sysWORXX CTR-700 and CTR-710

Technical Specifications

Version 1.0

Edition Sepemtber 2023

Document No.: L-2843e_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

Changes

Date/Version	Section	Changes	Author/Editor
09.2023	-	Initial document	Ch. Schuster

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 based on 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.

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

Table of Contents

1 Introduction 6

2 Overview 7

3 Product Description 8

4 Interface of the Device 10
4.1 Pin Assignment 10
4.2 User Interface 13
4.3 Power Supply 13
4.4 Galvanic Isolation 14
4.5 Cable types and maximum cable lengths 14
4.6 Analog Inputs 15
4.7 Serial Interfaces 15
4.8 Shared Process Image 16

5 Firmware Function Scope 18

6 Library sysWORXX IO 23
6.1 IoResult Description 23
6.2 Public Interface Description 23
6.3 Device Interface and Signal Channel Mapping 24

7 Technical Specification 26

Index 28

List of Tables

Table 1: Overview of relevant manuals 7
Table 2: Pin assignment 11
Table 3: Description Switches 13
Table 4: Description LEDs 13
Table 5: Recommended cable lengths, types and wire cross section 14
Table 6: Serial interface to Linux device node path 15
Table 7: Assignment of in- and outputs to the process image 16
Table 8: Firmware functions and function blocks 18
Table 9: SYSWORXX IO Return Status Codes 23
Table 10: SYSWORXX IO Public Interface Functions 23
Table 11: Device Interface and Signal Channel Mapping 24

List of Figures

Figure 1: Top view of the sysWORXX CTR-700 8
Figure 2: Top view of the sysWORXX CTR-710 8
Figure 3: Interface and connector assignment 10
Figure 4: Galvanic isolation 14

1 Introduction

This document includes all device specific information for the sysWORXX CTR-700 and CTR-710. The devices are identical, except the build-in interface connections.

To get more general information on how to operate the device, please refer to the System Manual of our sysWORXX device series L-2782 (see Table 1).

2 Overview

Table 1 lists up all relevant manuals for the device.

Table 1: Overview of relevant manuals

Information about…	**In which manual?**
Specific technical information about the device	In this manual
Basic information about the device (configuration, administration, process image, connection assignment, firmware update, reference designs et cetera)	System Manual sysWORXX series – L-2782

3 Product Description

The sysWORXX CTR-700 and CTR-710 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 in- and 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 and interfaces, one can write their own applications for their specific needs and purposes.

