prior notice. Therefore, SYS TEC electronic AG shall not accept any obligation.

Furthermore, it is expressly stated that SYS TEC electronic AG does not assume warranty or legal responsibility or any liability for consequential damages which result from incorrect use of the hardware or software. The layout or design of the hardware can also be changed without prior notice. Therefore, SYS TEC electronic AG shall not accept any obligation.

© Copyright 2022 SYS TEC electronic AG, D-08468 Heinsdorfergrund

All rights reserved. No part of this manual may be reproduced, processed, copied or distributed in any way without the express prior written permission of SYS TEC electronic AG.

Contact	Direct	Your Local Distributor	
Address:	SYS TEC electronic AG Am Windrad 2 D-08468 Heinsdorfergrund GERMANY	Please contact us for	
Ordering Information:	+49 (0) 37 65 / 38 600-0 info@systec-electronic.com	information about your local distributor.	
Technical Support:	+49 (0) 37 65 / 38 600-0 support@systec-electronic.com	info@systec-electronic.com	
Fax:	+49 (0) 37 65 / 38 600 4100		
Web Site:	https://www.systec-electronic.com		

Table of Contents

1	Introduction	8			
2	Overview	9			
2	Safety Guidelines 11				
4		40			
4	Intended use	12			
5	Product Description	13			
6	Interface of the Device	16			
	6.1 Pin Assignment	16			
	6.2 User Interface	18			
	6.4 Power Supply	19			
	6.5 In- and Outputs for Industrial Standard Signals	19			
	6.5.1 Digital Inputs	20			
	6.5.2 Digital Outputs	21			
	6.5.2.1 Alternate Function: DO2, DO3: Pulse outputs P0 and P1	21			
	6.6.1 Serial Interfaces	22			
	6.6.2 CAN Interfaces	23			
	6.6.3 Ethernet Interfaces	23			
	6.6.4 USB-Host	23			
7	Configuration and Administration	24			
	7.1 System Requirements and Necessary Software Tools	24			
	7.2 Connection Over the Serial Interface	24			
	7.3 Activation/Deactivation of Linux Auto-Boot	25			
	7.4 Ethernet Configuration	26			
	7 4 2 DHCP Configuration	20			
	7.4.3 Static IP Address Configuration	26			
	7.5 Setting Up a USB Wi-Fi Adapter	27			
	7.6 PLC configuration	28			
	7.6.1 OpenPCS License Key Handling	28			
	7.6.2 PLC configuration via WED-Fromend	20			
	7.7 Service Configuration and Boot Scripts	32			
	7.7.1 Extend Shell Script in /etc/rc.local	32			
	7.7.2 Add Custom Systemd Services	32			
	7.7.3 Configure Services	33			
	7.8 Predefined User Accounts	34			
	7.9 Remote Login to the Command Shell	34 34			
	7.9.2 Login to the SFTP Server	35			
	7.10 Adding and Deleting User Accounts	36			
	7.11 How to Change the Password for User Accounts	37			
	7.12 Setting the System Time and Time Zone	37			
	7.13 File Systems	38			
	7.14 Software installation and opdate	39			
	7.14.2 Install Debian GNU/Linux to a SD Card for update/recovery	41			
	7.14.3 Install/Update Debian GNU/Linux to EMMC	42			
	7.15 TeamViewer IoT Agent	43			
	7.16 qbee.io	43			
8	Node-RED Programming Environment	44			
	8.1 Running Node-RED	44			
	δ.∠ Creating a demo application	45			
9	Mono/C#	50			
_	© SYSTEC electronic AG 2022 L-2562e_01 Page 4	—			

9.1	Create a C# application	50
9.2	Run the application	52
10 Ja	ava	53
10.1	Create a Java Project	53
10.2	Run the Java application	56
10.3	Debug the Java application	57
11 Py	/thon	59
11.1	Create a Python application	59
11.2	Run the Python application	60
11.3	Debug the Python application	60
12 PI	LC Functionality	61
12.1	Starting the PLC Runtime	61
12.2	Programming the Device	61
12.3	Process Image of the Device	61
12.3	1 Local In- and Outputs	61
12.3	.2 Network Variables for CAN1	62
12.4	Communication Interfaces	62 62
12.4	2 CAN Interfaces	62
12.4	.3 Ethernet Interface	62
12.5	Specific Peripheral Interfaces	63
12.5	1 Pulse Outputs	63
12.6	Control and Display Elements	63
12.6	.1 Run/Stop Switch	63
12.6	.2 Run-LED (green)	63
12.6	.3 Error-LED (red)	64
12.7	Local Deletion of a PLC Program	65
12.0	1 CAN Interface CANO	66
12.8	.2 CAN Interface CAN1	66
40 D	to Evolution with Character Discover Image	70
13 D	Overview of the Shared Process Image	70
13.1	API of the Shared Process Image Client	73
13.3	Creating a User-Specific Client Application	77
13.4	Example for Using the Shared Process Image	78
14 Tı	roubleshooting	83
Appondi	v A: Eirmwara Eunation Scona	00
Append		04
Append		09
Append	IX C: Inird Party Software Components	91
Append	x D: Declaration of RoHS-Conformity 1	39
Index		40

List of Tables

Table 1: Overview of relevant manuals	9
Table 2: Pin assignment	16
Table 3: Description Switches	18
Table 4: Description LEDs	18
Table 5: Recommended cable lengths, types and wire cross section	20
Table 6: Serial interface to Linux device node path	23
Table 7: Configuration entries of the CFG file	31
Table 8: systemd services	33
Table 9: Predefined user accounts	34
Table 10: File system configuration	39
Table 11: Assignment of in- and outputs to the process image	61
Table 12: Allocation between impulse channels and outputs	63
Table 13: Display status of the Run-LED	64
Table 14: Display status of the Error-LED	64
Table 15: Representation of network variables for CAN1 by entries in the object dictionary	67
Table 16: Preconfigured PDOs for interface CAN1	68
Table 17: Content of the archive files "shpimgdemo.tar.gz"	77
Table 18: Firmware functions and function blocks	84

List of Figures

Figure 1: Top view of device	13
Figure 2: Interface and connector assignment	16
Figure 3: Galvanic isolation	19
Figure 4: Setup of digital inputs	20
Figure 5: Setup of digital outputs	22
Figure 6: Terminal configuration using the example of "TeraTerm"	25
Figure 7: Login screen TeraTerm	25
Figure 8: Example – get the IP address for eth0	26
Figure 9: User login dialog of the WEB-Frontend	29
Figure 10: PLC configuration via WEB-Frontend	30
Figure 11: SSH login PuTTY	34
Figure 12: Login to the device	35
Figure 13: Login settings for WinSCP	35
Figure 14: FTP client for Windows "WinSCP"	36
Figure 15: Changing the password for a user account	37
Figure 16: Setting and displaying the system time	38
Figure 17: Setting and displaying the system time	38
Figure 18: File transfer in SFTP client "WinSCP"	40
Figure 19: Installing PLC firmware	41
Figure 20: Write SD Card Image using Win32 Disk Imager	42
Figure 21: Node-RED editor in the web browser	45
Figure 22: Node-RED demo application	46
Figure 23: Node: "trigger"	46
Figure 24: Node: "8-bit shifter"	47
Figure 25: Node: "8-bit mask"	48
Figure 26: Node: "DO0"	49
Figure 27: Visual Studio console application	50
Figure 28: Java Demo: Project setup	53
Figure 29: Java Demo: Main class	54
Figure 30: Java Demo: Project structure	54
Figure 31: Java Demo: Run configuration	56
Figure 32: Java Demo: Export Java JAR archive	56
Figure 33: Java Demo: Debug configuration	57
Figure 34: Java Demo: Debug example	58
Figure 35: Positioning of network variables for CAN1 within the marker section	67
Figure 36: Overview of the shared process image	70
Figure 37: Generating the demo project "shpimgdemo" in the Linux development system	78
Figure 38: Terminal outputs of the demo program "shpimgdemo" after start	81
Figure 39: Terminal outputs of the demo program "shpimgdemo" after user inputs	82

1 Introduction

Thank you for choosing the SYS TEC sysWORXX CTR-500. This product provides to you an innovative, Linux-based and high-capacity compact controller to process standard industrial signals. Due to its numerous in- and outputs and communication interfaces it is well-suitable as central control in distributed automation appliances.

Please take some time to read through this manual carefully. It contains important information about the commissioning, configuration and programming of this device. It will assist you in getting familiar with the functional range and usage. While there may be some instances, where another device is mentioned, the relevant paragraph is still applicable to the sysworxx CTR-500. This document is complemented by other manuals, e.g for the OpenPCS IEC 61131 programming system. Table 1 in section 2 lists relevant manuals for this device. This table also references documentation to other software components and programming languages which are supported, such as Node-RED, C# or Java. Please also refer to those complementary documents.

For more information, optional products, updates et cetera, we recommend you to visit our website: <u>https://www.systec-electronic.com</u>. The content of this website is updated periodically and provides you with downloads of the latest software releases and manual versions.

2 Overview

Table 1 lists up all relevant manuals for the device.

Table 1: Overview of relevant manuals

Information about	In which manual?	
Basic information about the device (configuration, administration, process image, connection assignment, firmware update, reference designs et cetera)	In this manual	
Basics about the <i>OpenPCS</i> IEC 61131 programming system	Brief instructions for the programming system (Entry " <i>OpenPCS Dokumentation</i> " in the <i>OpenPCS</i> program group of the start menu) (Manual no.: L-1005)	
Complete description about the <i>OpenPCS</i> IEC 61131 programming system, basics of PLC programming according to IEC 61131-3	Online help about the <i>OpenPCS</i> programming system	
Command overview and description of standard function blocks according to IEC 61131-3	Online help about the <i>OpenPCS</i> programming system	
SYS TEC extension for IEC 61131-3: - String functions - UDP function blocks - SIO function blocks - FB for RTC, Counter, EEPROM, PWM/PTO	User Manual "SYS TEC-specific extensions for OpenPCS / IEC 61131-3" (Manual no.: L-1054)	
<i>CANopen</i> extension for IEC 61131-3 (network variables, <i>CANopen</i> function blocks)	User Manual "CANopen extension for IEC 61131- 3" (Manual no.: L-1008)	
Textbook about PLC programming according to IEC 61131-3	IEC 61131-3: Programming Industrial Automation Systems John/Tiegelkamp Springer-Verlag ISBN: 3-540-67752-6 (a short version is available as PDF on the <i>OpenPCS</i> installation CD)	

One can also use Node-RED, Java or C# to program the device. The following table provides links to the websites of those projects. On these sites you will also find guides and references on how to use these programming environments for developing applications. Getting started with these environments is described in this document.

Project	Website
Node-RED	https://nodered.org/
Mono / C#	http://www.mono-project.com/
OpenJDK / Java	http://openjdk.java.net/

Section 6	Describes electric connections of the product and their application; moreover, it documents their internal structure .
Section 7	Makes available details about the configuration , e.g. the configuration of Ethernet and CAN interfaces, the configuration of Linux services and the selection of the firmware version. In addition, the administration is explained, e.g. the login to the system, the user administration and the execution of software updates.
Section 8	Provides basic information on how to enable Node-RED and how to use the custom nodes to access digital inputs and outputs, or to access OpenPCS variables.
Section 9	Provides information on how to develop and debug Mono/C# applications in form of a step-by-step guide of a sample project, which uses the I/O driver for the device.
Section 10	Provides information on how to develop and debug Java applications in form of a step- by-step guide of a sample project, which uses the I/O driver.
Section 11	Provides information on how to develop and debug Python applications in form of a step-by-step guide of a sample project, which uses the I/O driver.
Section 12	Includes details about the usage of the device , e.g. the setup of the process image , the meaning of control elements and this section provides basic information about programming the module. Furthermore, it gives information about the usage of CAN interfaces in combination with CANopen .
Section 13	Covers information about data exchange between a PLC program and a user-specific C/C++ application via shared process image .

3 Safety Guidelines

This product may only be operated by personnel qualified for the specific task in accordance with the corresponding documentation for this specific task. This relates in particular to its warning notices and safety instructions. Qualified personnel are those who, when working with these products, are capable of identifying risks and avoiding potential hazards.

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. These notices shown below are graded according to the degree of danger.



Danger signs warn for potential life-threatening situations. There may be occur damage not only to the device and its surroundings, but also to the operating personnel. **Do under no circumstance ignore these warnings!**



WARNING

DANGER

Warnings show, when certain situation could permanently damage the device or its surroundings. Make sure to always follow the instruction, to not harm any parts of the hardware.

NOTICE

Notices describe, which things could inflict potentially harmful things to the device. There is no imminent danger to personnel or the surrounding hardware, but the functionality of the device could be permanently impaired.

4 Intended use

The device is designed for use in the industrial sector (EN 61000-6-2:2005 & EN 61000-6-4:2007 + A1:2011) only.

NOTICE

The approvals are voided, if certain modifications are made

- The device was opened.
- The device was physically modified, for example, additional openings were created.
- Cables are routed from the inside out of the device or from the outside into the device, for example, to connect additional peripherals.
- The specified cable lengths for the interfaces must not be exceeded.

Under certain conditions one can use the device for residential purposes and for commercial operations, if that does not significantly impact the residential purpose. In that case, one must comply with the radio frequency interference limits of the generic standard EN 61000-6-3.

Please contact us, if there is the need to use the device in any condition or sectors, which are not specified in this document.

5 Product Description

The sysWORXX CTR-500 extends the SYS TEC electronic AG product range within the field of IoT and control applications. It is an innovative, Linux-based compact controller for universal processing purposes of standard industrial signals. The controller module provides to the user numerous local inand outputs as well as versatile communication interfaces. Due to CAN and Ethernet interfaces, it is suited for realizing decentral control tasks in distributed fieldbus systems of automation technology. With all these tools, one is able to write their own applications for their specific needs and purposes.



Figure 1: Top view of device

These are some significant features of device:

- Linux-based compact PLC for industrial controls
- High-capacity CPU kernel (Freescale i.MX7 series Dual ARM Cortex-A7 Core 1GHz, Real-time Core Cortex-M4 200MHz)
- Up to 1024 MiB RAM, 8GiB eMMC FLASH Memory
- 1x USB 2.0 Host interface
- Up to 2x 10/100 Mbps Ethernet LAN interface
- 1x WLAN (802.11 ac/a/b/g/n)
- 1x integrated Meshnet module
- 1x CAN 2.0B interface, usable as CANopen Manager (CiA 302-conform)
- Up to 2x asynchronous serial ports (UART), usable as RS-232 or RS-485 (switchable via software)
- Up to 4 digital inputs 24VDC, galvanic isolated
 - Alternate function: 1 high-speed counter input, galvanic isolated
 - Alternate function: 1 A/B-Encoder
- Up to 4 digital outputs 24VDC/500mA, galvanic isolated, short-circuit-proof
 - Alternate function: 2 PWM/PTO¹ outputs 24VDC/500mA 1KHz
- RTC (with buffer capacitor)
- 2 temperature sensors, CPU and System temperature
- On-board software: Linux, PLC firmware with CANopen Master, Node-RED, HTTP and SFTP server
- Programming with C/C++, C#, Java and Python
- Linux-based (other user programs are executable in parallel)
- Remote login via SSH

¹ PTO functionality is not implemented yet. It will be available in a future release. The PWM functionality is already available.

- Dimensions: 105 x 91 x 60mm
- Temperature 0° ... 55°C
- Suitable for DIN top hat rail mounting
- Future features:
 - Programmable according to IEC 61131-3
 - Easy, HTML-based OpenPCS configuration via web browser
 - Functions for the PLC runtime:
 - Function block libraries for communication (CANopen, Ethernet and UART)
 - Function block libraries for hardware components (RTC, Counter, PWM/PTO)

The default hardware comes with a Linux operating system. This base system can be used to program in different programming languages and also provides the Node-RED programming environment. To identify each individual device, the manufacturer, revision, serial and order number are be printed on a label on the right side of the device. Some additional hardware and software components are also available:

Feature:	19031000	19032000	19033000
	Full Version	Sink Node Version	Low Cost IO Version
Description	The basic Debian/GNU	The basic Debian/GNU	The basic Debian/GNU
	Linux installation and	Linux installation,	Linux installation,
	the default hardware	mesh net capabilities	without mesh net
		and without digital I/O's	capabilities and
		and overall, less	overall, less interfaces
		interfaces	
Digital Inputs	4	0	4
Digital Outputs	4	0	4
Meshnet	Yes	Yes	No
WLAN	Yes, certified 802.11	Yes, certified 802.11	Yes, certified 802.11
	ac/a/b/g/n	ac/a/b/g/n	ac/a/b/g/n
CAN	1	No	1
Ethernet	2	1	1
RS-232/485	2	1	1
Preinstalled Software	 Node.js 		
	Node-RED		
	Python		
	OpenJDK		
	Mono		
	Future feature: 0	OpenPCS (needs a separa	ate commercial licence)

Additionally available licenses to purchase: Order no.: **240011**: Runtime license OpenPCS RT sysWORXX CTR-500

Order no.: 240012:

Runtime license OPC-UA basis server sysWORXX CTR-500

This device is an all-round PLC for complex industrial control tasks. As Linux-based compact controller, the module is programmable in C/C++, C#, Java and Python and in IEC 61131-3. Also, the Node-RED node editor is available. All of these options allow highly efficient software development for this module.

There are specific nodes for applications in Node-RED, which are able to control different aspects of the device, e.g. nodes to control the different digital outputs, read the digital/analog inputs and also to read or write variables for a running PLC program. The nodes are already installed on the device, but can also be found at GitHub (<u>https://github.com/systec-electronic</u>) and NPM (<u>https://www.npmjs.com/package/node-red-contrib-ctr700-io</u>, <u>https://www.npmjs.com/package/node-red-contrib-ctr700-io</u>,

The on-board firmware of contains the entire PLC runtime environment including CANopen connection with CANopen Master Functionality. Thus, the module is able to operate control tasks such as linking

© SYSTEC electronic AG 2022	L-2562e_01	Page 14
-----------------------------	------------	---------

in- and outputs or converting rule algorithms. Data and occurrences can be exchanged with other nodes (e.g. superior main controller, I/O slaves and so forth) via CANopen network, Ethernet (UDP protocol) and serial interfaces (UART). The numerous in- and outputs that the module provides can be decentrally extended by CANopen devices. CANopen IO modules of *sysWORXX Automation Series* are well-suited for this. Those modules are also designed for processing industrial standard signals (24VDC, 0-10VDC, 0-20mA etc.).

Programming the device takes place according to IEC 61131-3 using the programming system *OpenPCS* of the company infoteam Software GmbH (<u>http://www.infoteam.de</u>). This programming system has been extended and adjusted for this device by the company SYS TEC electronic AG. Hence, it is possible to program the device graphically in KOP/FUB, AS and CFC as well as textually in AWL or ST. Downloading the PLC program onto the module takes place via Ethernet or CANopen – depending on the firmware configuration. Addressing in- and outputs and creating a process image follows the SYS TEC scheme for compact control units. Hence, PLC programs developed by the user can be operated on different SYS TEC control modules without adjustments. Like all other SYS TEC controls, this device supports backward documentation of the PLC program as well as the debug functionality including watching and setting variables, single cycles, breakpoints and single steps.

This device uses Debian GNU/Linux as operating system. This allows for an execution of other userspecific programs while PLC firmware is running. If necessary, those other user-specific programs may interchange data with the PLC program via the process image or the Node-RED nodes for reading and writing of variable values. More information about this is provided in section 8 and 13.

The Linux applied is licensed under GNU General Public License, version 2. **Appendix C** contains the license text. All sources of Linux BSP are included in the software package 3912005 ("Oracle VM VirtualBox-Image of the Linux development system"). If you require the Linux BSP sources independently from the Oracle VirtualBox-Image of the Linux development system, please contact our support:

support@systec-electronic.com

The PLC system and the PLC-, C# and C/C++ programs developed by the user are **not** subject to GNU General Public License!

When using the pre-installed software, the user must comply with the provisions of the respective licenses.

6 Interface of the Device

6.1 Pin Assignment



Figure 2: Interface and connector assignment

Figure 2 shows the positioning of connectors on the device as an overview. Table 2 lists all connectors in detail.

Terminal	Port	Signal name	Remark	
Digital Outputs	1	GND0	241/	
X1	2 9	DO0 DO7	240	
Digital Inputs	1	GND1	241/	
X5	2 9	DI0 DI7	241	
µSD-Card-Holder				
X9	-	-	-	
µUSB (console)				
X10	-	-	-	
Power	1	PE		
	2	GND	24V	
X11	3	24VDC		
USB-Host	-	-	-	

X12			
Ethernet 0			
X13	-	-	-
Ethernet 1			
X14	-	-	-
	1	RX	
	2	-	BS 222
	3	ТХ	R5-232
Serial Interface 0	4	GND	
X15	1	A/D0	
-	2	B/D1	DS 495/Modbus DTU
	3	-	K3-485/MOUDUS KTO
	4	GND	
	1	RX	
	2	-	RS-232
	3	ТХ	10-202
Serial Interface 1	4	GND	
X16	1	A/D0	
	2	B/D1	RS-485/Modbus RTU
	3	-	
	4	GND	
CAN 0	1	HIGH	
X4.0	2	LOW	-
X18	3	GND_CAN0	
Meshnet antenna	_	_	_
X600			

6.2 User Interface

Switch	Port	OFF	ON
S1	-	-	RESET
62		Position = left (see Figure 2)	Position = right (see Figure 2)
32	-	PLC stop (outputs deactivated)	PLC run (program re-starts)
		SERIAL0:	SERIAL0:
	1	RS-485: Bus termination off	RS-485: Bus termination on
		RS-232: Must be set!	RS-232: Do not set!
		SERIAL1:	SERIAL1:
S3 ²	2	RS-485: Bus termination off	RS-485: Bus termination on
		RS-232: Must be set!	RS-232: Do not set!
	3	Configuration: ON	Configuration: OFF
	4	Normal Booting	Boot in U-Boot Bootloader CLI
	5	Boot mode: EMMC	Boot mode: SD-Card
	1	CAN0:	CAN0:
S4	I	Termination off	Termination on
	2	Reserved	Reserved

Table 3: Description Switches

Table 4: Description LEDs

LED	Color	Feature
D01	Green	Status of the power supply for the DO's
D02 D05	Yellow	Signal status of the DO's (on = high; D02 = DO0, D03 = DO1,)
D06	Green	Status of the power supply for the DI's
D07 D10	Yellow	Signal status of the DI's (on = high; D19 = DI0, D20 = DI1,)
D11	Green	Status of the PLC: RUN f=1 Hz; STOPP f=0.5 Hz
D12	Red	ERROR-LED, signals an occurring (PLC or IO-driver) error
D13	Green	Status of the power supply for the device

6.3 Mounting

NOTICE

Mounting

This device is intended to be mounted on DIN rail. It is designed to be attached horizontally only (orientation does not matter). Mounting it vertically may disturb the air flow and can cause overheating of the internal components.