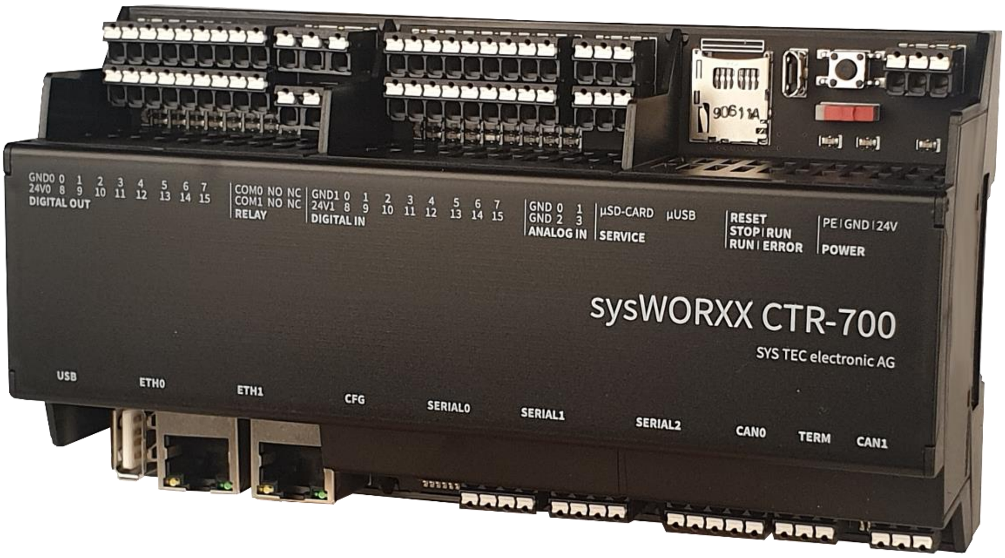

Figure 1: Top view of the sysWORXX CTR-700

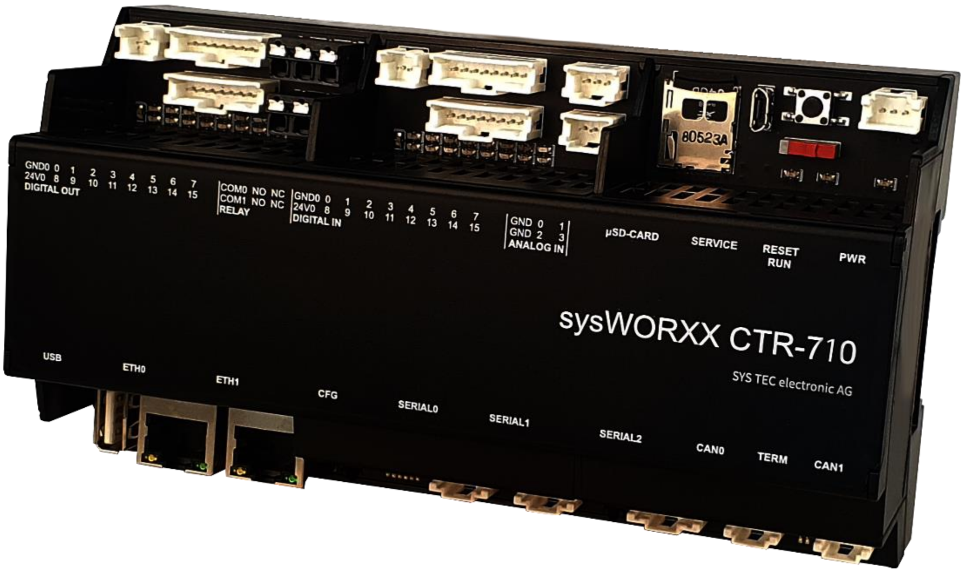

Figure 2: Top view of the sysWORXX CTR-710

The most significant features of the device are listed below:

- 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
- 2x 10/100/1000 Mbps Ethernet LAN interface
- 2x CAN 2.0B interface, usable as CANopen Manager (CiA 302-conform)
- 3x asynchronous serial ports (UART), usable as RS-232 or RS-485
- 16 digital inputs 24VDC, galvanic isolated
    - Alternate function: 1 high-speed counter input, galvanic isolated
    - Alternate function: 1 A/B-Encoder
- 16 digital outputs 24VDC/500mA, galvanic isolated, short-circuit-proof
    - Alternate function: 2 PWM outputs 24VDC/500mA 1KHz
- 2 Relay outputs (2x change-over relay)
- 4 analog inputs 0-10VDC or 0-20mA with 12-Bit resolution
- 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
- Programmable according to IEC 61131-3, C/C++, C#, Java, and Python
- Function block libraries for communication (CANopen, Ethernet and UART)
- Function block libraries for hardware components (RTC, Counter, PWM)
- Linux-based (other user programs are executable in parallel)
- Easy, HTML-based OpenPCS configuration via web browser
- Remote login via SSH
- Dimensions: 162 x 91 x 60mm
- Temperature 0° … 55°C
- Suitable for DIN top hat rail mounting

The default hardware comes with a Linux operating system. This base system can be used to program in different programming languages and 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:

Order no.: **16061003**: sysWORXX CTR-700 with basic Debian/GNU Linux installation, including microUSB cable for serial terminal via SERVICE plug

Order no.: **19371003**: sysWORXX CTR-710 with basic Debian/GNU Linux installation, including microUSB cable for serial terminal via SERVICE plug and Wire-to-Board connectors

Order no.: **240011**: Runtime license OpenPCS RT sysWORXX devices

To develop software with our device, one can find sources on our GitHub (https://github.com/systec-electronic) and our website (https://www.systec-electronic.com/).

4 Interface of the Device

4.1 Pin Assignment

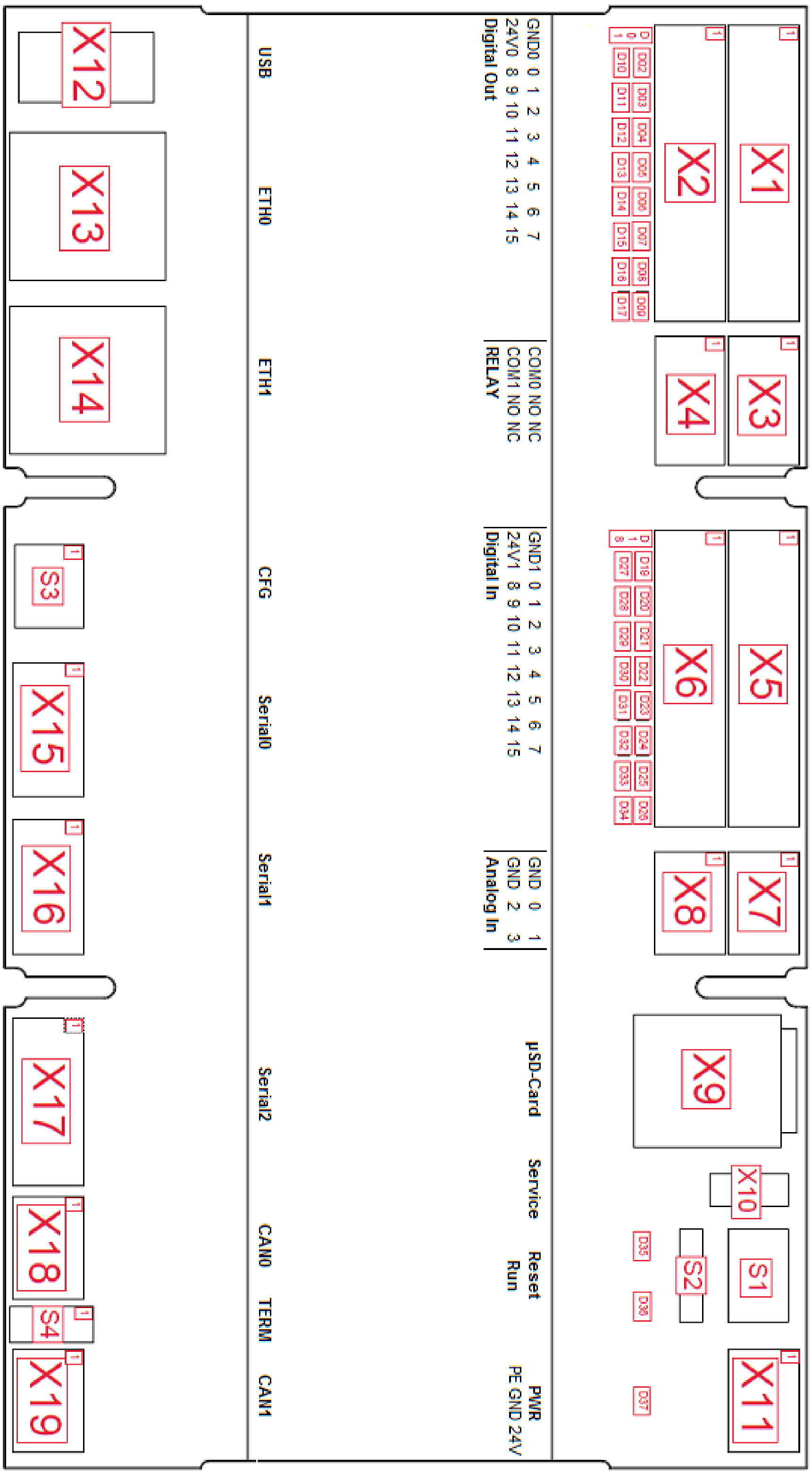

Figure 3: Interface and connector assignment

Figure 3 shows the positioning of connectors on the device as an overview. Table 2 lists all connectors in detail.