NOTICE

Limitation of liability

Technical specifications and approvals of this device only apply, if expansion components with a valid CE approval are used.

Also, the installation conditions for expansion components, described in the associated documentation, must be followed.

SYS TEC electronic is not liable for functional limitations caused by the use of third-party devices or components.

² The switches for the serial interfaces are in reversed order (2, 1, 0) on all devices labeled as prototype.

6.4 Power Supply

The device features three power supply inputs (24VDC \pm 20%) for CPU unit and two peripheries.

The connector supplies the CPU unit, the digital in- and outputs. This input has reverse polarity protection.

If the device experiences a power fail (< 14,9V), there is a 10 ms time window to act accordingly, for example to safe your data. During this 10 ms, the device will work normally then shuts down. To access the power fail signal, one has to use the driver library in a separate program (see Sections 8, 9, 10, 11).



Galvanic Isolation

Figure 3 shows the galvanic isolation of the different interfaces and system components.



Figure 3: Galvanic isolation

6.5 In- and Outputs for Industrial Standard Signals

NOTICE
NOTICE
Limitation of liability
Technical specifications and approvals of this device only apply, if expansion components with a valid CE approval are used.
Also, the installation conditions for expansion components, described in the associated documentation, must be followed.
SYS TEC electronic is not liable for functional limitations caused by the use of third-party devices or components.

The following table shows, which cable types are recommended for the different interfaces:

Interface	Cable length	Recommended cable types	Wire cross section
Digital outputs	<30m	Any cable suitable to the specific usage.	0,2 - 1,5mm² or AWG24 - 16
Digital inputs	<30m		
CAN	<30m		
Serial	RS232: <15m Modbus RTU: <30m		
Ethernet	<30m	F/FTP S/FTP SF/FTP F/STP S/STP SF/STP	-
USB	<3m	USB standard cable	-

Table 5: Recommended cable lengths, types and wire cross section

6.5.1 Digital Inputs

The device features four digital inputs (DI0 ... DI3). Each input has the same supply potential. The inputs are high active with the following switching threshold:

- Input voltage > 15 VDC: is shown as '1' in the process image
- Input voltage < 5 VDC: is shown as '0' in the process image

Digital inputs have the internal structure as shown in Figure 4.



Figure 4: Setup of digital inputs

The digital inputs in a PLC program are accessible via the process image (see Table 11 in section 12.3.1).

6.5.2 Digital Outputs

The device features four digital high-side switch outputs (DO0 ... DO3). The outputs each connect the supply voltage Vcc of the appliance (switching positively). The maximum load current for each 24V output is 0.5A for ohmic, inductive or capacitive load. The outputs are short-circuit-proof and galvanically isolated from the CPU unit. The outputs are protected against excess voltage, reverse polarity and excess temperature. The outputs are accessed high active:

- '1' in process image: output transistor active, appliance connected with Vcc
- '0' in process image: output transistor inactive, appliance disconnected from Vcc

The digital transistor outputs have the internal structure as shown in Figure 5. At and after powerup and reset the outputs are in off state.

6.5.2.1 Alternate Function: DO2, DO3: Pulse outputs P0 and P1

The device features two Pulse outputs (P0 and P1) to output PWM and PTO signal sequences. The maximum load current for each 24V output is 0.5 A for ohmic, inductive or capacitive load. The outputs are short-circuit-proof and galvanically isolated from the CPU unit. The performance drivers used are protected against excess voltage, reverse polarity and excess temperature. The transistor outputs are also activated low active.

Pulse outputs P0 and P1 use the same internal structure as the digital outputs DI14/DI15 shown in Figure 5.

In a PLC program, the PWM/PTO functionality of Pulse outputs is accessible via function block "PTO_PWM" (see manual "SYS TEC-specific extensions for OpenPCS / IEC 61131-3", manual no.: L-1054).



Figure 5: Setup of digital outputs

The digital outputs in a PLC program are accessible via the process image (see Table 11 in section 12.3.1).

6.6 Communication Interfaces

6.6.1 Serial Interfaces

This device features one service and two serial interfaces (X15 and X16).

SERVICE (X10)

Interface Service serves as service interface to administer the device. The connection to a computer is established via Micro-USB.

The maximum cable length for the SERVICE interface is 3m.

SERIAL0 and SERIAL1 (X15 and X16)

Interface Serial 0 can be used for data exchange under control of the PLC program. RS-232 signals Rx, Tx and GND or RS-485 signals D0, D1 and COM are available at a X15, X16 connector. Additionally, the configuration for the RS-485 has to be added to the source code of an application. When used as RS-485, termination resistors can be activated via the DIP-Switch (see Table 3: Description Switches). The termination should only be used, if the device is the last in line. By default, the interface uses RS-232 signals.

Setting up a serial interface for RS-485

RS-485 is available for the serial interfaces **SERIAL0** and **SERIAL1**. To set up an interface for RS-485, the following configuration has to be executed. Substitute **INTERFACE** with the targeted interface (see Table 6):

```
#include <stdio.h>
#include <string.h>
#include <strings.h>
#include <unistd.h>
#include <errno.h>
#include <stdlib.h>
#include <fcntl.h>
#include <termios.h>
#include <sys/select.h>
#include <sys/ioctl.h>
#include <linux/serial.h>
int main() {
    int iInterface = open("INTERFACE", O RDWR | O SYNC);
    struct serial rs485 RS485;
    ioctl(iInterface, TIOCGRS485, &RS485);
    RS485.flags |= SER RS485 ENABLED;
    RS485.flags &= ~SER RS485 RX DURING TX;
    ioctl(iInterface, TIOCSRS485, &RS485);
    close(iInterface);
}
```

This code has to be put into a C-file, compiled and can then be executed. **Hint**: After the device is restarted, this configuration has to be repeated!

© SYSTEC electronic AG 2022	L-2562e_01	Page 22
-----------------------------	------------	---------

Table 6: Serial interface to Linux device node path

Interface	Linux-Device (INTERFACE)
SERIAL0	/dev/ttymxc6
SERIAL1	/dev/ttymxc1
SERVICE ³	/dev/ttymxc0

6.6.2 CAN Interfaces

The device features two CAN interfaces (CAN0 and CAN1). Those two CAN-Bus-Transceivers are galvanically isolated to one another and to the CPU. The transceivers are supplied via two on-board DC/DC converter. CAN-Bus signals CAN0 HIGH, CAN0 LOW, CAN1 HIGH, CAN1 LOW and CAN0 GND/CAN1 GND are available from withdrawable terminal-block connectors.

Section 12.8 provides detailed information about the usage of both CAN interfaces in connection with CANopen.

CAN cable: The CAN-Bus usually is a twisted pair line. At both ends of the cable, a termination resistor of **120 Ohm termination** is necessary **between CAN_H and CAN_L**. CiA (CAN in Automation) must use CAN-GND in CiA DRP 303-1. For more information please refer to the appropriate CiA standards.

Both CAN interfaces also support the use of an internal termination resistor for one end of the CAN bus. The DIP-switch to make use of these resistors is described in Table 3.

6.6.3 Ethernet Interfaces

The device features up to two Ethernet interfaces (ETH0 and ETH1) which are designed as 10Base-T/100Base-TX.

The Ethernet interface serves as service interface to administer the device and it can be used for data exchange with any other devices.

6.6.4 USB-Host

The device features a USB 2.0 host interface (X12).

The maximum cable length for the USB-host interface is 3m.

³ The **SERVICE** interface is used as the default Linux console for serial access to the device. Do not use this for custom applications unless you really know how to handle this without any conflicts.

7 Configuration and Administration

7.1 System Requirements and Necessary Software Tools

The administration requires any Windows or Linux computer that has available an Ethernet or USB interface. These allow a connection to administer the device via a Linux command line-interface.

All examples referred to in this manual are based on an administration using a Windows computer. Procedures using a Linux computer would be analogous.

To administrate the device the following software tools are necessary:

Terminal program A Terminal program allows the communication with the **command shell** via **the integrated USB-to-UART bridge (USB service console) of the** device. This is required for the Ethernet configuration as described in section 7.4. After completing the Ethernet configuration, all further commands can either be entered in the Terminal program or alternatively in an SSH client (see below).

A suitable Terminal program would be "*TeraTerm*", which is available as Open Source Software (*BSD* License). The project page is located at: <u>http://ttssh2.osdn.jp/</u>.

Secure Shell (SSH) SSH allows the encrypted communication with command shell via Ethernet. Using SSH requires a completed Ethernet configuration of the device according to section 7.4. As alternative solution to SSH, all commands can be used via a Terminal program.

> Suitable as SSH client would be "*PuTTY*" or "*TeraTerm*", which can also be used as Terminal program (see above). "*PuTTY*" is licensed under *MIT*-License and can be downloaded at: <u>https://www.chiark.greenend.org.uk/~sgtatham/putty/</u>.

SFTP client An SFTP client allows file transfer between the device and the computer. This allows for example editing configuration files by transferring those onto the computer where they can be edited and get transferred back to the device. Downloading files is also necessary to update the PLC firmware. (Advice: The update of *PLC firmware* is not identical with the update of the *PLC user program*. The PLC program is directly transferred to the module from the *OpenPCS* programming environment. No additional software is needed for that.)

Suitable as SFTP client would be "*WinSCP*" which is available as Open Source Software (GNU GPL).

Page 24

For programs that communicate via Ethernet interface, such as SFTP client or TFTP server, it must be paid attention to that rights in the Windows-Firewall are released. Usually Firewalls signal when a program seeks access to the network and asks if this access should be permitted or denied. In this case access is to be permitted.

7.2 Connection Over the Serial Interface

Setting up a connection over the serial interface needs a Terminal Program like *"TeraTerm"* (see 7.1) and the *"Silicon Labs USB to UART Bridge"* driver installed on the computer. The driver can be found here: <u>https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers</u>

Additionally, the USB host interface (SERVICE, see section 6) of the device has to be connected to the computer with a μ USB cable.

© SYSTEC electronic AG 2022 L-2562e_01
--

If both are installed and the device is connected to the computer, "*TeraTerm*" must be started and configured as follows (see Figure 6):

- 115200 Baud
- 8 Data bit
- 1 Stop bit
- no parity
- no flow control

Note: The Port has to be the COM-interface installed with the Silicon Labs driver. This is depending on the computer on which it was installed. COM3 as shown in Figure 6 is only an example, the real number of the COM interface can vary on other computers.

Tera Term: Serial port set	up	×		
Port:	COM3 ~	ОК		
<u>B</u> aud rate:	115200 ~			
<u>D</u> ata:	8 bit \sim	Cancel		
P <u>a</u> rity:	none ~			
<u>S</u> top:	1 bit \sim	<u>H</u> elp		
<u>F</u> low control:	none ~			
Transmit delay O msec <u>/c</u> har O msec <u>/l</u> ine				

Figure 6: Terminal configuration using the example of "TeraTerm"

Clicking on OK will start the command shell. After pressing any key, the login screen should be visible and the user is able to interact with the device (see Figure 7).



Figure 7: Login screen TeraTerm

For the login you need a valid user account. There are predefined and ready to use accounts already available (see section 7.8). In this example, the user *"root"* was used.

7.3 Activation/Deactivation of Linux Auto-Boot

During standard operation mode, the bootloader "U-Boot" automatically starts the Linux operating system of the module after a reset (or power-on reset). Afterwards, the operating system loads all enabled services such as OpenPCS runtime or Node-RED (see section 7.7.3 on how to enable system services). For certain service tasks it may be required to access the "U-Boot" command prompt instead.

© SYSTEC electronic AG 2022	L-2562e_01	Page 25
-----------------------------	------------	---------

Communicating with the bootloader "U-Boot" only takes place via the serial interface "Service". To disable the auto-boot, the corresponding DIP-Switch has to be set (see Table 3). After the switch is set and the system is rebooted, the "U-Boot" command prompt is activated.

7.4 Ethernet Configuration

The device has up to two ethernet interfaces ETH0 and ETH1.The main ethernet configurations are saved in the configuration file in */etc/network/interfaces*. By default, only ETH0 is used and configured to use DHCP and the interface ETH1 has no configuration. The following configuration examples below use "ethX" as placeholder. Substitute "ethX" with the targeted network interface *eth0* or *eth1*. Modifications are adopted upon the next reboot.

Advice: After the configuration is finished, the serial connection between PC and the device is no longer necessary.

7.4.1 Get the Current IP Address

To get the current IP addresses of the device, one has to set up a connection with a Terminal program (see 7.1). After login one can use the following command, to get a list of IP addresses:

ifconfig eth0

The parameter eth0 is optional. If not given, ifconfig will print the IP addresses of all interfaces.

The following example shows the IP address 192.168.10.134 for the network interface ethO:

🚾 COM3 - Tera Term VT	-		\times
D <u>a</u> tei B <u>e</u> arbeiten Ein <u>s</u> tellungen Ste <u>u</u> erung <u>F</u> enster <u>H</u> ilfe			
<pre>root@ctr-700:-# ifconfig eth0 eth0 Link encap:Ethernet HWaddr f8:dc:7a:14:d1:a0 inet addr:192.168.10.134 Bcast:192.168.10.255 Mask inet6 addr: fe80::fadc:7aff:fe14:d1a0/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:5202 errors:0 dropped:0 overruns:0 frame: TX packets:57 errors:0 dropped:0 overruns:0 carrier: collisions:0 txqueuelen:1000 RX bytes:394356 (385.1 KiB) TX bytes:4632 (4.5 KiB)</pre>	:255.25 0 0	5.255.	0
root@ctr-700:~#			

Figure 8: Example - get the IP address for eth0

7.4.2 DHCP Configuration

Add the configuration options listed below to the configuration file, to change the network interface to DHCP.

allow-hotplug ethX iface ethX inet dhcp

7.4.3 Static IP Address Configuration

Add the configuration options listed below to the configuration file, to change the network interface to static. Use the proper configuration for your network infrastructure.

```
allow-hotplug ethX
iface ethX inet static
address 192.168.0.100
netmask 255.255.255.0
```

```
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

7.5 Setting Up a USB Wi-Fi Adapter

The device is capable of using a wide range of USB Wi-Fi Adapters. There are several vendors and products available. This section will describe the steps necessary to setup of a Wi-Fi adapter based on the RTL8192CU chip. Check the Linux kernel configuration, which Wi-Fi adapters are supported.

Open a terminal session via serial service interface or SSH and plug the adapter to the USB-Host plug. Check with *Isusb* if the Wi-Fi adapter is the right one. In this case it must be the *RTL8192CU 802.11n WLAN Adapter*. If this is the case, install the *wpasupplicant* and *firmware-realtek* packages.

apt install wpasupplicant firmware-realtek

Setup the SSID and PSK of the wireless network by issuing the following command. (replace the placeholders <SSID> and <PSK> to your network configuration)

wpa passphrase <SSID> <PSK> > /etc/wpa supplicant.conf

Add the following lines to the end of /etc/wpa_supplicant.conf.

```
update_config=1
ctrl interface=DIR=/var/run/wpa supplicant
```

Put the rtl8192cu module on the blacklist, to force the kernel to use another driver module. Otherwise the kernel will mistakenly try to use this one, which will not work as expected.

echo "blacklist rtl8192cu" > /etc/modprobe.d/rtl8192cu-blacklist.conf

Restart Wi-Fi adapter by replacing the interface <WLAN> to your Wi-Fi interface name.

ip link set <WLAN> down
ip link set <WLAN> up

Start the wpa_supplicant with Wi-Fi interface in background and run the DHCP client.

```
wpa_supplicant -i <WLAN> -B -c /etc/wpa_supplicant.conf
dhclient <WLAN>
```

Now the network / internet should be working smoothly. This can be tested by pinging to an IP address in the internet or your local network.

ping <SOMEIPADDRESS>

To automatically connect to the network after booting, add the following lines to the file /*etc/network/interfaces* and replace <WLAN>, <SSID> and <PSK> with the correct settings.

```
auto <WLAN>
iface <WLAN> inet dhcp
wpa-ssid <SSID>
wpa-psk <PSK>
```

7.6 PLC configuration

7.6.1 OpenPCS License Key Handling

For the usage of the PLC functions, the device needs an installed OpenPCS license key. If ordered that way, the key is normally pre-installed. To check, if a key is installed, the following command can be used:

cat /vendor/device

This command outputs not only the license key, but also other device and vendor information, such as the device specific serial number. If the line for the "LicKey" is empty, no license key is installed.

The key can also be installed, after purchase. For that, please contact us and we provide you with the license key for OpenPCS. With the following command, it can be installed (or deleted):

vedor setup vendor

During the following setup, the device specific serial number and the license key has to be added. After that, reboot the device. The service for OpenPCS can now be used as described below.

7.6.2 PLC configuration via WEB-Frontend

After finishing the Ethernet configuration (see section 7.4), all further adjustments can take place via the integrated WEB-Frontend. The frontend service is disabled by default. To enable it, run the following command as described in section 7.7.3:

systemctl enable openpcs-lighttpd

To configure the device via WEB-Frontend it needs a WEB-Browser on the PC (e.g. Microsoft Internet Explorer, Mozilla Firefox, etc.). To call the configuration page, prefix "*http://*" must be entered into the address bar of the WEB-Browser prior to entering the IP address of the device as set in section 7.4, e.g. "*http://192.168.10.193*". Figure 9 exemplifies calling the CTR-700 configuration page in the WEB-Browser.

The standard setting (factory setting) requires a user login to configure the device via WEB-Frontend. This is to prevent unauthorized access. Therefore, user name and password must be entered (see Figure 9). On delivery of the module, the following user account is preconfigured (see section 7.8):

User: PlcAdmin Password: Plc123



Figure 9: User login dialog of the WEB-Frontend

All configuration adjustments for the device are based on dialogs. They are adopted into the file "/home/plc/bin/<device>.cfg" (e.g. "/home/plc/bin/ctr-700.cfg") by activating the pushbutton "Save Configuration" (also compare section 7.6.3). After activating Reset starts automatically using the active configuration. Figure 10 shows the configuration via WEB-Frontend.



Figure 10: PLC configuration via WEB-Frontend

The standard setting (factory setting) requires a user login to access the WEB-Frontend. Therefore, only the user name indicated in configuration file "/home/plc/bin/<device>.cfg" (e.g. "/home/plc/bin/ctr-700.cfg") is valid (entry "User=" in section "[Login]", see section 7.6.3). Procedures to modify the user login password are described in section 7.11. To allow module configuration to another user, an appropriate user account is to be opened as described in section 7.10. Afterwards, the new user name must be entered into the configuration file "/home/plc/bin/<device>.cfg" (e.g. "/home/plc/bin/ctr-700.cfg"). Limiting the user login to one user account is cancelled by deleting the entry "User=" in section "[Login]" (see 7.6.3). Thus, any user account may be used to configure the module. By deactivating control box "This configuration requires a Login" in the field "User Authorization" of the configuration page (see Figure 10) free access to the module configuration is made available without previous user login.

7.6.3 Setup of the Configuration File

The configuration file "/home/plc/bin/<device>.cfg" allows for comprehensive configuration of the device. Although, working in it manually does not always make sense, because most of the adjustments may easily be edited via WEB-Frontend (compare section 7.6.1). The setup of the configuration file is similar to the file format "Windows INI-File". It is divided into "[Sections]" which include different entries "Entry=". Table 7 lists all configuration entries.

Table 7: Configuration er	ntries of the CFG file
---------------------------	------------------------

Section	Entry	Value	Meaning
[CAN0]	Enabled	0, 1	0: Interface CAN0 is deactivated
			1: Interface CAN0 is activated, configuration takes place via entries of the configuration file below
	NodelD	1 127 or 0x01 0x7F	Node number for interface CAN0 (decimal or hexadecimal with prefix "0x")
	Bitrate in Kbit/s	10, 20, 50, 125, 250, 500, 800, 1000	Bitrate for interface CAN0
	MasterMode	0, 1	1: Master mode is activated
			0: Master mode is deactivated
[CAN1]	Enabled	0, 1	0: Interface CAN1 is deactivated
			1: Interface CAN1 is activated, configuration takes place via entries of the configuration file below
	NodelD	1 127 or 0x01 0x7F	Node number for interface CAN1 (decimal or hexadecimal with prefix "0x")
	Bitrate in Kbit/s	10, 20, 50, 125, 250, 500, 800, 1000	Bitrate for interface CAN1
	MasterMode	0, 1	1: Master mode is activated
			0: Master mode is deactivated
[ETH0]	PortNum	Default Port no: 8888	Port number for the communication with the Programming-PC and for program download
[ProcImg]	EnableSharing	0, 1	0: No sharing of process image
			1: Sharing of process image is enabled
			(see section 13)
[Login]	Authorization	0, 1	0: Configuration via WEB- Frontend is possible without user login
			1: Configuration via WEB- Frontend requires user login
	User	Default Name: PlcAdmin	If entry "User=" is available, only the user name defined is accepted for the login to configure via WEB-Frontend.
			If the entry is not available, any user registered on the device (see section 7.10) may login via WEB-Frontend.

The configuration file "/home/plc/bin/<device>.cfg" includes the following factory settings:

```
[Login]
Authorization=1
User=PlcAdmin
[CAN0]
Enabled=1
NodeID=0x20
Baudrate=125
MasterMode=1
[CAN1]
Enabled=0
NodeID=0x30
Baudrate=125
MasterMode=0
[ETH0]
PortNum=8888
[ProcImg]
EnableExtIo=1
```

```
EnableSharing=0
```

7.7 Service Configuration and Boot Scripts

The Debian GNU/Linux installed on the device uses "systemd" for managing services. Besides the default services of the operating system, there are some additional services available. For executing simple commands on bootup of the system, one can also extend the old-fashioned *rc.local* start script.

7.7.1 Extend Shell Script in /etc/rc.local

The start script /*etc/rc.local* will be executed automatically at startup of the system. This file can be altered by the user to execute additional shell commands. One has to keep in mind to not block the execution of the script for a long time or start long running commands in background.

7.7.2 Add Custom Systemd Services

A much more flexible way to execute applications on boot-up or running Linux daemons, is to use "systemd" services. Users can add custom services on their own. System service files have to be added in /etc/systemd/system/<YOUR_ SERVICE>.service. The service-file has to contain at least the following options:

```
[Unit]
Description=<YOUR_SERVICE_DESCRIPTION>
[Service]
ExecStart=/usr/bin/YOUR_SERVICE_EXECUTABLE
[Install]
WantedBy=multi-user.target
```

Description is the name for the service and *ExecStart* is the path to the executable file or script. The service can be started automatically at boot-up with the following command:

systemctl enable YOUR_SERVICE

One can also disable it with the following command:

© SYSTEC electronic AG 2022	L-2562e_01	Page 32
	2 20020_01	i ugo or

systemctl disable YOUR SERVICE

As more in-depth examples, one could look up the service files of *OpenPCS* or *Node-RED*. The official documentation can be found in the provided man-pages or on the project site of *systemd*. Important man-pages regarding system services:

man systemd
man systemd.unit
man systemd.service

Additional information can be found on the project homepage: <u>https://www.freedesktop.org/wiki/Software/systemd/</u>

7.7.3 Configure Services

The devicecomes with a few "systemd" services such as *OpenPCS, OPC UA Basis Server* or *Node-RED*. There are two ways to run these services:

1. The following commands are used to manually start or stop *OpenPCS* services:

```
systemctl start openpcs-z5
systemctl stop openpcs-z5
```

 There is also the possibility to start the OpenPCS services automatically on power-on reset or reboot. These are the same commands as mentioned in Section 7.7.2: systemctl enable openpcs-z5 To disable the automatic start, the following command is used: systemctl disable openpcs-z5

Table 8 shows a list of services, which can be configured by the user. Use the commands as above to run each of the services by substituting the name of the service.

Service file / name	Enabled by default	Description
adc-modes.service	Yes	Oneshot service, which sets up the ADC configuration from /etc/systec/adc_modes
node-red.service	No	Enables the Node-RED programming tool, listens by default on port 1880
openpcs- lighttpd.service	No	Web configuration frontend for OpenPCS
openpcs-z4.service	No	OpenPCS RT, uses the CANopen (CAN0) for communication to OpenPCS on the PC
		Hint: This service can only be started if "openpcs-z5.service" is not running.
openpcs-z5.service	No	OpenPCS RT, uses ethernet (UDP) for communication to OpenPCS on the PC
		Hint: This service can only be started if "openpcs-z4.service" is not running.
docker.service	No	Docker container engine
containerd.service	No	Container runtime

Table 8: systemd services

7.8 Predefined User Accounts

All user accounts listed in Table 9 are predefined upon delivery. Those allow for a login to the command shell and at the SFTP server of the device.

Table 9: Predefined user accounts

User name	Password	Remark
root	root	Predefined root user for the administration (system configuration, user administration, software updates etc.)
user	user	Main user account
PlcAdmin	Plc123	Administration user for OpenPCS (root user alias; provided for compatibility reasons to older products)

Hint: To secure the device, users have to change all predefined passwords (see 7.11). If users are not needed, they can be disabled or removed except the "root" user. Additionally, SSH can be configured for public key authentication for even better security.

7.9 Remote Access

7.9.1 Remote Login to the Command Shell

In some cases, the administration requires the ability to typing shell commands manually. Therefore, the user must be directly logged in at the module. There are two different possibilities:

- Logging in is possible with the help of a **Terminal program** (e.g. TeraTerm, see section 7.1) via the serial interface **SERVICE**.
- Alternatively, the login is possible using an **SSH client** (e.g. PuTTY or also TeraTerm) via the Ethernet interface **ETH0**.

For logging in with SSH via PuTTY or TeraTerm, the IP address provided in section 7.1 must be used.

🕵 PuTTY Configuration	×
Category: - Session - Logging - Terminal - Keyboard - Bell - Features - Window - Appearance - Behaviour - Translation - Colours - Colours - Connection - Data - Proxy - Tehet - Riogin - ScH - ScH	Basic options for your PuTTY session Specify the destination you want to connect to Host Name (or IP address) Port 192.168.10.197 22 Connection type: Rogin • SSH O Ray
About	<u>Upen</u> <u>Cancel</u>

Figure 11: SSH login PuTTY

Logging in is possible in the Terminal window (if connected via Service) or in the SSH window (if connected via ETH0). The following user account is preconfigured for the administration of the module upon delivery (also compare section 7.8):

© SYSTEC electronic AG 2022	L-2562e_01	Page 34
-----------------------------	------------	---------

User: root Password: root



Figure 12: Login to the device

Figure 12 shows the login using PuTTY.

7.9.2 Login to the SFTP Server

The device has available a SFTP server that allows file exchange with any computer (up- and download of files). "*WinSCP*" - which is available as open source - is suitable as SFTP client for the computer (see section 7.1). It consists of only one EXE file, needs no installation and may be started immediately. After program start, dialog "*WinSCP Login*" appears (see Figure 14) and must be adjusted according to the following configurations:

File protocol:	SFTP
Host name:	IP address for the device as set in section 7.4
User name:	root (for predefined user account, see section 7.8)
Password:	root (for predefined user account, see section 7.8)

🌆 Login		– 🗆 X
♥ New Site ♥ root@webbox1416.server-home.net	Session Ele protocol: SFTP Host name: 192.168.10.197 User name: Password root Save	Port number: 22 😒 d: Advanced 🔽
<u>I</u> ools ▼ <u>M</u> anage ▼	Login V Clos	se Help

Figure 13: Login settings for WinSCP

After using pushbutton "Login", the SFTP client logs in to the device and lists up the active content of directory "/root" in the right window. Figure 14 shows SFTP client "WinSCP" after successful login to a device.

🌆 / - root@192.168.10.197	7 - WinSCP				-		×
Local <u>Mark Files Commands Session Options Remote H</u> elp							
🛲 😤 🚔 Synchronize 🔳 🖉 🐼 🏟 Queue 🗸 🛛 Transfer Settings Default 🔹 🧖 🗸							
proot@192.168.10.197	New Sess	sion					
- C: Lokaler Datenträge	- 🖪 🔽 🕴	🔶 • 🐟 • 🕞 🔂	n 2 %	I < root> ▼ (2) ▼ → ▼ → ▼	h 🖓	🖸 Find Fil	es 📴
	V / B.	Dramartian 🔗 🕞			ortion		
C:\Tmp\CTR-700		-		/			
Name	Size	Туре	Changed	Name	Size	Changed	
t		Parent directory	15.05.2018 14:34:5	± .		15.05.2018	17:37:10
install-ctr-700-0510_0	937 KB	SH-Datei	15.05.2018 14:19:0	bin		15.05.2018	17:36:12
				boot		15.05.2018	17:37:10
				dev		16.05.2018	10:12:15
				etc		16.05.2018	11:05:00
				home		16.05.2018	11:15:48
				ID last found		15.05.2018	17:58:07
				iost+tound		16.05.2010	17,07,27
				met		00.05.2010	16:57:10
				ont		16.05.2010	09-20-00
				proc		01.01.1970	01:00:00
				root		16.05.2018	10:55:46
				run		16.05.2018	11:16:04
				sbin		15.05.2018	17:36:14
				srv		15.05.2018	17:07:37
				sys		16.05.2018	10:12:09
				n tmp		16.05.2018	11:17:01
				usr		15.05.2018	17:21:34
				var		15.05.2018	17:17:36
				- vendor		01.01.1970	01:00:00
<			>	<			>
0 B of 937 KB in 0 of 1				0 B of 0 B in 0 of 20			
	SFTP-3 🗐 0:00:21						

Figure 14: FTP client for Windows "WinSCP"

After successful login, configuration files on the device may be edited by using pushbuttons "F4" or "F4 Edit" within the SFTP client "WinSCP" (select transfer mode "Text"). With the help of pushbutton "F5" or "F5 Copy", files may be transferred between the computer and the device, e.g. for data backups or to transfer installation files for firmware updates (select transfer mode "Binary").

7.10 Adding and Deleting User Accounts

Adding and deleting user accounts requires the login as described in section 7.9.1.

Adding a new user account takes place via Linux command "useradd". To create a new user, one can use the command "useradd" as follows:

useradd [options] [username]

Advice: If the new user account shall be used to access web frontend, the user name must be entered into the configuration file (for details about logging in to WEB-Frontend please compare section 7.6.1 and 7.6.3).
To **delete** an existing user account from the device, Linux command *"userdel"* plus the respective user name must be used:

userdel [options] [username]

To get a full list of options for one of the commands, run it with the "--help" argument.

7.11 How to Change the Password for User Accounts

Changing the password for user accounts requires login as described in section 7.9.1.

To change the password for an existing user account, Linux command "passwd" plus the respective user name must be entered:

```
passwd <username>
```

Figure 15

exemplifies the password change for a user named "testuser".



Figure 15: Changing the password for a user account

7.12 Setting the System Time and Time Zone

Setting the system time requires login to the device as described in section 7.9.1.

The current date and time must be set using the Linux command "timedatectl set-time". Linux command "timedatectl set-time" is structured as follows:

timedatectl [options] set-time "YYYY-MM-DD hh:mm:ss"

Example:

timedatectl set-time "2017-12-01 10:20:55"

The current system time is displayed by entering Linux command *"timedatectl"* (without parameter). Figure 16 exemplifies setting and displaying the system time.

😕 COM3:115200baud - Tera Term VT
D <u>a</u> tei B <u>e</u> arbeiten Ein <u>s</u> tellungen Ste <u>u</u> erung <u>F</u> enster <u>H</u> ilfe
<pre>rootBctr-700-382731: # timedatect1 Local time: Fr: 2017-12-01 10:21:13 UTC Universal time: Fr: 2017-12-01 10:21:13 UTC RTC time: Fr: 2017-12-01 10:21:13 UTC NTP enabled: no NTP synchronized: no RTC in local TZ: no DST active: n/a rootBctr-700-382731: # timedatect1 set-time "2017-12-01 10:20:55" rootBctr-700-382731: # timedatect1 Local time: Fr: 2017-12-01 10:20:55 UTC Universal time: Fr: 2017-12-01 10:20:55 UTC Universal time: Fr: 2017-12-01 10:20:55 UTC NTP synchronized: no NTP synchronized: no NTP synchronized: no NTP synchronized: no NTP synchronized: no</pre>

Figure 16: Setting and displaying the system time

Upon start of the device, date and time are taken over from the RTC and set as current system time of the module.

The current time zone must be set using the Linux command "timedatectl set-timezone". Linux command "timedatectl set-timezone" is structured as follows:

timedatectl [options] set-timezone [TIMEZONE]

Example:

timedatectl set-timezone Europe/Berlin

Figure 17 exemplifies setting and displaying the time zone setting.



Figure 17: Setting and displaying the system time

With the following command, all available time zones can be looked up:

timedatectl list-timezones

7.13 File Systems

Table 10 lists the default filesystems and mountpoints of the device.

Path	Size	Description
/	8 GiB / size of SD Card	Root filesystem where the Linux is installed to. The content of this partition is stored on the EMMC or SD Card depending on the current Boot-Mode (see Table 3)
/mnt	-	Target for mounting remote directories
/vendor	4 MiB	Read only configuration data of the device. These data should not be altered by the user. The content is stored on one of the general-purpose partitions (aka. "boot0") to keep its contents event after a firmware update.

Table 10: File system configuration

Size, usage and path of file systems which are mounted can be identified by using Linux command "df" ("disk free").

Advice:	The general purpose EMMC partitions "/dev/mmcblk2boot0" and "/dev/mmcblk2boot1"
	contain vendor specific data. These partitions should not be used or altered by customers.
	Otherwise the device will not work as expected!

Particular information about the system login and handling the Linux command shell is given attention in section 7.9.

7.14 Software Installation and Update

All necessary firmware components to run the device are already installed on the module upon delivery. Hence, firmware updates should only be required in exceptional cases, e.g. to input new software that includes new functionality.

7.14.1 Updating the PLC Firmware

PLC firmware represents the run time environment of the PLC. The *PLC firmware* can only be generated and modified by the producer; **it is not identical with the PLC user program** which is created by the PLC user. The PLC user program is directly transferred from the *OpenPCS* programming environment onto the module. No additional software is needed.

Updating the PLC firmware requires login to the command shell as described in section 7.9.1 and login to the SFTP server as described in section 7.9.2.

Updating the PLC firmware takes place via a self-extracting firmware archive that is transferred onto the device via SFTP. The respective firmware archive can be transferred into directory "/tmp" (see Figure 18).

🌆 tmp - root@192.168.10	.197 - WinSCI	þ			-		×
Local Mark Files Comm	nands <u>S</u> essio	n <u>O</u> ptions <u>R</u> emote	<u>H</u> elp				
🖶 🔁 🚔 Synchronize	🗩 🦑 📓	🛯 🛞 🗿 Queue 🗣	Transfer Settin	ngs Default 🔹 🛃 🔹			
💻 root@192.168.10.197	💣 New Sess	sion					
🟪 C: Lokaler Datenträge	- 겸 🔽 🛛	🔶 🔹 🚽 🔹 🔁	🏫 🎜 💁	- tmp - 🚰 🔽 🗢 - 🖻	1 🗖 🏠 🛃	🔍 Find Fil	es 📴
🗐 Upload 👻 📝 Edit 👻	×AB	Properties 📑 🖻	+ - V	📲 🚰 Download 👻 📝 Edit 👻 🛃 🕞 Pr	operties 📑 [à 🗄 E	
C:\Tmp\CTR-700				/tmp			
Name	Size	Туре	Changed	Name	Size	Changed	
t		Parent directory	15.05.2018 14:34:5	t		15.05.2018	17:37:10
install-ctr-700-0510_0	937 KB	SH-Datei	15.05.2018 14:19:0	install-ctr-700-0510_0100.sh	937 KB	15.05.2018	14:19:04
<			>	<			>
0 B of 937 KB in 0 of 1			-	937 KB of 937 KB in 1 of 1			5 hidden
					SETP-3	iii 0:0	06:48

Figure 18: File transfer in SFTP client "WinSCP"

Important: To transfer the firmware archive via SFTP, transfer type "Default" or "Binary" must be chosen. If SFTP client "WinSCP" is used, the appropriate transfer mode is to be chosen from the menu bar. After downloading the firmware archive, it must be checked if the file transferred to the device has the exact same size as the original file on the computer (compare Figure 18). Any differences in that would indicate a mistaken transfer mode (e.g. "Text"). In that case the transfer must be repeated using transfer type "Binary".

After downloading the self-extracting archive, the PLC firmware must be installed on the device. Therefore, the following commands are to be entered in the SSH window. It must be considered that the file name for the firmware archive is labeled with a version identifier (e.g. "install-ctr-700-0510_0100.sh" for version 5.10.01.00). This number must be adjusted when commands are entered:

cd /tmp chmod +x install-ctr-700-0510_0100.sh ./install-ctr-700-0510_0100.sh

Advice: The command shell is able to automatically complete names if the Tab key is used ("tab completion"). Hence, it should be sufficient to enter the first letters of each file name and the system will complement it automatically. For example, "./ins" is completed to "./installctr-700-0510_0100.sh" if the Tab key is used.

Putry 192.168.10.197 - Putry	_	×
root@ctr-700:/tmp# chmod +x install-ctr-700-0510_0100.sh root@ctr-700:/tmp# ./install-ctr-700-0510_0100.sh		^
CTR-700 Runtime System Installer		
Running installation please wait		
./etc/		
./etc/systemd/		
./etc/systemd/system/		
./etc/systemd/system/openpcs-z5.service		
./etc/systemd/system/openpcs-lighttpd.service		
<pre>./etc/systemd/system/openpcs-z4.service //bttp/</pre>		
/http/		
/http/lighttpd.conf		
/http/mime.types		
/http/ntml/		
/http/ntml/Ctr700Sam.ntml		
/http/html/CTP_700_ppg		
/http/html/evetec_logo_ing		
/http/html/SamEvecFileDegDegeTnl html		
/http/html/sam_html		
/http/html/index.html		
/http/cgi-bin/		
/http/cgi-bin/sam.cgi		
/http/cgi-bin/cfgsetup.cfg		
/http/cgi-bin/webvisu.cfg		
./http/cgi-bin/webvisu.fcgi		
./http/cgi-bin/cfgsetup.cgi		
./http/cgi-bin/sam.cfg		
./install.sh		
./plc/		
./plc/stopplc		
./plc/visudata/		
./plc/delplcprog		
./plc/version		
./plc/runplc		
./plc/bin/		
./plc/bin/iodrvdemo		
./plc/bin/ctr-700-z4		
./plc/bin/ctr-700-z5		
./plc/bin/ctr-700.cfg		
./plc/bin/shpimgdemo		
./pic/picdata/		
./pic/printlog		
Flush file buffers		
Installation has been finished.		
Please restart system to activate the new firmware.		
root@ctr-700:/tmp#		~

Figure 19: Installing PLC firmware

Figure 19 shows the installation of PLC firmware. After reboot the module is started using the updated firmware.

Advice: If the PLC firmware is updated, the configuration file is overwritten. This results in a reset of the PLC configuration to default settings. Consequently, after an update, the configuration described in section 7.6 should be checked and if necessary it should be reset.

7.14.2 Install Debian GNU/Linux to a SD Card for update/recovery

Advice: Installing Debian GNU/Linux to the SD Card will format all partitions of the SD Card. This means all data on the device will be overwritten.

The device supports to boot from SD Card as well as from EMMC. The following steps describe how to install a new firmware version to an SD Card and boot from it. This card can then be used to install Debian GNU/Linux to the internal EMMC. The SD Card has to have a size of at least 4 GiB. The SD Card software is provided as a compressed image, which contains the whole file system of the operating system.

© SYSTEC electronic AG 2022	
-----------------------------	--

The SD Card Image is provided by SYS TEC. The following steps assume the file *"ctr-700-sdcard-v0100.img.zip"* is used. Newer versions will have a slightly different file name. Follow the steps to install this file to an SD Card.

- 1. Unzip the file
- 2. Download the tool *"Win32 Disk Imager"*: <u>https://sourceforge.net/projects/win32diskimager/</u> This tool will be used to copy the image file to the SD Card image.
- 3. Insert your SD Card
- 4. Run "Win32 Disk Imager"
 - a. Choose the uncompressed SD Card image
 - b. Choose the drive letter of your SD Card
 - c. Click on the "Write" button to write the image to your SD Card

🖢 Win32 Disk Imager - 1.0	_		×
Image File		Device	
C:/Tmp/CTR-700/ctr-700-sdcard-v0100.img	2	[E:\]	•
Hash			
None Generate Copy			
Read Only Allocated Partitions			
Progress			
Cancel Read Write Veri	fy Only	Exit	
Contraction of the second seco			

Figure 20: Write SD Card Image using Win32 Disk Imager

- 5. Insert the SD Card and switch the boot mode to SD Card (turn DIP-Switch 6 off, see Table 3)
- 6. Power-on the device

The device will now boot from the SD Card. Use a command shell as described in section 7.9.1 to work with the new firmware.

7.14.3 Install/Update Debian GNU/Linux to EMMC

Advice: Installing Debian GNU/Linux to the EMMC will format all partitions of the EMMC. This means all data on the device will be overwritten except vendor data partitions, which contain ADC calibration data or information of purchased licenses.

To be able to install Debian GNU/Linux to the internal EMMC of the device, one has to create an SD Card as described in section 7.14.2.

After a SD Card is available, follow the steps to install the Linux Image from SD Card to the EMMC and boot it afterwards:

- 1. Insert the SD Card and switch the boot mode to SD Card (turn DIP-Switch 6 off, see Table 3)
- 2. Start / boot the device
- Login to a command line shell as described in section 7.9.1
 User: root
 Password: root
- 4. Execute the following command to install the Linux to the EMMC. This will take a few minutes to execute.



- 5. Switch the boot mode to EMMC (turn DIP-Switch 6 on, see Table 3)
- 6. Reboot or shutdown the system. The device will now boot from EMMC.

reboot

Now the Debian GNU/Linux is installed to the EMMC memory and it can be booted. Using the EMMC has several benefits. The storage is most of the time faster than using an SD Card and it is more reliable in terms of durability.

7.15 TeamViewer IoT Agent

The device can be used with the TeamViewer IoT Agent. Separate licensing is required for the use of the agent. It has a command line user interface. All commands are available via the command line interface tool teamviewer-iot-agent.

The agent is not pre-installed by default and has to be loaded and activated separately. The full documentation and download instruction can be found on the TeamViewer website: <u>https://community.teamviewer.com/t5/TeamViewer-IoT-Labs/TeamViewer-IoT-for-the-CTR-700/td-p/44477</u>

7.16 qbee.io

The device can be used with an agent developed by qbee. Separate licensing is required for the use of this software and platform.

The agent is not pre-installed by default and has to be loaded and activated separately. The full documentation and download instruction can be found on the qbee website: https://gbee.io/docs/index.html

For more detailed information contact us at <u>info@systec-electronic.com</u> or qbee at <u>sales@qbee.io</u>.

8 Node-RED Programming Environment

The Node-RED programming environment allows a simple flow-driven approach to program *Internet of Things* applications in a web browser. This is based on a node editor, which provides nodes for:

- Inputs: These nodes have a single output and trigger a flow by sending a *msg* object. A trigger for an input node can be receiving a timer, a network packet, change of a digital input of the device and much more.
- Outputs: These nodes finish the flow execution and provide some reaction to the outside world. Possible reactions can be sending a network packet or tweet, setting a digital output of the device and much more.
- Functions: These nodes will operate on an input *msg* and provide some outputs. These are used to map data value, convert data or multiplex data. There are a lot of predefined nodes for common cases. For more advanced functionality they can also be programmed in *JavaScript*.

All nodes have zero or one Input and zero to n Outputs. SYS TEC provides Node-RED preinstalled on the device together with some custom Nodes. These nodes can be used to access inputs and outputs of the device. Additionally, nodes to access OpenPCS variables are provided.

The following section will provide information on the creation of a simple demo program, using the Node-RED programming environment and the nodes for the I/O Driver.

8.1 Running Node-RED

As described in section 7.7.3, Node-RED has to be started by issuing the following command in a command line shell:

```
systemctl enable node-red
systemctl start node-red
```

This will enable it permanently after boot-up and start the environment without the need for a reboot. Starting Node-RED may take a few seconds. After this, one is able to open the Node-RED editor in a web browser. To access the editor, one can use the hostname or the IP address of the device. In case the device is configured for DHCP, one can find the IP address by issuing the following command.

ifconfig

To get the hostname of the device use the following command:

hostname

Now use one of the following links to get access to the Node-RED editor.

http://192.168.10.100:1880/ (replace with your IP address) or http://ctr-500-000000:1880/

Node-RED : 192.168.10.1	29 × +			—		×
← → Q 192.168.10.10	0:1880/#flow/27a8b028.a2e638				GĀ	≡
Node-RED				- Deploy	• E	≣
Q filter nodes	Flow 1	+	info	debug	dashboard	ĸ
✓ input ^		^	✓ Information	on		^
⇒ inject			Flow	"27a8b028.a2	e638"	
y inject			Name	Flow 1		
catch			Status	Enabled		_
status			✓ Flow Des	cription		
🗦 link 🖓			None			
)) mqtt						
http						
websocket						
🤌 tcp 🖓						×
) udp			Move the s	elected nodes u	ising the 🔶	
			↑ ↓ and	→ keys. Hold	to nudge	2
✓ output				them further		
debug 🗐 🗸 🤇		>				
A ¥	-	- 0 +				

Figure 21: Node-RED editor in the web browser

Node-RED is configured for listening on port 1880 by default. This and the Node-RED configuration can be changed in the file */root/.node-red/settings.js.* The file contains a lot of comments describing the different options. For more documentation lookup the project site of Node-RED:

https://nodered.org/

8.2 Creating a demo application

This section describes the steps to create a simple run light application by using some of the default Node-RED nodes and the nodes from SYS TEC. Node-RED has to be running as described in the previous section.

The following figure shows the final application with all nodes. After a brief explanation of the setup the different settings of each node is explained in detail.

Rode-RED : 192.168.	10.129 × +						-	
← → Q 192.168.10	.100:1880/#flow/27a8b	028.a2e638					G	$\overline{+}$ =
Node-RED						-⁄= (Deploy 🔻	
Q filter nodes	Flow 1				+	info	debug	das 🕸
✓ input ^					^	v Info	ormation	^
⇒ iniect					DOA	Flow	"2522eab2	.224166
-						Name	Flow 1	
catch P	trigger υ	8-bit shifter	8-bit mask		DO1	Status	Enabled	
status					0	~ Flo	w Descript	ion
🔄 link 🖓					DO2	None		
) mgtt								
					0			
nttp					DO4			
websocket					0			
🔅 tcp 🛛					DO5	<		>
					1			2 × ^
							Your flow	
✓ output					007	confi	guration no	odes
				~		are	e listed in t	he
debug	<				>	sideb	ar panel. I	t can
A V					- 0 +			

Figure 22: Node-RED demo application

The application contains 3 kinds of nodes starting from the left. The first node called "trigger" gives a repeating input signal, which triggers the execution of the flow. The second node "*8-bit shifter*" will shift a numeric value to the left at each incoming message and outputs the current value. The third node "*8-bit mask*" converts a numeric value (0-255) and provides an output for each value. The nodes "DOO" to "DO7" take the value 0 or 1 and set a digital output accordingly.

Setup the "*trigger*" node:

- Add a node of kind "inject" by dragging it to the editor pane and double click on it
- Set "Repeat" to interval, with a value of 0.1 seconds

Node	RED Depicy -
Edit inject noo	e
Delete	Cancel Done
v node prop	erties
🗠 Payload	✓ timestamp
Topic	
	\Box Inject once after 0.1 seconds, then
C Repeat	interval ~
	every 0.1 seconds
Name	trigger
Note: "inter	al between times" and "at a specific time" will u

Figure 23: Node: "trigger"

- Click on "Done" to confirm the changes

Setup the "8-bit shifter" node:

- Add a node of kind "function" by dragging it to the editor panel and double click on it
- Copy the following code to the "Function" text area and adjust the settings as shown in the screenshot.



Click on "Done" to confirm the changes

Setup the "8-bit mask" node:

- Add a node of kind "function" by dragging it to the editor pane and double click on it
- Copy the following code to the "*Function*" text area, set *Outputs* to 8 and adjust the settings as shown in the screenshot.