Table 2: Pin assignment

Terminal	Pin	Signal name	Remark
Digital Outputs X1	1	GND0	24V
	2 … 9	DO0 … DO7	
Digital Outputs X2	1	24V0	
	2 … 9	DO8 … DO15	
Relay 0 X3	1	COM0	230V
	2	NO	
	3	NC	
Relay 1 X4	1	COM1	230V
	2	NO	
	3	NC	
Digital Inputs X5	1	GND1	24V
	2 … 9	DI0 … DI7	
Digital Inputs X6	1	24V1	
	2 … 9	DI8 … DI15	
Analog Inputs X7	1	GND	0 … 10V/0 … 20mA
	2, 3	AIN 0, AIN 1	
Analog Inputs X8	1	GND	
	2, 3	AIN 2, AIN 3	
µSD-Card-Holder X9	-	-	-
µUSB (console) X10	-	-	-
Power X11	1	PE	24V
	2	GND	
	3	24VDC	
USB-Host X12	-	-	-
Ethernet 0 X13	-	-	-
Ethernet 1 X14	-	-	-

Terminal	Pin	Signal name	Remark
Serial Interface 0 X15	1	RX	RS-232
	2	-	
	3	TX	
	4	GND	
	1	A/D0	RS-485/Modbus RTU
	2	B/D1	
	3	-	
	4	GND	
Serial Interface 1 X16	1	RX	RS-232
	2	-	
	3	TX	
	4	GND	
	1	A/D0	RS-485/Modbus RTU
	2	B/D1	
	3	-	
	4	GND	
Serial Interface 2 X17	1	RX	RS-232
	2	CTS	
	3	TX	
	4	RTS	
	5	GND	
	1	A/D0	RS-485/Modbus RTU
	2	B/D1	
	3	-	
	4	-	
	5	GND	
CAN 0 X18	1	HIGH	-
	2	LOW	
	3	GND_CAN0	
CAN 1 X19	1	HIGH	-
	2	LOW	
	3	GND_CAN1	

NOTICE

Damage through additional/improper system expansions

The installation of additional expansions (sensors, actuators, …) may damage the device or machine. Device and system expansions may also violate safety rules and regulations regarding radio interference suppression. If you install or exchange system expansions and damage your device, you void your warranty.

Install only expansions or devices which are specified to be used with this device. When in doubt contact your local technical support team or the SYS TEC electronic support at support@systec-electronic.com

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.

4.2 User Interface

Table 3: Description Switches

Switch	Port	OFF	ON
S1	-	-	RESET
S2	-	Position = left (see Figure 3) PLC stop (outputs deactivated)	Position = right (see Figure 3) PLC run (program re-starts)
S3[1]	1	**SERIAL0:** RS-485: Bus termination off RS-232: Must be set!	**SERIAL0:** RS-485: Bus termination on RS-232: Do not set!
	2	**SERIAL1:** RS-485: Bus termination off RS-232: Must be set!	**SERIAL1:** RS-485: Bus termination on RS-232: Do not set!
	3	**SERIAL2**: RS-485: Bus termination off RS-232: Must be set!	**SERIAL2:** RS-485: Bus termination on RS-232: Do not set!
	4	Configuration: **ON**	Configuration: **OFF**
	5	Normal Booting	Boot in U-Boot Bootloader CLI
	6	Boot mode: SD-Card	Boot mode: eMMC
S4	1	CAN0: Termination off	CAN0: Termination on
	2	CAN1: Termination off	CAN1: Termination on

Table 4: Description LEDs

LED	Color	Feature
D01	Green	Status of the power supply for the DO’s
D02 … D17	Yellow	Signal status of the DO’s (on = high; D02 = DO0, D03 = DO1, …)
D18	Green	Status of the power supply for the DI’s
D19 … D34	Yellow	Signal status of the DI’s (on = high; D19 = DI0, D20 = DI1, …)
D35	Green	Status of the PLC: **RUN** f=1 Hz; **STOPP** f=0.5 Hz
D36	Red	ERROR-LED, signals an occurring (PLC or IO-driver) error
D37	Green	Status of the power supply for the device

4.3 Power Supply

The devices feature power supply inputs (24VDC ±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, there is a short time window to act accordingly, for example to safe your data. During this time frame, the device will work normally then shuts down. To access the power fail signal, one must use the driver library in a separate program (see L-2782, Section 6). The exact values for power fail and the delay time are documented in section 7.

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

4.4 Galvanic Isolation

Figure 4 shows the galvanic isolation of the different interfaces and system components.

Digital Out 0..15	24V0 GND0	Relais 0	CAN1	ETH0	Analog In USB Serial 0..2 System Power
Digital In 0..15	24V0 GND0	Relais 1	CAN0	ETH1	

Figure 4: Galvanic isolation

- Analog In, USB and serial interfaces have **no isolation**.
- Digital out/in and CAN have **functional isolation** of 50 VDC
- Interfaces ETH0 and ETH1 have a **basic isolation** of 230 VAC
- Relays 0 and 1 have **reinforced isolation** of 230 VAC

4.5 Cable types and maximum cable lengths

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

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

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			
Analog inputs			
Analog outputs			
Relay Output			
RTD			
Thermocouple			
CAN	<30m	Shielded twisted pair	
	RS232: <15m	Shielded twisted pair	

SERIAL	Modbus RTU: <30m		
Ethernet	<30m	S/FTP Cat 6	-
USB, Service	<3m	USB standard cable	-
MESHNET	-	-	
WLAN	-	-	
POWER	<30m	Any cable suitable to the specific usage.	1mm^2 - 1,5mm^2 or AWG17 - 16

4.6 Analog Inputs

To get the real value for the voltage or current measurement, the LSB must be multiplied by the measured digits. The specific values and calculation are as follows:

Example for voltage measurement:

U = 1 LSB * DIGIT
U = 355,26µV * 28145 = 9,999V

Example for current measurement:

I = 1 LSB * DIGIT
I = 819.87nA * 16384 = 13,432mA

4.7 Serial Interfaces

The following table shows the list of serial interfaces and the corresponding Linux-Devices:

Table 6: Serial interface to Linux device node path

Interface	**Linux-Device (*INTERFACE*)**
SERIAL0	/dev/ttymxc6
SERIAL1	/dev/ttymxc5
SERIAL2	/dev/ttymxc1
SERVICE [2]	/dev/ttymxc0

[2] 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.

4.8 Shared Process Image

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 7.

Table 7: Assignment of in- and outputs to the process image

I/O of the device	Address and Data type in the Process Image	
DI0 ... DI7	**%IB0.0** %IX0.0 … %IX0.7	as Byte with DI0 … DI7 as single Bit for each input
DI8 ... DI15	**%IB1.0** %IX1.0 … %IX1.7	as Byte with DI8 … DI15 as single Bit for each input
AI0	**%IW8.0**	15Bit + sign (0 … +32767)
AI1	**%IW10.0**	15Bit + sign (0 … +32767)
AI2	**%IW12.0**	15Bit + sign (0 … +32767)
AI3	**%IW14.0**	15Bit + sign (0 … +32767)
C0	**%ID40.0** counter input: DI14 (%IX1.6), direction: DI15 (%IX1.7)	31Bit + sign (-2^{31} - 2^{31} -1)
CPU Temperature Sensor	**%ID72.0**	31Bit + sign as 1/10000 °C
System Temperature Sensor	**%ID76.0**	31Bit + sign as 1/10000 °C
DO0 ... DO7	**%QB0.0** %QX0.0 … %QX0.7	as Byte with DO0 … DO7 as single Bit for each output
DO8 … DO15	**%QB1.0** %QX1.0 … %QX1.7	as Byte with DO8 … DO15 as single Bit for each output
REL0 and REL1 (corresponds to DO16 … DO17)	**%QB2.0** %QX2.0 … %QX2.1	as Byte with REL0 and REL1 as single Bit for each Relay
DO Mask	**%QD1984.0** a bitmask of digital outputs, which will **not** be written from process image to hardware registers. (LSB relates to DO0, MSB relates to DO15)	Mask of digital outputs
AI0 Configuration	**%QB1928.0** 0: keep configuration, 1: set to voltage measurement, 2: set to current measurement	Configuration of AI0
AI1 Configuration	**%QB1929.0** see AI0 Configuration	Configuration of AI1
AI2 Configuration	**%QB1930.0** see AI0 Configuration	Configuration of AI2
AI3 Configuration	**%QB1931.0** see AI0 Configuration	Configuration of AI3

Advice: The device works with Little-Endian format (“Intel-Notation). Consequently, and on the 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  AT %IB0.0 : BYTE;
bInByte1  AT %IB1.0 : BYTE;
wInWord0  AT %IW0.0 : 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.