```
// -----
let input = msg.payload;
let outputs = [];
for (let i = 0; i < 8; i++) {
    let channel = {};</pre>
```

© SYSTEC electronic AG 2022

L-2562e_01

```
channel.topic = "";
if (input & (1 << i)) {
    channel.payload = 1;
} else {
    channel.payload = 0;
}
outputs.push(channel);
}
```

return outputs; // -----



Figure 25: Node: "8-bit mask"

- Click on "Done" to confirm the changes

Setup the "DOO" node:

- Add a node of kind "ctr700_out" by dragging it to the editor pane and double click on it
- Set the *Channel* to DO0, the *Init state* to 0, select *Use alternative topic* and set it to "#" like shown in the screenshot below:

Node-	RED Ceploy -						
Edit ctr700_out node							
Delete	Cancel Done						
v node prope	rties						
i≣ Channel	DO0 v						
☑ Init State	0						
13	□ Negate output state						
	Use alternative topic #						
> node settings							

Figure 26: Node: "DO0"

- Click on "Done" to confirm the changes

Copy the output node for *DO0* multiple times for each output until *DO7* and adjust the settings for each digital output. Connect all nodes as shown in Figure 22. Now the application is complete. To start it, click on the *Deploy* button in the top right corner of the editor interface. The output LED should now show a run light on *DO0* to *DO7*.

9 Mono/C#

The device comes with the preinstalled mono package together with C# bindings for the I/O driver. This allows to write C# applications for the device in Microsoft Visual Studio 2017. The application will implement a simple run light for digital outputs 0 to 7. Here on the example of the CTR-700.

9.1 Create a C# application

New Project ? × ▷ Recent Sort by: Default • # E - م Installed Type: Visual C# Blank App (Universal Windows) Visual C# A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS. ▲ Visual C++ Windows Universal <u>۲:</u> Console App (.NET Core) Visual C# ▷ Visual C# Class Library (.NET 묎 Visual Basic Visual C# ▷ JavaScript Other Project Types Shared Project Visual C# ▷ Online 묎 Class Library (Legacy Portable) Visual C# 묎 Class Library (Universal Windows) Visual C# 툈 Windows Runtime Component (Universal Windows) Visual C# 屃 Unit Test App (Universal Windows) Visual C# Not finding what you are looking for? Demo Name: Location: C:\Users\Dinter\source\repos <u>B</u>rowse... ✓ Create <u>d</u>irectory for solution Solution name: Demo Add to Source Control Cancel

Create a console application as shown in the screenshot below:

Figure 27: Visual Studio console application

The bindings to the I/O Driver are located in the *Oracle VM VirtualBox* provided for the device. (3912005 "Oracle VM VirtualBox-Image of the Linux development system") The source code of the bindings needs to be copied to the project. The path, for example for the CTR-700, to the bindings in the *Oracle VM VirtualBox* is:

/projects/CTR-700/driver/ctr700drv/Bindings/CSharp/Ctr700Driver/Ctr700.cs

The content of the file Program.cs needs to be replaced with the following source code:

```
using System;
using Systec.Ctr700Driver;
using Mono.Unix;
using Mono.Unix.Native;
namespace Demo
{
    class Program
    {
        static int Main(string[] args)
        {
             using (var ctr700 = new Ctr700())
             {
                 MainLoop(ctr700);
             }
             return 0;
        }
        private static void MainLoop(Ctr700 ctr700)
        {
             const byte START = 1;
             const byte DELAY_MS = 100;
             byte bMask = START;
            UnixSignal sigint = new UnixSignal(Signum.SIGINT);
            while (!sigint.WaitOne(0))
             {
                 bMask <<= 1;</pre>
                 if (bMask == 0)
                 {
                     bMask = START;
                 }
                 for (byte bChannel = 0; bChannel < 8; bChannel++)</pre>
                 {
                     bool fValue = (bMask & (1 << bChannel)) != 0;</pre>
                     ctr700.SetDigitalOutput(bChannel, fValue);
                 }
                 System.Threading.Thread.Sleep(DELAY MS);
             }
             for (byte bChannel = 0; bChannel < 8; bChannel++)</pre>
             ł
                 ctr700.SetDigitalOutput(bChannel, false);
             }
        }
    }
}
```

The application uses the class Ctr700Drv, which contains static methods to access the driver. This will be used to set the digital outputs of the CTR-700. The constructor for this class also deconstructs it. (see variable creation before calling the MainLoop-Method)

Inside the main loop a mask will be shifted and reset, if the value is zero. The value of the mask will be signaled by bitwise activating the digital outputs represented by the mask. Each loop cycle is finished by sleeping for 100 milliseconds.

Now the project can be built by clicking on Build / Build Demo.

© SYSTEC electronic AG 2022	L-2562e_01	Page 51

9.2 Run the application

To run the application on the target, copy the build Demo.exe in the projects output directory to the device. This can be done by using SFTP (see section 7.9.2). Login the device with a command shell via SSH or a Terminal program. Run the application by running the following commands.

```
cd <DIRECTORY_CONTAINING_THE_DEMO>
mono Demo.exe
```

The application can be stopped by pressing CTRL-C. One can also directly execute applications without using mono explicitly if the binary is marked as executable file. The following commands show how to achieve this:

```
chmod +x Demo.exe
./Demo.exe
```

10 Java

The device comes with the preinstalled OpenJDK package and also with Java bindings for the I/O driver. The following section will describe how to setup a Java application in the Oracle VM VirtualBox provided for the device. (3912005 "Oracle VM VirtualBox-Image of the Linux development system") The application will implement a simple run light for digital outputs 0 to 7. Here on the example of the CTR-700.

10.1 Create a Java Project

The following steps describe the steps necessary to create the demo application. Start Eclipse, choose a workspace and create a new *Java Project*.

▼ New Java Project	+ ×
Create a Java Project Create a Java project in the workspace or in an external log	cation.
Project name: Demo	
✓ Use default location	
Location: /home/vbox/workspace/Demo	Browse
JRE	
• Use an execution environment JRE:	JavaSE-1.7 ‡
 Use a project specific JRE: 	java-8-openjdk-amd64 🌲
O Use default JRE (currently 'java-8-openjdk-amd64')	Configure IREs
Project layout	
O Use project folder as root for sources and class file	s
• Create separate folders for sources and class files	Configure default
Working sets	
Add project to working sets	
Working sets:	\$ Select
The default compiler compliance level for the current project will use a project specific compiler compliant	nt workspace is 1.4. The new ce level of 1.7.
Sack Next >	Cancel Finish

Figure 28: Java Demo: Project setup

Create the Project structure:

Open /projects/CTR-700/driver/ctr700drv/Bindings/Java/ in the file manager and copy the *lib* directory and the file *src/com/systec/Ctr700Drv.java* to the project. The file manager is not needed anymore and can be closed.

After this, one can use the auto-fix feature of *Eclipse* to move *Ctr700Drv.java* to the correct subdirectory of the project. By right clicking on *lib/jna.jar* in the *Package Explorer* and clicking on *Build Path / Add to Build Path* the JNA library can be used by the demo application.

Create a new class Main in the package demo by right clicking on *src* and then click *New / Class* and adjust setting as shown below:

© SYSTEC electronic AG 2022	
-----------------------------	--

+	New Java Class		+ ×
Java Class			C
Create a new Java c	ass.		9
Source folder:	Demo/src		Browse
Package:	demo		Browse
Enclosing type:			
Nama	N-1-		
Name:	Main		
Modifiers:	public O default O private	 protected 	
	abstract final static		
Superclass:	java.lang.Object		Browse
Interfaces:			Add
Which method stubs	would vou like to create?		
	v public static void main(String[] args)		
	Constructors from superclass		
	Inherited abstract methods		
Do you want to add	comments? (Configure templates and def	ault value <u>here</u>)	
	Generate comments		
?		Cancel	Finish

Figure 29: Java Demo: Main class

The structure in *Project Explorer* should now look like in the screenshot below. Both *Java* files have to exist under the *src* directory and the file *jna.jar* has to be a member of *Referenced Libraries*.



Figure 30: Java Demo: Project structure

Copy the following Java demo source code to Main.java:

```
package demo;
import com.systec.Ctr700Drv;
import com.systec.Ctr700Drv.Ctr700Exception;
public class Main {
      final static int START = 1;
      final static int DELAY MS = 100;
      static Ctr700Drv ctr700;
      public static void main(String[] args) throws
                                     InterruptedException {
            int iMask = START;
            try {
                  ctr700 = new Ctr700Drv();
                  ctr700.init();
                  Runtime.getRuntime().addShutdownHook(new Thread() {
                         @Override
                        public void run() {
                              for (int i = 0; i < 8; i++) {
                                     ctr700.setDigiOut(i, false);
                               }
                              ctr700.shutDown();
                         }
                  });
                  // Main loop
                  while (true) {
                         // Output LSB of the mask
                         for (int i = 0; i < 8; i++) {
                              boolean fValue =
                                     (iMask & (1 << i)) != 0;
                               ctr700.setDigiOut(i, fValue);
                         }
                         iMask <<= 1;</pre>
                         if (iMask > 256) {
                               iMask = START;
                         }
                        Thread.sleep(DELAY MS);
                  }
            } catch (Ctr700Exception e) {
                  System.err.println(e.getMessage());
            }
      }
}
```

The application creates an object of Ctr700Drv. This will be used to set the digital outputs of the CTR-700. Before using it one needs to call the *init()*-Method. On exit of the application the *shutdown*()-Method should be called. Inside the main loop a mask will be shifted and reset, if the value is to large to show the enabled output(s). The value of the mask will be signaled by bitwise activating the digital outputs represented by the mask. Each loop cycle is finished by sleeping for 100 milliseconds.

The demo application is now complete and can be run or debugged.

© SYSTEC electronic AG 2022	L-2562e_01	Page 55
-----------------------------	------------	---------

10.2 Run the Java application

Create a new *Run Configuration* of kind "*Java Application*" and change the settings as shown in the screenshot below. *Apply* and *Close*.

-	Run Configurations	+	×
Create, manage, and ru Run a Java application	n configurations		
Yep filter text C /C++ Application Ci /C++ Application Ci /C++ Unit E Clipse Application G / Gradle Project Java Applet Java Application Jo Java Application Jo Junit J ⁰ JUnit Plug-in Test Launch Group Ø SGi Framework Filter matched 11 of 11 items	Name: Demo Main @* Arguments A JRE C Classpath & Source Project: Demo Main class: demo.Main Include system libraries when searching for a main class Include inherited mains when searching for a main class Stop in main Apply	Browse Search Revert	
?	Close	Run	

Figure 31: Java Demo: Run configuration

Export a runnable JAR file by *File / Export... / Runnable JAR file*. Click on *Next*, choose the created run configuration and choose a destination path to export the JAR file to. This JAR file will then contain the demo source with all dependencies packaged as a single file.

-	Runnable JAR File Export		+ ×
Runnable JAR Fil Select a 'Java Applica	le Specification ation' launch configuration to use to create a runnable IA	R.	
	,		
Launch configuration	1:		
Demo - Demo			* *
Export destination:			
/projects/tftpboot/d	emo.jar	•	Browse
Library handling: Extract required Package require Copy required lib	libraries into generated JAR d libraries into generated JAR praries into a sub-folder next to the generated JAR		
Save as ANT scri	ipt		
ANT script location:	/home/vbox/workspace	~	Browse
(?)	<back next=""> Cance</back>		Finish

Figure 32: Java Demo: Export Java JAR archive

Copy the created JAR file to the device by using SFTP for example. This is described in section 7.9.2. Login to the device with a command shell via SSH or a Terminal program. Run the application by running the JVM with the JAR file as argument.

cd <DIRECTORY_CONTAINING_THE_JAR>
java -jar demo.jar

The application can be stopped by pressing CTRL-C.

10.3 Debug the Java application

Start the application on the target with some additional arguments to the JVM. This will start JVM in a debug mode, waiting for a remote to connect.

```
java -Xdebug \
    -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000 \
    -jar demo.jar
```

In *Eclipse* create another *Run Configuration* for debugging a *Remote Java Application* as shown below but substitute the *Host* IP address with the one of your device. After this click on *Debug* to start the debugging session.

•	Debug Configurations	+ ×
Create, manage, and run cor Attach to a Java virtual machine accep	Ifigurations ting debug connections	Ť
 C/C++ Application C/C++ Application C/C++ Application C/C++ Postmortem Debugger C/C++ Remote Application C/C++ Remote Application Java Applet Java Applet Java Applet Java Applet Jour Application Jour Applet Semote Java Application Remote Java Application Remote Java Script Rhino JavaScript Filter matched 17 of 17 items 	Name: Main Ame: Main Project: Demo Connection Type: Standard (Socket Attach) Connection Properties: Host: 192.168.10.129 Port: 8000 Allow termination of remote VM Apply	Browse
?	Close	Debug

Figure 33: Java Demo: Debug configuration

Now all common debugging features can be used to debug the application. This includes features such as pause execution, stepping through your code, setting breakpoints, watching variable values and much more.

Debug - Demo/src/demo/Main. File Edit Source Refactor Navigate Search Project Run Window	.java - Eclipse Platform Help	- + x
] ➡ ▼ (a) 	E IS Debug ♣Java [Resource
🏇 Debug 😫 🙀 i⇒ 🗖	🗱 🗣 Breakpoints	- 8
🔻 🖳 Main [Remote Java Application]	<u>*</u> =	🛚 🗉 🖉 🗶 🔆 📑 🖻 😤 🔻
🛡 🔗 OpenJDK Zero VM[192.168.10.129:8000]	Name	Value
🔻 🔊 Thread [main] (Suspended)	args	String[0] (id=25)
Main.main(String[]) line: 36	© iMask	8
NativeMethodAccessorImpl.invoke0(Method, Object, Object[]) line: i	0 i	3
NativeMethodAccessorImpl.invoke(Object, Object[]) line: 57	fValue	true
DelegatingMethodAccessorImpl.invoke(Object, Object[]) line: 43		
Method.invoke(Object, Object) line: 606		0
JarRsrcLoader.main(String[]) line: 58		
D Main.java 🕱		- 0
<pre>} }; }; }; }; // Main loop while (true) { // Output LSB of the mask for (int i = 0; i < 8; i++) { booleam fValue = (iMask & (1 << i ctr700.setDigiOut(i, fValue); } iMask <<= 1; if (iMask > 256) { iMask << 1; if (iMask > 256) { iMask << 5TART; } Thread.sleep(DELAY_MS); } catch (Ctr700Exception e) { System.err.println(e.getMessage()); } }</pre>)) != 0;	
□ Writable Smart Insert 36 : 43		# 📮 🖉 🐍 🕥

Figure 34: Java Demo: Debug example

11 Python

The device comes with Python 2 and Python 3 and provides Python bindings for the I/O driver. This allows to write Python applications for the device, which allow access to the low-level peripherals. The following application will implement a simple run light for digital outputs 0 to 7. Here on the example of the CTR-700.

11.1 Create a Python application

The bindings to the I/O Driver are located in the *Oracle VM VirtualBox*. (3912005 "Oracle VM VirtualBox-Image of the Linux development system") The source code of the bindings needs to be copied to the project. The path to the bindings in the *Oracle VM VirtualBox* is:

/projects/CTR-700/driver/ctr700drv/Bindings/Python/ctr700drv.py

Create a file demo.py and fill it with the following source code:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import signal
import time
import sys
from ctr700drv import Ctr700Drv
# signal handler to intercept ctrl+c
def signal handler(sig, frame):
    for uChannel in range(0, 8):
        ctr700.set digi out(uChannel, False)
    ctr700.shutdown()
    sys.exit(0)
signal.signal(signal.SIGINT, signal handler)
if name == " main ":
    uDigiOut = 1
    ctr700 = Ctr700Drv()
    ctr700.init()
    while True:
        uDigiOut = (uDigiOut << 1 | uDigiOut >> (8 - 1)) & Oxff
        for uChannel in range(0, 8):
            if (uDigiOut & (1 << uChannel)):</pre>
                fDoState = True
            else:
                fDoState = False
            ctr700.set digi out (uChannel, fDoState)
        time.sleep(0.3) # wait 300ms
```

The application uses the class Ctr700Drv, which contains methods to access the driver. This will be used to set the digital outputs of the device. Before using it, one needs to call the *init()*-Method. On exit

© SYSTEC electronic AG 2022	L-2562e_01	Page 59
-----------------------------	------------	---------

of the application the *shutdown*()-Method should be called. Inside the main loop a mask will be shifted. The value of the mask will be signaled by bitwise activating the digital outputs represented by the mask. Each loop cycle is finished by sleeping for 300 milliseconds.

11.2 Run the Python application

To run the application on the target, copy the created demo.py together with the Python bindings ctr700drv.py to the device. This can be done by using SFTP (see section 7.9.2). Login with a command shell via SSH or a Terminal program.

It is necessary to install all mandatory python dependencies to run the application successfully. Therefore, copy the one of the Python setup files, which are located in */usr/share/libctr700drv/demo/python/* and execute it. The following steps will prepare to run it for Python 3. The same can be achieved for Python 2 by copying and using the file *setup_py2.sh* instead of *setup_py3.sh*.

```
cp /usr/share/libctr700drv/demo/python/ setup_py3.sh .
chmod +x setup_py3.sh
./setup py3.sh
```

After that, run the application by executing the following commands.

```
cd <DIRECTORY_CONTAINING_THE_PYTHON>
python demo.py
```

The application can be stopped by pressing CTRL-C.

11.3 Debug the Python application

To debug the application, the built-in debugger *pdb* can be used. Execute the following commands on the device to start a debugging session.

```
cd <DIRECTORY_CONTAINING_THE_PYTHON>
python -m pdb demo.py
```

Now the debugger is waiting for certain commands to control it. To get a list of all commands type *h* or *help*. A full reference of commands can be found in Python documentation: <u>https://docs.python.org/3.5/library/pdb.html</u>.

All common debugging features are available to debug the application. For example, a breakpoint can be set on line 27 with the following command.

b demo.py:27

To run the application until this breakpoint is reached, the *c* or *continue* command can be executed.

12 PLC Functionality

Further information about the supported PLC functionalities, e.g. retain variables or function blocks, can be found in the documents referenced inTable 1.

12.1 Starting the PLC Runtime

The PLC runtime is not enabled by default. To activate the automatic start, one has to enable the services *"openpcs-z4"* or *"openpcs-z5"* as described in section 7.7.3.

If enabled, the device loads all necessary firmware components upon power-on or reset and starts running the PLC program afterwards (if enabled). Hence, the device is suitable for the usage in independent control systems. In case of power breakdown, such systems resume the execution of the PLC program independently and without user intervention.

12.2 Programming the Device

The device is programmed with IEC 61131-3-conform *OpenPCS* programming environment. There exist additional manuals about *OpenPCS* that describe the handling of this programming tool. Those are part of the software package "*OpenPCS*". All manuals relevant are listed in Table 1.

The firmware is based on standard firmware for SYS TEC's compact control units. Consequently, it shows identical properties like other SYS TEC control systems. This affects especially the process image setup (see section 12.3) as well as the functionality of control elements (DIP-Switch, Run/Stop switch, Run-LED, Error-LED).

Depending on the firmware version used, it provides numerous function blocks to the user to access communication interfaces. Section 7.7.3 describes the selection of the appropriate firmware version.

Table 18 in **Appendix A** contains a complete listing of firmware functions and function blocks that are supported by the device. Detailed information about using the CAN interfaces in connection with CANopen is provided in section 12.8.

12.3 Process Image of the Device

12.3.1 Local In- and Outputs

Compared to other SYS TEC compact control systems, this device obtains a process image with identical addresses. All supported in- and outputs listed in Table 11.