5 Firmware Function Scope

Table 8 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 8: Firmware functions and function blocks

Name	**Type**	**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	
CTU	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 OpenPCS specific task controlling			
GETVARDATA	FB	Online Help	
GETVARFLATADDRESS	FB	Online Help	
GETTASKINFO	FB	Online Help	
Functions and Function Blocks for handling of non-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 handling of time			
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 counter inputs and pulse outputs			
CNT_FUD	FB	L-1054	CHANNEL:={0,1,2}
PTO_PWM	FB	L-1054	CHANNEL:={0,1}
PTO_TAB	FB	L-1054	CHANNEL:={0,1}
Function Block for PID regulator			
PID1	FB	L-1054	

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 interfaces / 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	

Functions and Function Blocks for Modbus communication			
MODBUS_OPEN_INSTANCE	FB	L-1829	
MODBUS_CLOSE_INSTANCE	FB	L-1829	
MODBUS_REGISTER_VAR_LIST	FB	L-1829	
MODBUS_READ_REGS	FB	L-1829	
MODBUS_WRITE_SINGLE_REG	FB	L-1829	
MODBUS_WRITE_MULTI_REGS	FB	L-1829	
MODBUS_READ_WRITE_REGS	FB	L-1829	
MODBUS_READ_INPUT_REGS	FB	L-1829	
MODBUS_READ_DISCR_INPUTS	FB	L-1829	
MODBUS_READ_COILS	FB	L-1829	
MODBUS_WRITE_SINGLE_COIL	FB	L-1829	
MODBUS_WRITE_MULTI_COILS	FB	L-1829	
MODBUS_RAW_PDU_REQUEST	FB	L-1829	
Functions and Function Blocks for MQTT communication			
MQTT_GET_CAPABILITIES	FB	Demo	
MQTT_CONNECT	FB	Demo	
MQTT_DISCONNECT	FB	Demo	
MQTT_GET_CONNECT_STATE	FUN	Demo	
MQTT_SUBSCRIBE	FB	Demo	
MQTT_UNSUBSCRIBE	FB	Demo	
MQTT_GET_ARRIVED_MESSAGE	FB	Demo	
MQTT_PUBLISH	FB	Demo	

[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 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.

[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 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.

6 Library sysWORXX IO

This section describes the usage of the sysWORXX IO library that enables access to the different inputs and outputs of the device as well as additional device information. The header file with the public interface is located at “**/usr/include/sysworxx_io.h**”. The driver is managed via a systemd service “*iodaemon.service*” that is enabled by default and located at “**/lib/systemd/system**”.

To enable access to the interfaces of the device include the header in a *C/C*++ application and use the public interface to program your functionality. Afterwards compile the application with the GCC-compiler and link against the corresponding library located at “**/usr/lib/libsysworxx_io.so**”. For the necessary compiler flags and usability of the “GNU Compiler Collection” refer to https://gcc.gnu.org/onlinedocs/gcc-7.3.0/gcc/.

6.1 IoResult Description

Each of the public functions listed in *Table 10* returns a status code that is listed in *Table 9* to indicate the success of the function’s behavior.

Table 9: SYSWORXX IO Return Status Codes

Error	**Return Status Code**
IoResult_Success	0
IoResult_Error	255
IoResult_NotImplemented	254
IoResult_InvalidParameter	253
IoResult_InvalidChannel	252
IoResult_InvalidMode	251
IoResult_InvalidTimebase	250
IoResult_InvalidDelta	249
IoResult_PtoParamTabFull	248
IoResult_DevAccessFailed	247
IoResult_AddressOutOfRange	243
IoResult_WatchdogTimeout	242

6.2 Public Interface Description

The functions listed in *Table 10* are used to access the peripheral as well as status flags of the device. You may us it to read or write digital as well as analog values and setup configurations for e.g., RDT inputs. For details about the parameters of the functions please refer the header file “**sysworxx_io.h**”.

Table 10: SYSWORXX IO Public Interface Functions

Function Name	**Description**
IoInit	Initializes the I/O driver
IoShutdown	De-initialization of the I/O driver
IoGetVersion	Get the version of the I/O driver
IoGetTickCount	Get the tickcount of the system in milliseconds
IoEnableWatchdog	Enable the systems watchdog
IoServiceWatchdog	Service the system watchdog
IoGetHardwareInfo	Get information about device revision and available I/O channels
IoSetRunLed	Set/reset the RUN LED
IoSetErrLed	Set/reset the ERROR LED
IoGetRunSwitch	Get status of the RUN switch
IoGetConfigEnabled	Get status of the config switch

Function Name	Description
IoSetOutput	Set the value of a digital output e.g. REL[0,1]
IoGetInput	Get the value of a digital input e.g., D0…D9
IoRegisterInputCallback	Register a callback to signal changes on a digital input
IoUnregisterInputCallback	Un-register / disable interrupt handling for a digital input
IoAdcGetValue	Get the value of an ADC channel e.g., AI0 … AI7
IoAdcSetMode	Setup an ADC channel for voltage or current measurement
IoTmpGetValue	Get the value of a temperature sensor
IoCntEnable	Enable/disable a counter channel
IoCntSetup	Setup the counters mode e.g., mode, trigger, or direction
IoCntSetPreload	Set the initial value of the counter
IoCntGetValue	Get the value of a counter channel

6.3 Device Interface and Signal Channel Mapping

To access status signals or device inputs with the functions listed in *Table 10*, the target of a function is addressed with channel numbers. The following table lists the channels to the corresponding signals or device inputs/outputs. The actual channel names and values can vary depending on the device.

Table 11: Device Interface and Signal Channel Mapping

Name	Channel	Remarks
Digital Inputs		
DI0	0	Access through “IoGetInput” Event notification though “IoRegisterInputCallback” and “IoUnregisterInputCallback”
DI1	1	
DI2	2	
DI3	3	
DI4	4	
DI5	5	
DI6	6	
DI7	7	
DI8	8	
DI9	9	
DI10	10	
DI11	11	
DI12	12	
DI13	13	
DI14	14	
DI15	15	
Digital Outputs		
DI0	0	Access through “IoSetOutput”
DI1	1	
DI2	2	
DI3	3	
DI4	4	
DI5	5	
DI6	6	
DI7	7	
DI8	8	
DI9	9	
DI10	10	
DI11	11	
DI12	12	

Name	Channel	Remarks
DI13	13	
DI14	14	
DI15	15	
Relay Outputs		
REL0	0	Access through” IoSetOutput”
REL1	1	
Analog Inputs		
AI0	0	Access though “IoAdcGetValue” Setup though “IoAdcSetMode”
AI1	1	
AI2	2	
AI3	3	
Temperature Sensors		Default
CPU Temperature Sensor	0	Access through “IoTmpGetValue”
System Temperature Sensor	1	
Counters		
Counter 0	0	Access through “IoCntGetValue” Enable/Disable through “IoCntEnable” Configure through “IoCntSetPreload” and “IoCntSetup”
Counter 1	1	
LEDs		
PLC Run Led	-	Access through dedicated function “IoSetRunLed”
PLC Error Led	-	Access through dedicated function “IoSetErrorLed”
Status/Control Signals		
PowerFail24VDC (/PF)	32	Access through “IoGetInput” Event notification though “IoRegisterInputCallback” and “IoUnregisterInputCallback”
DiEr (DI_/ERR)	33	
UsbOc (USB_/OC)	34	
DoPf/DO_PF)	35	
DoDiag	36	

7 Technical Specification

Environmental Parameters		Typical	Minimum Maximum
Power Supply	V_{CPU}	24VDC	19.2…30VDC
	V_{IO}	24VDC	19.2…30VDC
	power fail level	18,2V	
	power fail delay time	10ms	
Current Consumption (Inactive IOs)	I_{CPU}	100mA	
	I_{IO}	30mA	
Temperature Range	Storage temperature		-20..+70°C
	Operating temperature		0…55°C
MTTF	According to SN29500 at 40°C		> 373134 h
Protection class	Housing	IP20	
Weight	without any cable and packing	295g	
Dimensions	Width Height Depth		162mm 61mm 91mm
Connector type	Spring type connector		
Lowest cycle time for PLC		5 ms	