I/O of the device	Address and Data ty	pe in the Process Image
DI0 DI4	%IB0.0 %IX0.0 %IX0.3	as Byte with DI0 DI3 as single Bit for each input
C0	%ID40.0 counter input: DI14 (%	31Bit + sign (-2 ³¹ - 2 ³¹ -1) bIX1.6), direction: DI15 (%IX1.7)
CPU Temperature Sensor	%ID72.0	31Bit + sign as 1/10000 °C
System Temperature Sensor	%ID76.0	31Bit + sign as 1/10000 °C

Table 11: Assignment of in- and outputs to the process image

DO0 DO3	%QB0.0 %QX0.0 %QX0.3	as Byte with DO0 DO3 as single Bit for each output
DO Mask	%QD1984.0 a bitmask of digital o process image to ha	Mask of digital outputs outputs, which will not be written from rdware registers. (LSB relates to DO0,

The device works with Little-Endian format ("Intel-Notation). Consequently, and on the Advice: contrary to controls using Big-Endian ("Motorola-Notation), it is possible to sum up several BYTE variables of the process image to one WORD or DWORD and to access Bits above Bit7. The following example shows issue described: bInByte0 АT 8IB0.0 : BYTE: bInByte1 AT8IB1.0 : BYTE; wInWord0 %IWO.O ΑT WORD; : wInWord0.0 == bInByte0.0 due to Little-Endian: wInWord0.0 <> bInByte1.0 wInWord0.8 == bInByte1.0 due to Little-Endian: wInWord0.8 <> bInByte0.0

In- and outputs are not negated in the process image. Hence, the H-level at one input leads to value "1" at the corresponding address in the process image. Contrariwise, value "1" in the process image leads to an H-level at the appropriate output.

12.3.2 Network Variables for CAN1

Contrary to interface CAN0, interface CAN1 of the device is designed as static object dictionary. Thus, at interface CAN1 the device acts as a CANopen I/O device. All static network variables for CAN1 are accessible via the marker section of the process image.

Section 12.3.2 includes more detailed information about CAN interface CAN1 and the network variables that are provided by it in the marker section.

12.4 Communication Interfaces

12.4.1 Serial Interfaces

The device features one service and three serial interfaces (X10, X15 ... X17). Details about hardware activation are included in section 6.6.1.

The interfaces may be used from a PLC program via function blocks of type "SIO_Xxx" (see manual "SYS TEC-specific Extensions for OpenPCS / IEC 61131-3", Manual no.: L-1054).

12.4.2 CAN Interfaces

Both CAN interfaces allow for data exchange with other devices via network variables and they are accessible from a PLC program via function blocks of type "CAN_Xxx" (see "User Manual CANopen Extension for IEC 61131-3", Manual no.: L-1008).

Section 12.3.2 provides detailed information about the usage of the CAN interfaces in connection with CANopen.

12.4.3 Ethernet Interface

Details about the hardware activation are included in section 6.6.3

© SYSTEC electronic AG 2022 L-2562e_01 Page 6	62
---	----

The interface is accessible from a PLC program via function blocks of type "LAN_Xxx" (see manual "SYS TEC-specific Extensions for OpenPCS / IEC 61131-3", Manual no.: L-1054).

The exemplary PLC program "*UdpRemoteCtrl*" illustrates the usage of function blocks of type "*LAN_Xxx*" within a PLC program.

12.5 Specific Peripheral Interfaces

12.5.1 Pulse Outputs

To release PWM and PTO signal sequences, the device features 2 pulse outputs (P0 and P1). Prior to its usage, all pulse outputs must be parameterized using function block "PTO_PWM" (see manual "SYS TEC-specific Extensions for OpenPCS / IEC 61131 3", Manual no.: L 1054). After the impulse generator is started, it takes over the control of respective outputs. After the impulse generator is deactivated, the respective output adopts the corresponding value that is filled in the process image for this output (see Table 11 in section 12.3.1). Table 12 lists the allocations between impulse channels and outputs.

Table 12: Allocation betweer	n impulse channels a	nd outputs
------------------------------	----------------------	------------

Impulse channel	Impulse output						
P0	P0 (DO14) %QX1.6						
P1	P1 (DO15) %QX1.7						

12.6 Control and Display Elements

12.6.1 Run/Stop Switch

The Run/Stop switch makes it possible to start and interrupt the execution of the PLC program. Together with start and stop pushbuttons of the *OpenPCS* programming environment, the Run/Stop switch represents a "logical" AND-relation. This means that the PLC program will not start the execution until the local Run/Stop switch is positioned to "*Run*" **AND** additionally the start command (cold, warm or hot start) is given by the *OpenPCS* user interface. The order hereby is not relevant. A run command given by *OpenPCS* while at the same time the Run/Stop switch is positioned to "*Stop*" is visible through quick flashing of the Run- and Error-LED.

12.6.2 Run-LED (green)

The Run-LED provides information about the activity state of the control system. The activity state is shown through different modes:

Table 13: Display status of t	the Run-LED
-------------------------------	-------------

LED Mode	PLC Activity State							
Off	The PLC is in state "Stop":							
	 the PLC does not have a valid program, 							
	 the PLC has received a stop command from the OpenPCS programming environment or 							
	the execution of the program has been canceled due to an internal error							
Quick flashing in	The PLC is on standby but is not yet executing:							
relation 1:8 to pulse	• The PLC has received a start command from the <i>OpenPCS</i> programming environment but the local Run/Stop switch is still positioned to " <i>Stop</i> "							
Slow flashing in relation 1:1 to pulse	The PLC is in state "Run" and executes the PLC program.							
Quick flashing in relation 1:1 to pulse	The PLC is in mode <i>"Reset"</i>							

12.6.3 Error-LED (red)

The Error-LED provides information about the error state of the control system. The error state is represented through different modes:

Table 14: Display status of	the Error-LED
-----------------------------	---------------

LED Mode	PLC Error State
Off	No error has occurred; the PLC is in normal state.
Permanent light	A severe error has occurred:
	• The PLC was started using an invalid configuration (e.g. CAN node address 0x00) and had to be stopped or
	• A severe error occurred during the execution of the program and caused the PLC to independently stop its state " <i>Run</i> " (division by zero, invalid Array access,), see below
Slow flashing in relation 1:1 to pulse	A network error occurred during communication to the programming system; the execution of a running program is continued. This error state will be reset independently by the PLC as soon as further communication to the programming system is successful.
Quick flashing in relation 1:1 to pulse	The PLC is in mode " <i>Reset</i> "
Quick flashing in relation 1:8 to pulse	 The PLC is on standby, but is not yet running: The PLC has received a start command from the <i>OpenPCS</i> programming environment but the local Run/Stop switch is positioned to "Stop"

In case of severe system errors such as division by zero of invalid Array access, the control system passes itself from state "*Run*" into state "*Stop*". This is recognizable by the permanent light of the Error-LED (red). In this case, the error cause is saved by the PLC and is transferred to the computer and shown upon next power-on.

© SYSTEC electronic AG 2022

12.7 Local Deletion of a PLC Program

PLC programs can only be deleted with an established connection via a terminal program, SSH or SFTP (see section 7.1). First, the device has to be stopped (S2 switched to left), then the file *PlcArchv.bin* found in *"/home/plc/plcdata/"* can be deleted. Only the file has to be deleted not the directory!

12.8 Using CANopen for CAN Interfaces

The device features two CAN interfaces (CAN0 ... CAN1), both are usable as CANopen Manager (conform to CiA Draft Standard 302). The configuration of both interfaces (active/inactive, node number, Bitrate, Master on/off) is described in section 7.6.

Both CAN interfaces allow for data exchange with other devices via network variables and they are usable from a PLC program via function blocks of type "CAN_Xxx". More details are included in "User Manual CANopen Extension for IEC 61131-3", Manual no.: L-1008.

The CANopen services **PDO** (*P*rocess *D*ata *O*bjects) and **SDO** (*S*ervice *D*ata *O*bjects) are two separate mechanisms for data exchange between single field bus devices. Process data sent from a node (**PDO**) are available as broadcast to interested receivers. PDOs are limited to 1 CAN telegram and therewith to 8 Byte user data maximum because PDOs are executed as non-receipt broadcast messages. On the contrary, **SDO** transfers are based on logical point-to-point connections ("Peer to Peer") between two nodes and allow the receipted exchange of data packages that may be larger than 8 Bytes. Those data packages are transferred internally via an appropriate amount of CAN telegrams. Both services are applicable for interface CAN0 as well as for CAN1.

SDO communication basically takes place via function blocks of type "CAN_SDO_Xxx" (see "User Manual CANopen Extension for IEC 61131-3", Manual no.: L-1008). Function blocks are also available for PDOs ("CAN_PDO_Xxx"). Those should only be used for particular cases in order to also activate non-CANopen-conform devices. For the application of PDO function blocks, the CANopen configuration must be known in detail. The reason for this is that the PDO function blocks only use 8 Bytes as input/output parameter, but the assignment of those Bytes to process data is subject to the user.

Instead of PDO function blocks, network variables should mainly be used for PDO-based data exchange. Network variables represent the easiest way of data exchange with other CANopen nodes. Accessing network variables within a PLC program takes place in the same way as accessing internal, local variables of the PLC. Hence, for PLC programmers it is not of importance if e.g. an input variable is allocated to a local input of the control or if it represents the input of a decentralized extension module. The application of network variables is based on the integration of DCF files that are generated by an appropriate CANopen configurator. On the one hand, DCF files describe communication parameters of any device (CAN Identifier, etc.) and on the other hand, they allocate network variables to the Bytes of a CAN telegram (mapping). The application of network variables only requires basic knowledge about CANopen.

For this device, the usage of PDO-based network variables is different for each CAN interface CAN0 and CAN1. Sections 12.8.1 and 12.8.2 provide more detail on this.

In a CANopen network, exchanging PDOs only takes place in status "OPERATIONAL". If the device is not in this status, it does not process PDOs (neither for send-site nor for receive-site) and consequently, it does not update the content of network variables. The CANopen Manager is in charge of setting the operational status "OPERATIONAL", "PRE-OPERATIONAL" etc. (mostly also called "CANopen Master"). In typical CANopen networks, a programmable node in the form of a PLC is used as CANopen-Manager. This device is able to take over tasks of the CANopen Manager at both CAN interfaces CANO and CAN1. How the Manager is activated is described in section 7.6.

As CANopen Manager, it is able to parameterize the CANopen I/O devices ("CANopen-Slaves") that are connected to the CAN bus. Therefore, upon system start via SDO it transfers DCF files generated by the CANopen configurator to the respective nodes.

12.8.1 CAN Interface CAN0

Interface CAN0 features a dynamic object dictionary. This implicates that after activating the PLC, the interface does not provide communication objects for data exchange with other devices. After downloading a PLC program (or its reload from the non-volatile storage after power-on), the required communication objects are dynamically generated according to the DCF file which is integrated in the PLC project. Thus, CAN interface CAN0 is extremely flexible and also applicable for larger amount of data.

For the PLC program, all network variables are declared as "VAR_EXTERNAL" according to IEC61131-3. Hence, they are marked as "outside of the control", e.g.:

```
VAR_EXTERNAL
NetVar1 : BYTE ;
NetVar2 : UINT ;
END_VAR
```

A detailed procedure about the integration of DCF files into the PLC project and about the declaration of network variables is provided in manual "User Manual CANopen Extension for IEC 61131-3" (Manual no.: L-1008).

When using CAN interface CAN0 it must be paid attention that the generation of required objects takes place upon each system start. This is due to the dynamic object directory. "Design instructions" are included in the DCF file that is integrated in the PLC project. **Hence, changes to the configuration can only be made by modifying the DCF file.** This implies that after the network configuration is changed (modification of DCF file), the PLC project must again be translated and loaded onto the device.

12.8.2 CAN Interface CAN1

On the contrary to interface CAN0, interface CAN1 is provided as static object dictionary. This means that the amount of network variables (communication objects) and the amount of PDOs available are both strongly specified. During runtime, the configuration of PDOs is modifiable. This implies that communication parameters used (CAN Identifier, etc.) and the allocation of network variables to each Byte of a CAN telegram (mapping), can be set and modified by the user. Thus, only the number of objects (amount of network variables and PDOs) is strongly specified in the static object dictionary. Consequently, application and characteristics of objects can be modified during runtime. For this reason, at interface CAN1 the device acts as a CANopen I/O device.

All network variables of the PLC program are available through the marker section of the process image. Therefore, 252 Bytes are usable as input variables and also 252 Bytes as output variables. To enable any data exchange with other CANopen I/O devices, the section of static network variables is mapped to different data types in the object dictionary (BYTE, SINT, WORD, INT, DWORD, DINT). Variables of the different data types are located within the same memory area which means that all variables represent the same physical storage location. Hence, a WORD variable interferes with 2 BYTE variables, a DWORD variable with 2 WORD or 4 BYTE variables. Figure 35 shows the positioning of network variables for CAN1 within the marker section.

System Manual sysWORXX CTR-500

	CANT	input va	riables													
	CAN1 IN0	CAN1 IN1	CAN1 IN2	CAN1 IN3	CAN1 IN4	CAN1 IN5	CAN1 IN6	CAN1 IN7	 CAN1 IN244	CAN1 IN245	CAN1 IN246	CAN1 IN247	CAN1 IN248	CAN1 IN249	CAN1 IN250	CAN1 IN251
BYTE / SINT, USINT	%MB 0.0 (Byte0)	%MB 1.0 (Byte1)	%MB 2.0 (Byte2)	%MB 3.0 (Byte3)	%MB 4.0 (Byte4)	%MB 5.0 (Byte5)	%MB 6.0 (Byte6)	%MB 7.0 (Byte7)	%MB 244.0 (Byte244)	%MB 245.0 (Byte245)	%MB 246.0 (Byte246)	%MB 247.0 (Byte247)	%MB 248.0 (Byte248)	%MB 249.0 (Byte249)	%MB 250.0 (Byte250)	%MB 251.0 (Byte251)
WORD / INT, UINT	%N 0. (Wo	1W 0 rd0)	%N 2. (Wo	1W 0 rd1)	%N 4. (Wo	/W 0 rd2)	%N 6. (Wo	/W .0 rd3)	 %N 244 (Word	/W 4.0 1122)	%N 24((Word	/W 6.0 1123)	%N 248 (Word	IW 3.0 1124)	%N 250 (Word	/W 0.0 1125)
DWORD / DINT, UDINT		%N 0. (Dw.)	MD .0 ord0)		%MD 4.0 (Dw ord1)		%MD 244.0 (Dw ord61)		%MD 248.0 (Dw ord62)							

CAN1 Input Variables

	CAN1	Output	Variable	s												
	CAN1 OUT0	CAN1 OUT1	CAN1 OUT2	CAN1 OUT3	CAN1 OUT4	CAN1 OUT5	CAN1 OUT6	CAN1 OUT7	 CAN1 OUT244	CAN1 OUT245	CAN1 OUT246	CAN1 OUT247	CAN1 OUT248	CAN1 OUT2490	CAN1 DUT250	CAN1 OUT251
BYTE / SINT, USINT	%MB 256.0 (Byte0)	%MB 257.0 (Byte1)	%MB 258.0 (Byte2)	%MB 259.0 (Byte3)	%MB 260.0 (Byte4)	%MB 261.0 (Byte5)	%MB 262.0 (Byte6)	%MB 263.0 (Byte7)	%MB 500.0 (Byte244)	%MB 501.0 (Byte245)	%MB 502.0 (Byte246)	%MB 503.0 (Byte247)	%MB 504.0 (Byte248)	%MB 505.0 (Byte249)	%MB 506.0 (Byte250)	%MB 507.0 (Byte251)
WORD / INT, UINT	%N 256 (Wor	1W 5.0 rd0)	%N 256 (Wo	/W B.0 rd1)	%N 260 (Wo	/W 0.0 rd2)	%N 262 (Wo	/W 2.0 rd3)	 %N 50 (Wor	/W 0.0 d122)	%N 50: (Wor	/IW 2.0 d123)	%N 50- (Word	/W 4.0 1124)	%N 50 (Wore	MW 6.0 d125)
DWORD / DINT, UDINT		%M 269 (Dwo	ИD 5.0 ord0)		%MD 260.0 (Dw ord1)		%MD 500.0 (Dw ord61)		%MD 504.0 (Dw ord62)							

Figure 35: Positioning of network variables for CAN1 within the marker section

Table 15 shows the representation of network variables through appropriate inputs in the object dictionary of interface CAN1.

Table 15: Representation of	f network variables for CAN1	by entries in the object dictionary
-----------------------------	------------------------------	-------------------------------------

OD section		OD variable EDS input	1	Data type CANopen	Data type IEC 61131-3					
Inputs (inputs f	Inputs (inputs for this device)									
Index Sub 1 252	2000H	CAN1InByte0 CAN1InByte251		Unsigned8	BYTE, USINT					
Index Sub 1 252	2001H	CAN1InSInt0 CAN1InSInt251		Integer8	SINT					
Index Sub 1 126	2010H	CAN1InWord0 CAN1InWord125		Unsigned16	WORD, UINT					
Index Sub 1 126	2011H	CAN1InInt0 CAN1InInt125		Integer16	INT					
Index Sub 1 … 63	2020H	CAN1InDword0 CAN1InDword62		Unsigned32	DWORD, UDINT					
Index Sub 1 … 63	2021H	CAN1InDInt0 CAN1InDInt62		Integer32	DINT					

Outputs (outputs for this device)							
Index Sub 1 252	2030H	CAN1OutByte0 CAN1OutByte251	Unsigned8	BYTE, USINT			
Index Sub 1 252	2031H	CAN1OutSInt0 CAN1OutSInt251	Integer8	SINT			
Index Sub 1 126	2040H	CAN1OutWord0 CAN1OutWord125	Unsigned16	WORD, UINT			
Index Sub 1 … 126	2041H	CAN1OutInt0 CAN1OutInt125	Integer16	INT			
Index Sub 1 63	2050H	CAN1OutDword0 CAN1OutDword62	Unsigned32	DWORD, UDINT			
Index Sub 1 63	2051H	CAN1OutDInt0 CAN1OutDInt62	Integer32	DINT			

The object dictionary of interface CAN1 in total has available 16 TPDO and 16 RPDO. The first 4 TPDO and RPDO are preconfigured and activated according to the Predefined Connection Set. The first 32 Byte of input and output variables are mapped to those PDOs. Table 16 lists all preconfigured PDOs for interface CAN1.

Table 16: Preconfigured PDOs for interface CAN1

PDO	CAN-ID	Data
1. RPDO	0x200 + NodeID	%MB0.0 %MB7.0
2. RPDO	0x300 + NodelD	%MB8.0 %MB15.0
3. RPDO	0x400 + NodeID	%MB16.0 %MB23.0
4. RPDO	0x500 + NodeID	%MB24.0 %MB31.0
1. TPDO	0x180 + NodeID	%MB256.0 %MB263.0
2. TPDO	0x280 + NodelD	%MB264.0 %MB271.0
3. TPDO	0x380 + NodelD	%MB272.0 %MB279.0
4. TPDO	0x480 + NodelD	%MB280.0 %MB287.0

Due to limitation to 16 TPDO and 16 RPDO, only 256 Bytes (2 * 16PDO * 8Byte/PDO) of total 504 Bytes for network variables in the marker section (2 252Bytes) can be transferred via PDO. Irrespective of that it is possible to access all variables via SDO.

The configuration (mapping, CAN Identifier etc.) of interface CAN1 typically takes place via an external Configuration Manager that parameterizes the object dictionary on the basis of a DCF file created by the CANopen configurator. By using default object inputs 1010H und 1011H, the device supports the persistent storage and reload of a backed configuration.

Alternatively, the configuration (mapping, CAN Identifier etc.) of the static object dictionary for interface CAN1 can take place from the PLC program by using SDO function blocks. Therefore, inputs *NETNUMBER* and *DEVICE* must be used as follows:

NETNUMBER	:= 1;	(*	Interface CAN1	*)
DEVICE	:= 0;	(*	local Node	*)

© SYSTEC electronic AG 2022	L-2562e_01	Page 68
-----------------------------	------------	---------

The PLC program example "*ConfigCAN1*" exemplifies the configuration of interface CAN0 through a PLC program by using function blocks of type "*CAN_SDO_Xxx*".

13 Data Exchange via Shared Process Image

13.1 Overview of the Shared Process Image

This device is using the operating system Debian, which is based on Linux. Thus, it is possible to execute other user-specific programs simultaneously to running the PLC firmware. The PLC program and a user-specific C/C++ application can exchange data by using the same process image (shared process image). Implementing user-specific applications **is based on the Software package** 3912005 ("Oracle VM VirtualBox-Image of the Linux development system").



Figure 36: Overview of the shared process image

Not all variables are utilizable via the shared process image within a C/C++ application. Only those directly addressed variables that the PLC program generates within the process image. As shown in Figure 36, two separate process images are used for the data exchange with an external application

© SYSTEC electronic AG 2022	L-2562e_01	Page 70
-----------------------------	------------	---------

inside of the PLC runtime system. This is necessary to meet the IEC 61131-3 requirement that the initial PLC process image may not be modified during the entire execution of one PLC program cycle. Thereby, the PLC program always operates with the internal process image that is locally generated within the PLC runtime system ("Local Image" in Figure 36). This is integrated within the PLC runtime system ("Local Image" in Figure 36). This is integrated within the PLC runtime system and is protected against direct accesses from the outside. On the contrary, the user-specific, external C/C++ application always uses the shared process image ("Shared Image" in Figure 36). This separation of two process images enables isolation between accesses to the PLC program and the external application. Those two in parallel and independently running processes now must only be synchronized for a short period of time to copy the process data.

An activation of **option** "Share PLC process image" within the PLC configuration enables data exchange with external applications (see section 7.6.1). Alternatively, entry " *EnableSharing=*" can directly be set within section "[*ProcImg*]" of the configuration file (see section 7.6.3). The appropriate configuration setting is evaluated upon start of the PLC firmware. By activating option "Share PLC process image", the PLC firmware creates a second process image as Shared Memory ("Shared Image" in Figure 36). Its task is to exchange data with external applications. Hereby, the PLC firmware functions as Server and the external, user-specific C/C++ application functions as Client.

ReadSectorTable and **WriteSectorTable** both control the copying of data between the two process images. Both tables are filled by the Client (external, user-specific C/C++ application) and are executed by Server (PLC runtime system). The Client defines ranges of the PLC process image from which it will read data (*ReadSectorTable*) or in which it will write data (*WriteSectorTable*). Hence, the terms "*Read*" and "Write" refer to data transfer directions from the viewpoint of the Client.

Sections to read and write may comprise all sections of the entire process image – input, output as well as marker sections. This allows for example that a Client application writes data into the input section of the PLC process image and reads data from the output section. Figure 36 shows the sequence of single read and write operations. Prior to the execution of a PLC program cycle, the physical inputs are imported into the local process image of the PLC (1). Afterwards, all sections defined in *WriteSectorTable* are taken over from the shared process image into the local process image (2). By following this sequence, a Client application for example is able to overwrite the value of a physical input. This may be used for simulation purposes as well as for setting input data to constant values (*"Forcen"*). Similarly, prior to writing the process image onto the physical outputs (6), sections defined in *WriteSectorTable* are taken over from the shared process image into the local process image. (4). Thus, a Client application is able to overwrite output information generated by the PLC program.

The PLC firmware provides the setup of the process image. The Client application receives information about the setup of the process image via function ShPImgCIntSetup(). This function enters start offsets and values of the input, output and marker sections into the structure of type tShPImgLayoutDscrpt. Function ShPImgCIntGetDataSect() provides the start address of the shared process image. Upon defining a variable within the PLC program, its absolute position within the process image is determined through sections (%I = Input, %Q = Output, %M = Marker) and offset (e.g. %MB128.0). In each section the offset starts at zero, so that for example creating a new variable in the marker section would be independent of values in the input and output section. Creating a corresponding pair of variables in the PLC program as well as in the C/C++ application allows for data exchange between the PLC program and the external application. Therefore, both sides must refer to the same address. Structure tShPImgLayoutDscrpt reflects the physical setup of the process image in the PLC firmware including input, output and marker sections. This is to use an addressing procedure for defining appropriate variables in the C/C++ application that is comparable to the PLC program. Hence, also in the C/C++ program a variable is defined in the shared process image by indicating the respective section and its offset. The following example illustrates the creation of a corresponding variable pair in the PLC program and C/C++ application:



As described above, **ReadSectorTable** and **WriteSectorTable** manage the copy process to exchange variable contents between the PLC and the C/C++ program. Following the example illustrated, the Client (C/C++ application) must enter an appropriate value into the *WriteSectorTable* to transfer the value of a variable from the C/C++ application to the PLC program (*WriteSectorTable*, because the Client "writes" the variable to the Server):

```
// specify offset and size of 'Varl' and define sync type (always or on demand?)
WriteSectTab[0].m_uiPImgDataSectOffs = ShPImgLayoutDscrpt.m_uiPImgMarkerOffs + 128;
WriteSectTab[0].m_uiPImgDataSectSize = sizeof(BYTE);
WriteSectTab[0].m_SyncType = kShPImgSyncOnDemand;
// define the WriteSectorTable with the size of 1 entry
```

ShPImgClntSetupWriteSectTable (WriteSectTab, 1);

If several variable pairs are generated within the same transfer direction for the data exchange between the PLC program and the C/C++ application, they should possibly all be defined in one coherent address range. Thus, it is possible to list them as one entry in the appropriate SectorTable. The address of the first variable must be set as the *SectorOffset* and the sum of the variable sizes as *SectorSize*. Combining the variables improves the efficiency and the performance of the copy processes.

For each entry of the *WriteSectorTable* an appropriate *SyncType* must be defined. It determines whether the section is generally taken over from the shared process image into the local image whenever there are two successive PLC cycles (*kShPImgSyncAlways*) or whether it is taken over on demand (*kShPImgSyncOnDemand*). If classified as *SyncOnDemand*, the data only is copied if the respective section before was explicitly marked as updated. This takes places by calling function *ShPImgCIntWriteSectMarkNewData()* and entering the corresponding *WriteSectorTable*-Index (e.g. 0 for *WriteSectTab[0]* etc.).

kShPImgSyncAlways is provided as *SyncType* for the *ReadSectorTable* (the value of the member element *m_SyncType* is ignored). The PLC firmware is not able to identify which variables were changed by the PLC program of the cycle before. Hence, all sections defined in *ReadSectorTable* are always taken over from the local image into the shared process image. Thus, the respective variables in the shared process image always hold the actual values.

The PLC firmware and the C/C++ application both use the shared process image. To prevent conflicts due to accesses from both of those in parallel running processes at the same time, the shared process image is internally protected by a semaphore. If one process requires access to the shared process image, this process enters a critical section by setting the semaphore first and receiving exclusive access to the shared process image second. If the other process requires access to the shared process image at the same time, it also must enter a critical section by trying to set the semaphore. In this case, the operating system identifies that the shared process image is already being used. It blocks the second process until the first process leaves the critical section and releases the semaphore. Thereby, the operating system assures that only one of the two in parallel running processes (PLC runtime system and C/C++ application) may enter the critical section and receives access to the shared process image. To ensure that both processes do not interfere with each other too much, they should enter the critical section as less as possible and only as long as necessary. Otherwise, the PLC cycle time may be extended and runtime variations (Jitter) may occur.
The client application has available two functions to set the semaphore and to block exclusive access to the shared process image. Function *ShPImgCIntLockSegment()* is necessary to enter the critical section and function *ShPImgCIntUnlockSegment()* to leave it. The segment between both functions is called protected section, because in this segment the client application holds access to the shared process image without competition. The consistency of read or written data is only guaranteed within such a protected section. Outside the protected section, the shared process image may anytime be manipulated by the PLC runtime system. The following example shows the exclusive access to the shared process image in the C/C++ application:

```
ShPImgClntLockSegment();
{
    // write new data value into Var1
    *pbVar1 = bAnyData;
    // mark new data for WriteSectorTable entry number 0
    ShPImgClntWriteSectMarkNewData (0);
}
ShPImgClntUnlockSegment();
```

For the example above, *kShPImgSyncOnDemand* was defined as *SyncType* upon generating entry *WriteSectorTable*. Hence, taking over variable *Var1* from the shared process image into the local image can only take place if the respective section was beforehand explicitly marked as updated. Therefore, it is necessary to call function *ShPImgCIntWriteSectMarkNewData()*. Since function *ShPImgCIntWriteSectMarkNewData()* does not modify the semaphore, it may only be used within a protected section (see example) – such as the code section between *ShPImgCIntLockSegment()* and *ShPImgCIntUnlockSegment()*.

The synchronization between local image and shared process image by the PLC runtime system only takes place in-between two successive PLC cycles. A client application (user-specific C/C++ program) is not directly informed about this point of time, but it can get information about the update of the shared process image from the PLC runtime system. Therefore, the client application must define a callback handler of the type *tShPImgAppNewDataSigHandler*, e.g.:

```
static void AppSigHandlerNewData (void)
{
    fNewDataSignaled_1 = TRUE;
}
```

This callback handler must be registered with the help of function **ShPImgCIntSetNewDataSigHandler()**. The handler is selected subsequent to a synchronization of the two images.

The **callback handler of the client application is called within the context of a Linux signal handler** (the PLC runtime system informs the client using Linux function *kill()*). Accordingly, all common **restrictions** for the Linux signal handler also apply to the callback handler of the client application. In particular, it is only allowed to call a few operating system functions that are explicitly marked as reentrant-proof. Please pay attention to not make reentrant calls of local functions within the client application. As shown in the example, only a global flag should be set for the signaling within the callback handler. This flag will later on be evaluated and processed in the main loop of the client application.

13.2 API of the Shared Process Image Client

As illustrated in Figure 36, the user-specific C/C++ application exclusively uses the API (Application Programming Interface) provided by the shared process image client. This API is declared in the header file *shpimgclient.h* and implemented in the source file *shpimgclient.c*. It contains the following types (partly defined in *shpimg.h*) and functions:

Structure tShPImgLayoutDscrpt

```
typedef struct
{
    // definition of process image sections
    unsigned int    m_uiPImgInputOffs;    // start offset of input section
    unsigned int    m_uiPImgOutputOffs;    // size of input section
    unsigned int    m_uiPImgOutputSize;    // size of output section
    unsigned int    m_uiPImgOutputSize;    // size of output section
    unsigned int    m_uiPImgMarkerOffs;    // start offset of marker section
    unsigned int    m_uiPImgMarkerSize;    // size of marker section
```

```
} tShPImgLayoutDscrpt;
```

Structure *tShPImgLayoutDscrpt* describes the setup of the process image given by the PLC firmware. The client application receives the information about the setup of the process image via function *ShPImgCIntSetup()*. This function enters start offsets and values of input, output and marker sections into the structure provided upon function calling.

Structure tShPImgSectDscrpt

```
typedef struct
{
    // definition of data exchange section
    unsigned int    m_uiPImgDataSectOffs;
    unsigned int    m_uiPImgDataSectSize;
    tShPImgSyncType    m_SyncType;
    BOOL     m_fNewData;
```

// only used for WriteSectTab

} tShPImgSectDscrpt;

Structure **tShPImgSectDscrpt** describes the setup of a *ReadSectorTable* or *WriteSectorTable* entry that must be defined by the client. Both tables support the synchronization between the local image of the PLC runtime system and the shared process image (see section 13.1). Member element *m_uiPImgDataSectOffs* defines the absolute start offset of the section within the shared process images. The respective start offsets of the input, output and marker sections can be determined through structure *tShPImgLayoutDscrpt*. Member element *m_uiPImgDataSectSize* determines the size of the section which may include one or more variables. Member element *m_SyncType* only applies to entries of the WriteSectorTable. It determines whether the section is generally taken over from the shared process image into the local image whenever there are two successive PLC cycles (*kShPImgSyncAlways*) or whether it is taken over on demand (*kShPImgSyncOnDemand*). If classified as *SyncOnDemand*, the data must be marked as modified by calling function *ShPImgCIntWriteSectMarkNewData()*. It sets the member element *m_fNewData* to TRUE. The client application should never directly modify this member element.

Function ShPImgCIntSetup

BOOL ShPImgClntSetup (tShPImgLayoutDscrpt* pShPImgLayoutDscrpt_p);

Function **ShPImgCintSetup()** initializes the *shared process image client* and connects itself with the storage segment for the shared process image which is generated by the PLC runtime system. Afterwards, it enters the start offsets and values of the input, output and marker sections into the structure of type *tShPImgLayoutDscrpt* provided upon function call. Hence, the client application receives notice about the process image setup managed by the PLC firmware.

If the PLC runtime system is not active when the function is called or if it has not generated a shared process image (option "Share PLC process image" in the PLC configuration deactivated, see section 13.1), the function will return with the return value FALSE. If the initialization was successful, the return value will be TRUE.

Function ShPImgCIntRelease

BOOL ShPImgClntRelease (void);

Function **ShPImgCintRelease()** shuts down the shared process image client and disconnects the connection to the storage segment generated for the shared process image by the PLC runtime system.

If executed successfully, the function delivers return value TRUE. If an error occurs, it will deliver return value FALSE.

Function ShPImgCIntSetNewDataSigHandler

BOOL

ShPImgClntSetNewDataSigHandler tShPImgAppNewDataSigHandler pfnShPImgAppNewDataSigHandler_p);

Function **ShPImgCIntSetNewDataSigHandler()** registers a user-specific callback handler. This callback handler is called after a synchronization of both images. Registered callback handlers are cleared by the parameter NULL.

The **callback handler is called within the context of a Linux signal handler**. Accordingly, all common **restrictions** for the Linux signal handler also apply to the callback handler (see section 13.1).

If executed successfully, the function delivers return value TRUE. If an error occurs, it will deliver return value FALSE.

Function ShPImgCIntGetHeader

tShPImgHeader* ShPImgClntGetHeader (void);

Function **ShPImgCintGetHeader()** provides a pointer to the internally used structure type *tShPImgHeader* to manage the shared process image. The client application does usually not need this structure, because all data that it includes can be read and written through functions of the API provided by the *shared process image client*.

Function ShPImgCIntGetDataSect

BYTE* ShPImgClntGetDataSect (void);

Function **ShPImgCintGetDataSect()** provides a pointer to the beginning of the shared process image. This pointer represents the basic address for all accesses to the shared process image; including the definition of sections *ReadSectorTable* and *WriteSectorTable* (see section 13.1).

Funktionen Functions ShPImgCIntLockSegment and ShPImgCIntUnlockSegment

- BOOL ShPImgClntLockSegment (void);
- BOOL ShPImgClntUnlockSegment (void);

To exclusively access the shared process image, the client application has available two functions - function **ShPImgCIntLockSegment()** to enter the critical section and function **ShPImgCIntUnlockSegment()** to leave it. The segment between both functions is called protected section, because in this segment the client application holds unrivaled access to the shared process image (see section 13.1). The consistency of read or written data is only guaranteed within such a protected section. Outside the protected section, the shared process image may anytime be manipulated by the PLC runtime system. To ensure that the client application does not interfere with the PLC runtime system too much, the critical sections should be set as less as possible and only as long as necessary. Otherwise, the PLC cycle time may be extended and runtime variations (Jitter) may occur.

(

If executed successfully, the function delivers return value TRUE. If an error occurs, it will deliver return value FALSE.

Function ShPImgCIntSetupReadSectTable

BOOL

ShPImgClntSetupReadSectTable tShPImgSectDscrpt* unsigned int uiNumOfReadDscrptUsed_p); (paShPImgReadSectTab p,

Function **ShPImgCintSetupReadSectTable()** initializes the *ReadSectorTable* with the values defined by the client. The client hereby determines those sections of the PLC process image from which it wants to read data (see section 13.1). Parameter *paShPImgReadSectTab_p* holds elements of the structure *tShPImgSectDscrpt* and must be transferred as start address of a section. Parameter *uiNumOfReadDscrptUsed_p* indicates how many elements the section has.

kShPImgSyncAlways is provided as SyncType for the ReadSectorTable.

If executed successfully, the function delivers return value TRUE. If an error occurs, it will deliver return value FALSE.

Function ShPImgCIntSetupWriteSectTable

BOOL ShPImgClntSetupWriteSectTable (
tShPImgSectDscrpt* paShPImgWriteSectTab_p,
unsigned int uiNumOfWriteDscrptUsed_p);

Function **ShPImgCIntSetupWriteSectTable**() initializes the *WriteSectorTable* with the values defined by the client. The client hereby determines those sections of the PLC process image from which it wants to write data (see section 13.1). Parameter *paShPImgWriteSectTab_p* holds elements of structure *tShPImgSectDscrpt* and must be transferred as start address of a section. Parameter *uiNumOfWriteDscrptUsed_p* indicates how many elements the section has.

For each entry in the *WriteSectorTable* the *SyncType* must be defined. This *SyncType* defines whether the section is always taken over into the local image between two PLC cycles (*kShPImgSyncAlways*) or only on demand (*kShPImgSyncOnDemand*). If taken over on demand, the respective section is explicitly marked as updated by calling *ShPImgCIntWriteSectMarkNewData(*).

If executed successfully, the function delivers return value TRUE. If an error occurs, it will deliver return value FALSE.

Function ShPImgCIntWriteSectMarkNewData

BOOL ShPImgClntWriteSectMarkNewData (unsigned int uiWriteDscrptIdx_p);

For the content of a section that is held by the *WriteSectorTable*, function *ShPImgCIntWriteSectMarkNewData()* marks this content as modified. This function is used (for sections with *SyncType kShPImgSyncOnDemand*) to initiate the copy process of data from the shared process image into the local image of the PLC.

Function *ShPImgCIntWriteSectMarkNewData()* directly accesses the header of the shared process image without setting a semaphore before. Hence, it may only be used within the protected section – in the code section between *ShPImgCIntLockSegment()* and *ShPImgCIntUnlockSegment()*.

If executed successfully, the function delivers return value TRUE. If an error occurs, it will deliver return value FALSE.

13.3 Creating a User-Specific Client Application

Software package 3912005 ("Oracle VM VirtualBox-Image of the Linux development system") is the precondition for the implementation of user-specific C/C++ applications. It contains a complete Linux development system in the form of a VirtualBox image. Hence, it allows for an easy introduction into the C/C++ software development for the device. Thus, the VirtualBox image is the ideal basis to develop Linux-based user programs on the same host PC that already has the *OpenPCS* IEC 61131 programming system installed on it. The VirtualBox image of the Linux development system includes the GNU-Crosscompiler Toolchain for ARM processors. Additionally, it includes essential server services that are preconfigured and usable for effective software development.

As illustrated in Figure 36, the user-specific C/C++ application uses the API (files *shpimgclient.c* and *shpimgclient.h*) which is provided by the *shared process image client*. The *shared process image client* is based on services provided by the *shared memory client* (files *shmclient.c* and *shmclient.c*). Both client implementations are necessary to generate a user-specific C/C++ application. The directory of the *shared process image demo* contains the respective files. The path is depending on the device, e.g. /*projects/CTR-700/user/shpimgdemo*. To create own user-specific client applications, it is recommended to use this demo project as the basis for own adaptations and extensions. Moreover, this demo project contains a Makefile with all relevant configuration adjustments that are necessary to create a Linux application. Table 17 lists all files of the directory *shpimgdemo* and classifies those as general part of the C/C++ application or as specific component for the demo project "*shpimgdemo*".

File	Necessary for all C/C++ applications	In particular for demo "shpimgdemo"
shpimgclient.c	x	
shpimgclient.h	х	
shmclient.c	x	
shmclient.h	x	
shpimg.h	x	
global.h	x	
Makefile	draft, to be adjusted	
shpimgdemo.c		х
trmio.c		Х
trmio.h		x
trace.c		X

Table 17: Content of the archive files "shpimgdemo.tar.gz"

The demo project can be built by calling command "make". Here on the example of the CTR-700:

```
cd /projects/CTR-700/user/shpimgdemo make
```

-	shpimgdemo : bash — Konsole — + ×									×					
File	Edit	View	Bookmarks	Settings	Help										
vbox@v vbox@v	box-xı box-xı	ıbuntu: ıbuntu:	/projects/CT /projects/CT	R-700/usei R-700/usei	r\$ cd r/shpi	shpi mgde	mgder mo\$r	no/ nake	2						
Make S CFL DNDEBU LDF CTR-70 r_som_ re-in- LDL	etting AGS = G' LAGS = 0/var_ mx7_de shared IBS =]s = '-00 _som_mx ebian/ri I-libs' = ''	-g -Wall -Wn sroot=/proje 7_debian/roo ootfs/usr/li	no-pointer ects/CTR-70 utfs/lib/an b/arm-linu	∙sign)0/var m-lir µx-gnu	-Wno som nux-g neabi	-unu: _mx7 nueal hf -N	sed- _deb bihf Wl,-	but-s ian/r -L/p unres	set-v rootf proje solve	ariak s -L/ cts/(d-syn	ble - /proj CTR-7 nbols	I. ect: '00/' ≔ig	- va no	
Compil Compil Compil Compil Compil Linkin Stripp Done.	Compiling 'shpimgdemo.c' Compiling 'shpimgclient.c' Compiling 'shmclient.c' Compiling 'trmio.c' Compiling 'trace.c' Linking 'shpimgdemo' Stripping 'shpimgdemo' Done.														
Copy e	xecute	eable '	shpimgdemo'	to destina	ation	'/tf	tpbo								
vbox@v	/box@vbox-xubuntu:/projects/CTR-700/user/shpimgdemo\$														
\$_		shpir	ngdemo : bash	1											

Figure 37: Generating the demo project "shpimgdemo" in the Linux development system

Section 13.4 describes the usage and handling of the demo project "shpimgdemo".

13.4 Example for Using the Shared Process Image

The demo project "*shpimgdemo*" (described in section 13.3) in connection with the PLC program example "*RunLight*" both exemplify the data exchange between a PLC program and a user-specific C/C++ application.

Technical Background

The PLC program generates some variables in the process image as directly addressable variables. In a C/C++ application, all those variables are usable via the shared process image. For the PLC program example "*RunLight*" those are the following variables:

```
(* variables for local control via on-board I/O's *)
bButtonGroup AT %IB0.0 : BYTE;
iAnalogValue AT %IW8.0 : INT;
bLEDGroup0
                      AT %QB0.0 : BYTE;
bLEDGroup1
                      AT %QB1.0
                                    : BYTE;
(* variables for remote control via shared process image *)
uiRemoteSlidbarLen AT %MW512.0 : UINT; (* out: length of slidebar
                                                                                             *)
bRemoteStatusAT %MB514.0 : BYTE;bRemoteDirCtrlAT %MB515.0 : BYTE;
                                                   (* out: Bit0: RemoteControl=on/off *)
                                                 (* in: direction left/right
(* in: speed
bRemoteDirCtrlAT %MB515.0 : BYTE;iRemoteSpeedCtrlAT %MW516.0 : INT;
                                                                                            *)
                                                                                            *)
```

Variables of the PLC program are accessible from a C/C++ application via the shared process image. Therefore, sections must be generated for the *ReadSectorTable* and *WriteSectorTable* on the one hand and on the other hand, pointers must be defined for accessing the variables. The following program extract shows this using the example "*shpimgdemo.c*". Function *ShPImgCIntSetup()* inserts the start

© SYSTEC electronic AG 2022	L-2562e_01	Page 78
-----------------------------	------------	---------

offsets of input, output and marker sections into the structure *ShPImgLayoutDscrpt*. Hence, on the basis of the initial address provided by *ShPImgCIntGetDataSect()*, the absolute initial addresses of each section in the shared process image can be determined. To identify the address of a variable, the variable's offset within the particular section must be added. For example, the absolute address to access the variable "*bRemoteDirCtrl AT %MB515.0 : BYTE;*" results from the sum of the initial address of the shared process image (pabShPImgDataSect), the start offset of the marker section (*ShPImgLayoutDscrpt.m_uiPImgMarkerOffs für "%M..."*) as well as the direct address within the marker section which was defined in the PLC program (515 for "%MB515.0"):

```
pbPImgVar_61131_bDirCtrl = (BYTE*) (pabShPImgDataSect
+ ShPImgLayoutDscrpt.m_uiPImgMarkerOffs + 515);
```

The following code extract shows the complete definition of all variables in the demo project used for exchanging data with the PLC program:

```
// ---- Setup shared process image client ----
fRes = ShPImgClntSetup (&ShPImgLayoutDscrpt);
if ( !fRes )
{
   printf ("\n*** ERROR *** Init of shared process image client failed");
}
pabShPImgDataSect = ShPImgClntGetDataSect();
// ---- Read Sector Table ----
// Input Section: bButtonGroup AT %IB0.0
ł
    ShPImgReadSectTab[0].m uiPImgDataSectOffs
               ShPImgLayoutDscrpt.m uiPImgInputOffs + 0;
    ShPImgReadSectTab[0].m uiPImgDataSectSize = sizeof(BYTE);
   ShPImgReadSectTab[0].m_SyncType
                                            = kShPImgSyncAlways;
   pbPImgVar_61131_bButtonGroup
                                                (BYTE*)
                                                              (pabShPImgDataSect
                                     =
              + ShPImgLayoutDscrpt.m uiPImgInputOffs + 0);
}
// Output Section: bLEDGroup0 AT %QB0.0
                     bledgroup1 AT %QB1.0
11
{
    ShPImgReadSectTab[1].m uiPImgDataSectOffs
               ShPImgLayoutDscrpt.m uiPImgOutputOffs + 0;
    ShPImgReadSectTab[1].m uiPImgDataSectSize = sizeof(BYTE) + sizeof(BYTE);
   ShPImgReadSectTab[1].m SyncType = kShPImgSyncAlways;
   pbPImgVar 61131 bLEDGroup0
                                    =
                                                (BYTE*)
                                                               (pabShPImgDataSect
              + ShPImgLayoutDscrpt.m_uiPImgOutputOffs + 0);
   pbPImgVar 61131 bLEDGroup1 =
                                               (BYTE*)
                                                               (pabShPImgDataSect
              + ShPImgLayoutDscrpt.m uiPImgOutputOffs + 1);
}
// Marker Section: uiSlidbarLen AT %MW512.0
                      bStatus AT %MB514.0
11
ł
   ShPImgReadSectTab[2].m uiPImgDataSectOffs
               ShPImgLayoutDscrpt.m uiPImgMarkerOffs + 512;
   ShPImgReadSectTab[2].m uiPImgDataSectSize = sizeof(unsigned short int)
                                             + sizeof(BYTE);
   ShPImgReadSectTab[2].m SyncType
                                            = kShPImgSyncAlways;
   pbPImgVar_61131_usiSlidbarLen = (unsigned short int*)
                                                              (pabShPImgDataSect
              + ShPImgLayoutDscrpt.m_uiPImgMarkerOffs + 512);
                                              (BYTE*)
   pbPImgVar_61131_bStatus
                                                               (pabShPImgDataSect
                                   =
              + ShPImgLayoutDscrpt.m uiPImgMarkerOffs + 514);
}
```

```
fRes = ShPImgClntSetupReadSectTable (ShPImgReadSectTab, 3);
if ( !fRes )
ł
   printf ("\n*** ERROR *** Initialization of read sector table failed");
ļ
// ---- Write Sector Table ----
// Marker Section: bDirCtrl
                                 AT %MB515.0
11
                       iSpeedCtrl AT %MB516.0
{
   ShPImgWriteSectTab[0].m uiPImgDataSectOffs
                ShPImgLayoutDscrpt.m uiPImgMarkerOffs + 515;
   ShPImgWriteSectTab[0].m_uiPImgDataSectSize = sizeof(BYTE) + sizeof(WORD);
   ShPImgWriteSectTab[0].m SyncType
                                              = kShPImqSyncOnDemand;
   pbPImgVar_61131 bDirCtrl
                                                (BYTE*)
                                                                 (pabShPImgDataSect
                                    =
              + ShPImgLayoutDscrpt.m uiPImgMarkerOffs + 515);
   psiPImgVar 61131 iSpeedCtrl = (short
                                                     int*)
                                                                 (pabShPImgDataSect
                   + ShPImgLayoutDscrpt.m_uiPImgMarkerOffs + 516);
}
fRes = ShPImgClntSetupWriteSectTable (ShPImgWriteSectTab, 1);
if ( !fRes )
{
   printf ("\n*** ERROR *** Initialization of write sector table failed");
}
```

Realization

To enable the execution of the *shared process image demo* without previous introduction into the Linuxbased C/C++ programming for the device, the module comes with a preinstalled, translated and readyto-run program version and PLC firmware (*"/home/plc/bin/shpimgdemo"*). The following description refers to this program version. Alternatively, the demo project can be newly-generated from the corresponding source files (see section 13.3) and can be started afterwards. As I/O-Simulator for practical controlling of the demo-program an I/O-Box is available from SYS TEC.

The following steps are necessary to run the shared process image demo:

- 1. Activate option "Shared PLC process image" in the PLC configuration (see sections 13.1, 7.6.1 and 7.6.3).
- 2. Open the PLC program example "*RunLight*" in the *OpenPCS* IEC 61131 programming system und build the project for a target hardware of the type, e.g. "SYSTEC CTR-700"
- 3. Select the network connection to the device and download the program.
- 4. Start the PLC program on the device.
- 5. Login to the command shell as described in section 7.9.1.
- 6. Switch to the directory "/home/plc/bin" and call the demo program "shpimgdemo":

cd ./shpimgdemo /home/plc/bin

The digital outputs of the device are selected as runlight. The speed is modifiable via the analog input Al0. With the help of digital inputs DI0 and DI1, the running direction can be changed. After starting the demo program *"shpimgdemo"*, actual status information about the runlight is indicated cyclically in the terminal (see Figure 38).

COM3:115200baud - Tera Term VT	
D <u>a</u> tei B <u>e</u> arbeiten Ein <u>s</u> tellungen Ste <u>u</u> erung <u>F</u> enster <u>H</u> ilfe	
манны и интерновальный и интерновальный и интерновальный и интерновальный и интерновальный и интерновальный и Shared Process Inage demo application for SYSTEC PLCnodule-C34 Version: 1.00 (c) 2011-2016 SYS TEC electronic GnbH, ини.systec-electronic.com	
Setup shared process inage client Shared process inage layout: InputOffs: 0000 InputSize: 2048 OutputOffs: 2048	
OutputSize: 2048 HarkerOffs: 4096 HarkerSize: 4086 pShPIngHeader = 0x76F09000 pabShPIngUatSect = 0x76F09138 Register signal handler Setup read and urite sector table Pointer to process inage valables: pPIngVar_61131 bButtonGroup = 0x76F09138 pbPIngVar_61131 bLEDGroup0 = 0x76F0938 pbPIngVar_61131 bLEDGroup1 = 0x76F0938 pbPIngVar_61131 bLEDGroup1 = 0x76F0838 pbPIngVar_61131 bSittus = 0x76F0838 pbPIngVar_61131 bDirtCtrl = 0x76F0838 psIPIngVar_61131_iSpeedCtrl = 0x76F0838 Run program cycle (exchange process inage)	
SET START PARAMETER: Dir=Left, Speed=D Slidebar(8): ***	-

Figure 38: Terminal outputs of the demo program "shpimgdemo" after start

7. By activating of digital input DI3, the control of the runlight direction and speed is handed over to the demo program "shpimgdemo". Afterwards, the running direction may be set by the C application by using the cursor pushbuttons left and right (← and →) in the terminal window and the speed may be changed by using cursor pushbuttons up and down (↑ and ↓).

ſ	🧕 COM3:11	5200baud -	Tera Term	VT			×
	D <u>a</u> tei B <u>e</u> ar	beiten Ein <u>s</u>	tellungen	Ste <u>u</u> erung	<u>F</u> enster	<u>H</u> ilfe	
	Pointer to pro pbPIngVar_ pbPIngVar_ pbPIngVar_ pbPIngVar_ pbPIngVar_ psiPIngVar Run progran cy SET START PARA	cess image va 61131 bButton 61131 bLEDGro 61131 bLEDGro 61131 bStatus 61131 bStatus 61131 bDirCtr 61131 iSpeed cle (exchange HETER: Dir=L	iables: Group = Ox upO = Ox barLen = Ox = Ox 1 = Ox Ctrl = Ox process im	76F32138 76F32938 76F32939 76F33338 76F33338 76F33338 76F3333C age)			
	RemoteControl Slidebar(8):	= enabled ***					L
	ButtonGroup=Dx Slidebar(8):	00 .***					
	SET NEH PARAHE Slidebar(8):	TER: Dir=Lef	t, Speed=1				
	SET NEH PARAME Slidebar(8):	TER: Dir=Lef ****	t, Speed=2				
	SET NEH PARAME Slidebar(8):	TER: Dir=Lef ***	t, Speed=3				
	SET NEH PARAHE Slidebar(8):	TER: Dir=Lef	t, Speed=4				
	SET NEH PARAME Slidebar(8):	TER: Dir=Lef .***∎	t, Speed=5				
							-

Figure 39: Terminal outputs of the demo program "shpimgdemo" after user inputs

Figure 39 shows the terminal outputs of the demo program "shpimgdemo" in answer to activating the cursor pushbuttons.

The demo program "shpimgdemo" may be terminated by pressing "Ctrl+C" in the terminal window.

14 Troubleshooting

If there are any problems with the device, which are not covered by this section or any other part of this document, please contact our support under +49 (0) 37 65 / 38 600-0 or <u>support@systec-electronic.com</u>.

Error Behavior:

The shown time or time zone on the device is wrong. This may include some applications like Node-RED.

NLD.

Cause:

The device has no internal battery, which means after a certain time, some settings may get lost.

Solution:

If the device has access to the internet, the time will set itself on the next restart. If this is not the case, one has to set the time manually, described in Section 7.12.

Appendix A: Firmware Function Scope

Table 18 lists all firmware functions and function blocks available on the device.

Sign explanation:

FB	Function block
FUN	Function
Online Help	OpenPCS online help
L-1054	Manual "SYS TEC-specific extensions for OpenPCS / IEC 61131-3", Manual no.:
	L-1054)
PARAM:={0,1,2}	values 0, 1 and 2 are valid for the given parameter

Table 18: Firmware functions and function blocks

Name	Туре	Reference	Remark					
PLC standard Functions and Function Blocks								
SR	FB	Online Help						
RS	FB	Online Help						
R_TRIG	FB	Online Help						
F_TRIG	FB	Online Help						
СТU	FB	Online Help						
CTD	FB	Online Help						
CTUD	FB	Online Help						
TP	FB	Online Help						
TON	FB	Online Help						
TOF	FB	Online Help						

Functions and Function Blocks for string manipulation							
ETRC	FB	L-1054					
PTRC	FB	L-1054					
GETVARPOINTER	FB	L-1054					
BIN_TO_STR	FUN	L-1054					
STR_TO_BIN	FUN	L-1054					
OBJ_TO_STR	FB	L-1054					
GETSTRINFO	FB	L-1054					
CHR	FUN	L-1054					
ASC	FUN	L-1054					
STR	FUN	L-1054					
VAL	FUN	L-1054					
LEN	FUN	L-1054					
LEFT	FUN	L-1054					
RIGHT	FUN	L-1054					
MID	FUN	L-1054					
CONCAT	FUN	L-1054					
INSERT	FUN	L-1054					
DELETE	FUN	L-1054					
REPLACE	FUN	L-1054					
FIND	FUN	L-1054					
STR UPPER	FUN						
STR LOWER	FUN						
STR TRIM	FUN						
Functions and Function Blocks for Open	PCS spe	cific task contr	olling				
GETVARDATA	FB	Online Help					
GETVARFLATADDRESS	FB	Online Help					
GETTASKINFO	FB	Online Help					
Functions and Function Blocks for handl	ing of no	on-volatile data					
NVDATA_BIT	FB	L-1054	DEVICE:={0} see (1)				
NVDATA_INT	FB	L-1054	DEVICE:={0} see (1)				
NVDATA_STR	FB	L-1054	DEVICE:={0} see (1)				
NVDATA_BIN	FB	L-1054	DEVICE:={0} see (1)				
Functions and Function Blocks for handl	ing of tir	ne	•				
GETTIME	FUN	Online Help					
GETTIMECS	FUN	Online Help					
TIME TO DINT	FUN						
DINT TO TIME	FUN						
DT CLOCK	FB	L-1054					
DT ABS TO REL	FB	L-1054					
DT REL TO ABS	FB	L-1054					
DT REL TO DT							
Functions and Function Blocks for count	er input	s and pulse out	puts				
CNT FUD	FB	L-1054	CHANNEL:={0,1.2}				
PTO PWM	FB	L-1054	CHANNEL:={0.1}				
ΡΤΟ ΤΑΒ	FB	L-1054	CHANNEL :={0,1}				
Function Block for PID regulator		2.001	·····				
		1	T				
1טוץ	FB	L-1054					

L-2562e_01

Functions and Function Blocks for Serial interfaces							
SIO_INIT	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_STATE	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_READ_CHR	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_WRITE_CHR	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_READ_STR	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_WRITE_STR	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_READ_BIN	FB	L-1054	PORT:={0,1,2,3} see (2)				
SIO_WRITE_BIN	FB	L-1054	PORT:={0,1,2,3} see (2)				
Functions and Function Blocks for CAN i	interface	s / CANopen					
CAN_GET_LOCALNODE_ID	FB	L-1008	NETNUMBER:={0,1}				
CAN_CANOPEN_KERNEL_STATE	FB	L-1008	NETNUMBER:={0,1}				
CAN_REGISTER_COBID	FB	L-1008	NETNUMBER:={0,1}				
CAN_PDO_READ8	FB	L-1008	NETNUMBER:={0,1}				
CAN_PDO_WRITE8	FB	L-1008	NETNUMBER:={0,1}				
CAN_SDO_READ8	FB	L-1008	NETNUMBER:={0,1}				
CAN_SDO_WRITE8	FB	L-1008	NETNUMBER:={0,1}				
CAN_SDO_READ_STR	FB	L-1008	NETNUMBER:={0,1}				
CAN_SDO_WRITE_STR	FB	L-1008	NETNUMBER:={0,1}				
CAN_SDO_READ_BIN	FB	L-1008	NETNUMBER:={0,1}				
CAN_SDO_WRITE_BIN	FB	L-1008	NETNUMBER:={0,1}				
CAN_GET_STATE	FB	L-1008	NETNUMBER:={0,1}				
CAN_NMT	FB	L-1008	NETNUMBER:={0,1}				
CAN_RECV_EMCY_DEV	FB	L-1008	NETNUMBER:={0,1}				
CAN_RECV_EMCY	FB	L-1008	NETNUMBER:={0,1}				
CAN_WRITE_EMCY	FB	L-1008	NETNUMBER:={0,1}				
CAN_RECV_BOOTUP_DEV	FB	L-1008	NETNUMBER:={0,1}				
CAN_RECV_BOOTUP	FB	L-1008	NETNUMBER:={0,1}				
CAN_ENABLE_CYCLIC_SYNC	FB	L-1008	NETNUMBER:={0,1}				
CAN_SEND_SYNC	FB	L-1008	NETNUMBER:={0,1}				
CANL2_INIT	FB	L-1008	NETNUMBER:={0,1} see (3)				
CANL2_SHUTDOWN	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_RESET	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_GET_STATUS	FB	L-1008	NETNUMBER:={0,1} see (3)				
CANL2_DEFINE_CANID	FB	L-1008	NETNUMBER:={0,1} see (3)				
CANL2_DEFINE_CANID_RANGE	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_UNDEFINE_CANID	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_UNDEFINE_CANID_RANGE	FB	L-1008	NETNUMBER:={0,1} see (3)				
CANL2_MESSAGE_READ8	FB	L-1008	NETNUMBER:={0,1} see (3)				
CANL2_MESSAGE_READ_BIN	FB	L-1008	NETNUMBER:={0,1} see (3)				
CANL2_MESSAGE_WRITE8	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_MESSAGE_WRITE_BIN	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_MESSAGE_UPDATE8	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				
CANL2_MESSAGE_UPDATE_BIN	FB	L-1008	NETNUMBER:={0,1} see ⁽³⁾				

Functions and Function Blocks for Ethernet interfaces / UDP							
LAN_GET_HOST_CONFIG	FB	L-1054	NETNUMBER:={0}				
LAN_ASCII_TO_INET	FB	L-1054	NETNUMBER:={0}				
LAN_INET_TO_ASCII	FB	L-1054	NETNUMBER:={0}				
LAN_GET_HOST_BY_NAME	FB	L-1054	NETNUMBER:={0}				
LAN_GET_HOST_BY_ADDR	FB	L-1054	NETNUMBER:={0}				
LAN_UDP_CREATE_SOCKET	FB	L-1054	NETNUMBER:={0}				
LAN_UDP_CLOSE_SOCKET	FB	L-1054	NETNUMBER:={0}				
LAN_UDP_RECVFROM_STR	FB	L-1054	NETNUMBER:={0}				
LAN_UDP_SENDTO_STR	FB	L-1054	NETNUMBER:={0}				
LAN_UDP_RECVFROM_BIN	FB	L-1054	NETNUMBER:={0}				
LAN_UDP_SENDTO_BIN	FB	L-1054	NETNUMBER:={0}				
Functions and Function Blocks for file access							
FILE_OPEN	FB	L-1828					
FILE_CLOSE	FB	L-1828					
FILE_READ	FB	L-1828					
FILE_READ_LINE	FB	L-1828					
FILE_WRITE	FB	L-1828					
FILE_SEEK	FB	L-1828					
FILE_SYNC	FB	L-1828					
FILE_STAT	FB	L-1828					
FILE_CHMOD	FB	L-1828					
FILE_TOUCH	FB	L-1828					
FILE_DELETE	FB	L-1828					
FILE_RENAME	FB	L-1828					
FILE_COPY	FB	L-1828					
FILE_SPLIT_PATH	FB	L-1828					
FILE_DIR_OPEN	FB	L-1828					
FILE_DIR_CLOSE	FB	L-1828					
FILE_DIR_READ	FB	L-1828					
FILE_GET_DIR	FB	L-1828					
FILE_SET_DIR	FB	L-1828					
FILE_MKDIR	FB	L-1828					
FILE_RMDIR	FB	L-1828					
FILE_MKFIFO	FB	L-1828					
FILE_EXEC_SYS_CMD	FB	L-1828					
FTYPE_TO_UINT	FUN	L-1828					
FSEEK_TO_UINT	FUN	L-1828					
FPERM_TO_STRING	FUN	L-1828					
SYSERR_TO_STRING	FUN	L-1828					

MODBUS_OPEN_INSTANCEFMODBUS_CLOSE_INSTANCEFMODBUS_REGISTER_VAR_LISTF	-B -B -B -B	L-1829 L-1829			
MODBUS_CLOSE_INSTANCE F MODBUS_REGISTER_VAR_LIST F	-B -B -B	L-1829			
MODBUS_REGISTER_VAR_LIST F	-в -в	1 1820			
	⁼В	L-1029			
MODBUS_READ_REGS		L-1829			
MODBUS_WRITE_SINGLE_REG F	⁼В	L-1829			
MODBUS_WRITE_MULTI_REGS F	⁼В	L-1829			
MODBUS_READ_WRITE_REGS F	-В	L-1829			
MODBUS_READ_INPUT_REGS F	⁼В	L-1829			
MODBUS_READ_DISCR_INPUTS F	⁼В	L-1829			
MODBUS_READ_COILS F	⁼В	L-1829			
MODBUS_WRITE_SINGLE_COIL F	⁼В	L-1829			
MODBUS_WRITE_MULTI_COILS F	⁼В	L-1829			
MODBUS_RAW_PDU_REQUEST F	⁼В	L-1829			
Functions and Function Blocks for Modbus	comm	nunication			
MQTT_GET_CAPABILITIES F	-B	Demo			
MQTT_CONNECT F	⁼В	Demo			
MQTT_DISCONNECT F	⁼В	Demo			
MQTT_GET_CONNECT_STATE F	-UN	Demo			
MQTT_SUBSCRIBE F	⁼В	Demo			
MQTT_UNSUBSCRIBE F	⁼В	Demo			
MQTT_GET_ARRIVED_MESSAGE F	⁼В	Demo			
MQTT_PUBLISH F	-В	Demo			
Functions and Function Blocks for Modbus communication					
WPC_CONNECT F	Ъ				
WPC_DISCONNECT F	-B				
WPC_READ_FRAME F	Ъ				
WPC_GET_STATE F	FВ				

- (1) All nonvolatile data is filed into directory "/home/plc/plcdata/PlcPData.bin". This file has a fix size of 32 KiB. By calling function blocks of type NVDATA_Xxx in a writing mode, the modified data is directly stored into file "/home/plc/plcdata/PlcPData.bin" ("flush"). Thus, unsecured data is not getting lost in case of power interruption.
- ⁽²⁾ Interface SERVICE (PORT:=3) primarily serves as service interface to administer the device. Hence, this interface should only be used for sign output. The module always tries to interpret and execute sign inputs as Linux commands (see section 7.9.1).
- ⁽³⁾ The usage of Function Blocks from type CANL2_Xxx is only possible, if the according CAN interface is not used already by CANopen. Due to its necessary to disable the according CAN interface in the PLC configuration (see section 7.6.1), otherwise the Function Blocks from type CANL2_Xxx can't be used. Alternatively, entry "Enable=" can directly be set to 0 within section "[CANx]" of the configuration file (see section 7.6.3).

Appendix B: Technical Specification

Environmental Paramet	ers	Typical	Minimum Maximum
Power Supply	VCPU	24VDC	19.230VDC
	V _{IO}	24VDC	19.230VDC
	power fail level	18,2V	
	power fail delay time	10ms	
Current Consumption	Ісри	100mA	
(inactive IOs)	lio	30mA	
Temperature Range	Storage temperature		-20+70°C
	Operating temperature		055°C
MTTF	According to SN29500 at 40°C		> 373134 h
Protection class	Housing	IP20	
Weight	without any cable and packing	295g	
Dimensions	Width		180mm
	Height		61mm
	Depth		91mm
Connector type	Spring type connector		
Lowest cycle time for PLO	C	5 ms	

I/O-configuration (digital)		Typical	Maximum	
Digital Outputs DO	0 3			
24VDC-Output (Side Switch)	High	U_{OH} at I_{OH} = 500 mA	V _{IO} -0.12V < U _{OH} <v<sub>IO</v<sub>	
		U _{OL} at I _{OL} = 0 mA		1 V
		Current limitation IOH_max		700 mA
		Max. current		4x0,5A
		Impedance		0.08 Ohm
		I _{OL(off)}		5 µA
		t _{off} at I _{OH} = 500 mA	30 µs	35 µs
		t _{on} at I _{OH} = 500 mA	20 µs	35 µs
		Frequency	ca. 200 Hz	1 KHz⁴
PWM Output (DO2 + DO3)				
24VDC-Output (High	Tjitter		25 µs
Side Switch)	-	Ton_min and min. pulse width		800 ns
		Frequency		1 kHz
Digital Inputs DI0	3			
24VDC- Inputs,		Uін	12.6V	30V
plus switching		UIL	-3V	12V
		Iін (Vіл=6.7V)	0.9mA	
		Iін (Vіл=30V)		5.5mA
		Input type according to IEC61131-2	Туре 1	
		T _{DLY}		100ns
Counter Input and	Step	Direction (DI2 + DI3)		
24VDC- Inputs,	-	Min. pulse width		25 µs
plus switching		Frequency		10 kHz
A/B-Encoder (DI2 +	DI3)		•	•
24VDC- Inputs,	í	Frequency		10 kHz
plus switching		Phase Margin (A/B-Encoder)		±45°

⁴ Frequency limit of hardware interface circuit. The actually frequency may vary due to software limitations.

Communication Interfaces		Minimum	Maximum
CAN-Bus			
CAN1	Baudrate	5kBaud	1Mbaud
	Max. number of nodes		64
	CAN-H, CAN-L short-circuit-proof towards 24V		
RS-232/RS-485			
SERIAL0	Baudrate	1200Baud	115200Baud
SERIAL1	Baudrate	1200Baud	115200Baud
SERVICE	Baudrate	1200Baud	115200Baud
Ethernet			
ETH0	Bandwith	10Mbit/s	1000Mbit/s
ETH1	Bandwith	10Mbit/s	1000Mbit/s

Appendix C: Third Party Software Components GNU General Public License

The Embedded Linux used on device is licensed under GNU General Public License, version 2. The entire license text is specified below. A German translation is available from http://www.gnu.de/documents/gpl-2.0.de.html. Be advised that this translation is not official or legally approved.

The PLC system used and the PLC and C/C++ programs developed by the user are **not** subject to the GNU General Public License!

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software -- to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

© SYSTEC electronic AG 2022	L-2562e_01	Page 93
-----------------------------	------------	---------

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

L-2562e_01

© SYSTEC electronic AG 2022	
-----------------------------	--

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items -- whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Paho MQTT Embedded/C

The Eclipse Paho MQTT package is a client library for MQTT embedded devices.

Project URL: https://github.com/eclipse/paho.mgtt.embedded-c

Copyright and license:

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mono

Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

Project URL: https://github.com/mono/mono

License:

In general, the runtime and its class libraries are licensed under the terms of the MIT license, and some third party code is licensed under the 3-clause BSD license. See the file "PATENTS.TXT" for Microsoft's patent grant on the Mono codebase.

The Mono distribution does include a handful of pieces of code that are used during the build system and are covered under different licenses, those include:

Build Time Code

This is code that is used at build time, or during the maintenance of Mono itself, and does not end up in the redistributable part of Mono:

* gettext

m4 source files used to probe features at build time: GPL

* Benchmark Source Files

Logic.cs and zipmark.cs are GPL source files.

* mono/docs/HtmlAgilityPack

MS-PL licensed

- * mcs/jay: 4-clause BSD licensed
- * mcs/nunit24: MS-PL
- * mcs/class/I18N/mklist.sh, tools/cvt.sh: GNU GPLv2

Runtime Code

The following code is linked with the final Mono runtime, the libmono embeddable runtime:

* support/minizip: BSD license.

* mono/utils/memcheck.h: BSD license, used on debug builds that use Valgrind.

* mono/utils/freebsd-dwarf.h, freebsd-elf_common.h, freebsd-elf64.h freebsdelf32.h: BSD license.

- * mono/utils/bsearch.c: BSD license.
- * mono/metadata/w32file-unix-glob.c, w32file-unix-glob.h: BSD license

Class Library code

These are class libraries that can be loaded by your process:

 \star mcs/class/RabbitMQ.Client: dual licensed in Apache v2, and Mozilla Public License 1.1

- * mcs/class/Compat.ICSharpCode.SharpZipLib and mcs/class/ICSharpCode.SharpZipLib are GPL with class-path exception. Originates with the SharpDevelop project.
- * mcs/class/System.Core/System/TimeZoneInfo.Android.cs

This is a port of Apache 2.0-licensed Android code, and thus is licensed under the Apache 2.0 license

http://www.apache.org/licenses/LICENSE-2.0

API Documentation

The API documentation is licensed under the terms of the Creative Commons Attribution 4.0 International Public License

The Licenses

These are the licenses used in Mono, the files are located:

MIT X11 License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Mozilla.MPL

MOZILLA PUBLIC LICENSE Version 1.1

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

© SYSTEC electronic AG 2022	L-2562e_01	Page 98
	—	5

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this

© SYSTEC electronic AG 2022	L-2562e_01	Page 99
-----------------------------	------------	---------

License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the

© SYSTEC electronic AG 2022	L-2562e_01	Page 100
-----------------------------	------------	----------

Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

© SYSTEC electronic AG 2022	L-2562e_01	Page 101
-----------------------------	------------	----------

(b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

© SYSTEC electronic AG 2022	L-2562e_01	Page 102
-----------------------------	------------	----------

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

© SYSTEC electronic AG 2022	L-2562e_01	Page 103
-----------------------------	------------	----------

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW

L-2562e_01	Page 104
	L-2562e_01

PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations

© SYSTEC electronic AG 2022	L-2562e_01	Page 105
-----------------------------	------------	----------

under the License.

The Original Code is

The Initial Developer of the Original Code is ______ Portions created by ______ are Copyright (C) ______ . All Rights Reserved.

Contributor(s):

Alternatively, the contents of this file may be used under the terms of the ______license (the "[____] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

Microsoft Public License

Microsoft Permissive License (Ms-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

(F) If you distribute the software or derivative works with programs you develop, you agree to indemnify, defend, and hold harmless all contributors from any claims, including attorneys' fees, related to the distribution or use of your programs. For clarity, you have no such obligations to a contributor for any claims based solely on the unmodified contributions of that contributor.

(G) If you make any additions or changes to the original software, you may only distribute them under a new namespace. In addition, you will clearly identify your changes or additions as your own.

Infozip BSD

This is version 2009-Jan-02 of the Info-ZIP license. The definitive version of this document should be available at ftp://ftp.info-zip.org/pub/infozip/license.html indefinitely and a copy at http://www.info-zip.org/pub/infozip/license.html.

Copyright (c) 1990-2009 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals: Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ed Gordon, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Steven M. Schweda, Christian Spieler, Cosmin Truta, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White.

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the above disclaimer and the following restrictions:

Redistributions of source code (in whole or in part) must retain the above copyright notice, definition, disclaimer, and this list of

© SYSTEC electronic AG 2022	L-2562e_01	Page 107
-----------------------------	------------	----------

conditions.

Redistributions in binary form (compiled executables and libraries) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution. Additional documentation is not needed for executables where a command line license option provides these and a note regarding this option is in the executable's startup banner. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled.

Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, versions with modified or added functionality, and dynamic, shared, or static library versions not from Info-ZIP--must be plainly marked as such and must not be misrepresented as being the original source or, if binaries, compiled from the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or the Info-ZIP URL(s), such as to imply Info-ZIP will provide support for the altered versions.

Info-ZIP retains the right to use the names "Info-ZIP," "Zip,"
"UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip"
for its own source and binary releases.

License Creative Commons 2.5 // Copyright 2006 James Newton-King // http://www.newtonsoft.com // // This work is licensed under the Creative Commons Attribution 2.5 License // http://creativecommons.org/licenses/by/2.5/ 11 // You are free: 11 * to copy, distribute, display, and perform the work * to make derivative works 11 * to make commercial use of the work 11 11 // Under the following conditions: 11 * For any reuse or distribution, you must make clear to others the license terms of this work. * Any of these conditions can be waived if you get permission from the 11 copyright holder. james.newtonking@gmail.com [mailto:james.newtonking@gmail.com] From: On Behalf Of James Newton-King Sent: Tuesday, June 05, 2007 6:36 AM To: Konstantin Triger Subject: Re: Support request by Konstantin Triger for Json.NET

Hey Kosta

I think it would be awesome to use Json.NET in Mono for System.Web.Extensions.

© SYSTEC electronic AG 2022	L-2562e_01	Page 108
-----------------------------	------------	----------
The CC license has the following clause: Any of the above conditions can be waived if you get permission from the copyright holder.

I can waive that statement for you and Mono. Would that be acceptable?

Regards, James

Creative Commons Attribution 4.0 International Public License

Attribution 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CClicensed material, or material used under an exception or limitation to copyright. More considerations for licensors: wiki.creativecommons.org/Considerations for licensors

Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason--for example, because of any applicable exception or limitation to copyright--then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More_considerations

© SYSTEC electronic AG 2022

for the public: wiki.creativecommons.org/Considerations for licensees

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

- a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
- c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.
- d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.
- f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.
- g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

© SYSTEC electronic AG 2022	L-2562e_01	Page 110
		0

- i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
- k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

- a. License grant.
 - Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
 - a. reproduce and Share the Licensed Material, in whole or in part; and
 - b. produce, reproduce, and Share Adapted Material.
 - Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
 - 3. Term. The term of this Public License is specified in Section 6(a).
 - 4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.
 - 5. Downstream recipients.
 - a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.
 - b. No downstream restrictions. You may not offer or impose

© SYSTEC electronic AG 2022	L-2562e_01	Page 111
-----------------------------	------------	----------

any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

- 6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).
- b. Other rights.
 - Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
 - 2. Patent and trademark rights are not licensed under this Public License.
 - 3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

- a. Attribution.
 - If You Share the Licensed Material (including in modified form), You must:
 - a. retain the following if it is supplied by the Licensor with the Licensed Material:
 - i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
 - ii. a copyright notice;
 - iii. a notice that refers to this Public License;
 - iv. a notice that refers to the disclaimer of
 warranties;
 - v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

© SYSTEC electronic AG 2022	L-2562e_01	Page 112
-----------------------------	------------	----------

- b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
- c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
- 2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
- If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.
- If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

- a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.
- b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT,

© SYSTEC electronic AG 2022	L-2562e_01	Page 113
-----------------------------	------------	----------

INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
 - automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
 - 2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
- d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision

© SYSTEC electronic AG 2022	L-2562e_01	Page 114
-----------------------------	------------	----------

cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

- c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
- d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." The text of the Creative Commons public licenses is dedicated to the public domain under the CCO Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

GPL version 2

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it

© SYSTEC electronic AG 2022	L-2562e_01	Page 115
-----------------------------	------------	----------

in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and

© SYSTEC electronic AG 2022	L-2562e_01	Page 116
-----------------------------	------------	----------

you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
 source code, which must be distributed under the terms of Sections
 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer

ə 117
e

to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to

© SYSTEC electronic AG 2022	L-2562e_01	Page 118
-----------------------------	------------	----------

apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

© SYSTEC electronic AG 2022	L-2562e_01	Page 119
-----------------------------	------------	----------

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

© SYSTEC electronic AG 2022	L-2562e_01	Page 120
-----------------------------	------------	----------

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Link: https://github.com/mono/mono/blob/master/LICENSE

Microsoft Patent Promise for Mono

Microsoft Corporation and its affiliates ("Microsoft") promise not to assert any Applicable Patents against you for making, using, selling, offering for sale, importing, or distributing Mono.

If you file, maintain, or voluntarily participate in any claim in a lawsuit alleging direct or contributory patent infringement by Mono, or inducement of patent infringement by Mono, then your rights under this promise will automatically terminate.

This promise is not an assurance that (i) any Applicable Patents are valid or enforceable or (ii) Mono does not infringe patents or other intellectual property rights of any third party. No rights except those expressly stated in this promise are granted, waived or received by Microsoft, whether by implication, exhaustion, estoppel or otherwise. This is a personal promise directly from Microsoft to you, and you agree as a condition of benefitting from it that no Microsoft rights are received from suppliers, distributors, or otherwise in connection with this promise.

Definitions:

"Mono" means those portions of the software development technology, as originally distributed by Xamarin, Inc. or the .NET Foundation under the name "Mono," that implement .NET Framework Functionality, provided that such portions at a minimum implement all of the required parts of the mandatory provisions of Standard ECMA-335 - Common Language Infrastructure (CLI).

".NET Framework Functionality" means any functionality in Microsoft's .NET Framework as described in Microsoft's API documentation on Microsoft's MSDN website, including the functionality in Windowsbase.dll, but excluding all other functionality in the Windows Presentation Foundation component of .NET Framework.

"Applicable Patents" are those patent claims, currently owned by Microsoft and acquired in the future, that are necessarily infringed by Mono. For clarity, Applicable Patents do not include any patent claims that are infringed (x) by any underlying or enabling technology that may be used, combined, or distributed in connection with Mono (such as hardware, operating systems, or applications that run on Mono), (y) only as a consequence of modification of Mono, or (z) only by the combination of Mono with third party code.

Link: https://github.com/mono/mono/blob/master/PATENTS.TXT

Open JDK

The Eclipse Paho MQTT package is a client library for MQTT embedded devices.

Project URL: http://openjdk.java.net/

Copyright and license:

Version 1.21, 2017/12/19

OpenJDK (the "Name") is a trademark of Oracle America, Inc. ("Oracle") (the "Trademark Owner").

Trademark Owner publishes source code (the "Original Software") at several World Wide Web locations (each a "Website"). These locations include:

http://download.java.net/openjdk/jdk6 http://download.java.net/openjdk/jdk7 http://download.java.net/openjdk/jdk8 http://hg.openjdk.java.net/jdk6 http://hg.openjdk.java.net/jdk7 http://hg.openjdk.java.net/jdk8 http://hg.openjdk.java.net/jdk8u http://hg.openjdk.java.net/jdk9 http://hg.openjdk.java.net/jdk10 http://hg.openjdk.java.net/jdk http://hg.openjdk.java.net/jdk

as well as any successor locations designated by Trademark Owner in future revisions of this Notice.

Each Website provides Original Software in two parts: A Java virtual machine (the "Virtual Machine") and an API library and tools (the "Library and Tools").

Trademark Owner permits any person obtaining a copy of this software (the "Software") that is based on Original Software to use the Name in the package names and version strings of the Software subject to the following conditions:

- (1) The Software is a substantially complete implementation of the OpenJDK development kit or runtime environment source code retrieved from a single Website, and the vast majority of the Software code is identical to that upstream Original Software, except that:
 - (a) Changes required to port Original Software to new operating systems or hardware architectures are permitted, so long as that work takes place in the context of an approved Project hosted in the OpenJDK Community; and
 - (b) A Virtual Machine from one Website may be combined with the Library and Tools of another Website, so long as the vast majority of the code in each is identical to the corresponding upstream Virtual Machine or Library and Tools component.
- (2) No permission is hereby granted to use the Name in any other manner, unless such use constitutes "fair use", for example "based on the OpenJDK source code" or "DistroXYZ's packaging of the OpenJDK 6 code".

© SYSTEC electronic AG 2022	L-2562e_01	Page 122
-----------------------------	------------	----------

- (3) Trademark Owner makes no warranties of any kind respecting the Name, and all representations and warranties, including any implied warranty of merchantability, fitness for a particular purpose or non-infringement are hereby disclaimed.
- (4) Finally, this notice and the following legend are included in all copies of the Software or portions of it:

Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates.

Trademark Owner intends to revise this Notice as necessary in order to meet the needs of the OpenJDK Community. Revisions to this notice will be announced on the public mailing list announce at openjdk.java.net, to which you may subscribe by visiting http://mail.openjdk.java.net. Please send questions or comments about this Notice to the discuss list at the same location.

Link: http://openjdk.java.net/legal/openjdk-trademark-notice.html

OpenJDK Assembly Exception

The OpenJDK source code made available by Oracle America, Inc. (Oracle) at openjdk.java.net ("OpenJDK Code") is distributed under the terms of the GNU General Public License <<u>http://www.gnu.org/copyleft/gpl.html</u>> version 2 only ("GPL2"), with the following clarification and special exception.

Linking this OpenJDK Code statically or dynamically with other code is making a combined work based on this library. Thus, the terms and conditions of GPL2 cover the whole combination.

As a special exception, Oracle gives you permission to link this OpenJDK Code with certain code licensed by Oracle as indicated at http://openjdk.java.net/legal/exception-modules-2007-05-08.html ("Designated Exception Modules") to produce an executable, regardless of the license terms of the Designated Exception Modules, and to copy and distribute the resulting executable under GPL2, provided that the Designated Exception Modules continue to be governed by the licenses under which they were offered by Oracle.

As such, it allows licensees and sublicensees of Oracle's GPL2 OpenJDK Code to build an executable that includes those portions of necessary code that Oracle could not provide under GPL2 (or that Oracle has provided under GPL2 with the Classpath exception). If you modify or add to the OpenJDK code, that new GPL2 code may still be combined with Designated Exception Modules if the new code is made subject to this exception by its copyright holder.

Link: http://openjdk.java.net/legal/assembly-exception.html

OpenJDK Designated Exception Modules

8 May 2007

For purposes of those files in the OpenJDK distribution that are subject to the <u>Assembly Exception</u>, the following shall be deemed Designated Exception Modules:

- Those files in the OpenJDK distribution available at openjdk.java.net, openjdk.dev.java.net, and download.java.net to which Sun has applied the Classpath Exception,
- 2. Any of your derivative works of #1 above, to the extent you license them under the GPLv2 with the Classpath Exception as defined in the OpenJDK distribution available at openjdk.java.net, openjdk.dev.java.net, or download.java.net,
- 3. Any files in the OpenJDK distribution that are made available at openjdk.java.net, openjdk.dev.java.net, or download.java.net under a binary code license, and
- 4. Any files in the OpenJDK distribution that are made available at openjdk.java.net, openjdk.dev.java.net, or download.java.net under an open source license other than GPL, and your derivatives thereof that are in compliance with the applicable open source license.

Link: http://openjdk.java.net/legal/exception-modules-2007-05-08.html

GNU	General	Public	License,	version	2,
with the Classpath	n Exception				
The GNU Genera	I Public License (GPL	_)			

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny

© SYSTEC electronic AG 2022	L-2562e_01	Page 124
-----------------------------	------------	----------

you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it. For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights. We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software. Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations. Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all. The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION 0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you". Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does. 1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

© SYSTEC electronic AG 2022	L-2562e_01	Page 125
-----------------------------	------------	----------

You may charge a fee for the physical act of transferring a copy, and you mav at your option offer warranty protection in exchange for a fee. 2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change. b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy (Exception: if the Program itself is interactive but of this License. does not normally print such an announcement, your work based on the Program is not required to print an announcement.) These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program. In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License. 3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and

2562e_01	Page 126
	-2562e_01

2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections

1

and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component

itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so

long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.

© SYSTEC electronic AG 2022	L-2562e_01	Page 127
-----------------------------	------------	----------

You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to

refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or

other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

L-2562e_01	Page 128
	L-2562e_01

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free

Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but $\ensuremath{\mathsf{WITHOUT}}$

© SYSTEC electronic AG 2022	L-2562e_01	Page 129
-----------------------------	------------	----------

ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc.,

59

Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it. starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes

with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are subject to the following clarification and special exception to the GPL, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of

the GNU General Public License cover the whole combination.

© SYSTEC electronic AG 2022	L-2562e_01	Page 130
-----------------------------	------------	----------

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library.

If you modify this library, you may extend this exception to your version of

the library, but you are not obligated to do so. If you do not wish to do

so, delete this exception statement from your version.

Link: http://openjdk.java.net/legal/gplv2+ce.html

Node-RED

A visual tool for wiring the Internet of Things.

Project URL: https://github.com/node-red/node-red

Copyright and license:

Copyright JS Foundation and other contributors, http://js.foundation

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

© SYSTEC electronic AG 2022	L-2562e_01	Page 132
-----------------------------	------------	----------

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

- 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
- 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
- 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

© SYSTEC electronic AG 2022	L-2562e_01	Page 133
-----------------------------	------------	----------

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

- 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
- 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
- 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
- 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
- 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

	© SYSTEC electronic AG 2022	L-2562e_01	Page 134
--	-----------------------------	------------	----------

Docker CE (Community Edition)

The Docker CE package is an open source operating-system-level virtualization (containerization) application, which provides resource isolation of "containers" by using various features of the Linux kernel instead of running actual virtual machines (VMs).

Project URL: https://github.com/docker/docker-ce/

Copyright and license:

Apache License Version 2.0, January 2004 https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

© SYSTEC electronic AG 2022 L-2562e_01	
--	--

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

- 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
- 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
- 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

© SYSTEC electronic AG 2022	L-2562e_01	Page 137
-----------------------------	------------	----------

the conditions stated in this License.

- 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
- 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
- 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
- 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
- 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2013-2017 Docker, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Appendix D: Declaration of RoHS-Conformity



RoHS-Konformitätserklärung Declaration of RoHS Conformity

Wir (Name des Herstellers) / We (name of the producer)

Adresse/address

SYS TEC electronic AG

Am Windrad 2, D-08468 Heinsdorfergrund

erklären abhängig von den Angaben unserer Lieferanten und der uns zur Verfügung stehenden Unterlagen, dass die Produkte: declare depending on the information and the documents from our suppliers, that the products:

Typ / type	1.
()p/type	Kundenname
1903 CTR-500 Gesamtgerät Rev.0	
1903 CTR-500 Sinkpode Rev 0	
1000 OTR 500 Sinkhoue Rev.0	-
1903 CTR-500 Low Cost IO Rev.0	
	Typ / type 1903 CTR-500 Gesamtgerät Rev.0 1903 CTR-500 Sinknode Rev.0 1903 CTR-500 Low Cost IO Rev.0

der Richtlinie 2011/65/EU sowie der um Richtlinie 2015/863/EU erweiterten Fassung zur Beschränkung der Verwendung bestimmter gefährlicher Stoffe in Elektro- und Elektronikgeräten (RoHS III) entsprechen.

comply with the Directive 2011/65/EU and the extension by Directive 2015/863/EU on the restriction of the use of certain hazardous substances in electrical and electronic devices (RoHS III).

Name / name	Ronald Sieber
Funktion / title	Vorstand / Board of Directors
Datum / date	19.12.2019
Unterschrift signature	7.
êve tra	

SYS TEC electronic AG • Am Windrad 2 • D-08468 Heinsdorfergrund • Tel. +49 3765 386000 • www.systec-electronic.com

Index

Α

Administration	
Adding and Deleting Users	36
Auto-Boot	26
Ethernet Configuration	26
Serial Connection	24
Software Tools	24

С

Cable Types and Lengths	
CAN Interfaces	23, 62
CAN0	66
CAN1	66
CANopen	65
Network Variables CAN1	62
Change Passwords	37
Command Shell	34

D

Digital Inputs	20
Digital Outputs	21
Alternate Function	21

Ε

Error-LED	64 63
F	
File Systems	.39
G	
Galvanic Isolation	.19
J	
Java	53
М	
Mono/C# Mounting	50 18
Ν	
Node-RED	.44
0	
OpenJDK OpenPCS Deleting PLC program	.53 .61
	00

Ρ

Pin assignment	16
	~ ~ ~ ~
Configuration File	
License Key	
Web Frontend	
Power Supply	
Predefined User Accounts	
Process Image	61
Product Description	
CPU	
Dimensions	
Python	59

R

Remote Access	. 34
Run/Stop Switch	. 63
Run-LED	63

S

Serial Interfaces	62
uUSB	22
Linux Device	23
RS-485	-0 22
SorialO	ານ
	22 22
	22
Service	24
X10	22
X15	22
X16	22
Service Configuration	
Systemd Services	32
Setting System Time	37
SETP-Server (35
Shared Process Image	7∩
Software Installation and Undate	20 20
Install Debian to a SD Card	53 44
Install Deblan to a SD Card	41
Install/Update Debian to ElviviC	42
Updating the PLC Firmware	39
τ	

-		
Document:	System Manual sysWORXX CTR-500	
Document Number:	L-2562e_01, Edition January 2022	
Do you have any sugge	stions for improving this manual?	
Did you find any mistak	tes in this manual?	Page
		-
Sent from:		
Customer number:		
Customer number.		
Name:		
<u> </u>		
Company:		
Address:		
Send to:	SYS TEC electronic AG	
	Am Windrad 2 D - 08468 Heinsdorfergrund	
	GERMANY	
	Fax: +49 (0) 37 65 / 38600-4100 Email: <u>info@systec-electronic.com</u>	