I/O-configuration (digital)		Typical	Maximum
Digital Outputs DO0 ... 15			
24VDC-Output (High Side Switch)	U_{OH} at I_{OH} = 500 mA	V_{IO}-0.12V < U_{OH} <V_{IO}	
	U_{OL} at I_{OL} = 0 mA		0.5V
	Current limitation I_{OH_max}		700mA
	Max. current		16x0,5A
	Impedance		0.11Ohm
	$I_{OL(off)}$		10µA
	t_{off} at I_{OH} = 500 mA	22 µs	27µs
	t_{on} at I_{OH} = 500 mA	27 µs	45µs
	Frequency	ca. 200 Hz	1KHz[3]
PWM Output (DO14 + DO15)			
24VDC-Output (High Side Switch)	T_{jitter}		25µs
	T_{on_min} and min. pulse width		800ns
	Frequency		1kHz
Digital Outputs RLY0/RLY1			
Relay output (N.O.)	Switching Voltage		220VDC 250VAC
	Switching Current		110VDC / 0.3A 30VDC / 2.0A 120VAC / 0.5A 240VAC / 0.25
	Contact rating		60W/62.5VA
	Durability (mechanical.)	100x10^{6}	
	Durability (electrical.) @12V/10mA @60V/500mA @30V/1000mA @30V/2000mA	 5x10^{7} 5x10^{5} 1x10^{6} 2x10^{5}	
	t_{on}	4ms	
	t_{off}	4ms	
	Isolation	1000Vrms	

[3] Frequency limit of hardware interface circuit. The frequency may vary due to software limitations.

Digital Inputs DI0 … 15			
24VDC- Inputs, plus switching	U_{IH}	13V	30V
	U_{IL}	-3V	12.3V
	I_{IH} (V_{IN}=6.7V)	1.3mA	
	I_{IH} (V_{IN}=30V)		3.5mA
	Input type according to IEC61131-2	Type 1	
	T_{DLY}		100ns
Counter Input and Step Direction (DI14 + DI15)			
24VDC- Inputs, plus switching	Min. pulse width		25µs
	Frequency		10kHz
A/B-Encoder (DI14 + DI15)			
24VDC- Inputs, plus switching	Frequency		10kHz
	Phase Margin (A/B-Encoder)		±45°

I/O-configuration (analog)		Typical	Maximum
Analog Inputs AI0 ... 3			
0 ... 10V	Measurement range U_I	0…11.64V	
	Measurement error	0.23%	0.5%[4]
	Destructive voltage U_{I_max}	-	30V
	Input resistance R_I	97kΩ ±0.1%	
	Physical Resolution	-	12Bit
	LSB	355.23 µV	
	Cut off frequency	70Hz	
0 ... 20mA	Measurement range I_I	0…26.865mA[5]	
	Measurement error	0.23%	0.5%[4]
	Input resistance R_I	67Ω ±0.1%	
	Physical Resolution	-	12Bit
	LSB	819.87 nA	
	Cut off frequency	160Hz	

Communication Interfaces		Minimum	Maximum
CAN-Bus			
CAN1, CAN2	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
SERIAL2	Baudrate	1200Baud	115200Baud
SERVICE	Baudrate	1200Baud	115200Baud
Ethernet			
ETH0	Bandwith	10Mbit/s	1000Mbit/s
ETH1	Bandwith	10Mbit/s	1000Mbit/s
Backplane Bus			
SERIAL1	Baudrate	1200Baud	115200Baud
SPI	Frequency		5MHz

Standards and approvals		
See the appendix or product folder on the website for the currently valid declaration of conformity		

[4] Value is defined over temperature range.

[5] Input is protected against overcurrent; max. voltage should not exceed 30V

Index

C

Cable Types and Lengths..............................*14*

G

Galvanic Isolation 14

P

Pin assignment 11
Power Supply..13, 26
Process Image..16
Product Description 8
  CPU 9
  Dimensions.. 9

S

Serial Interfaces
  Linux Device.. 15

U

User Interface
  LEDs.. 13
  Switches 13

Document:	**Technical Specifications sysWORXX CTR-700**
Document Number:	**L-2843e_01, Edition Sepemtber 2023**

Do you have any suggestions for improving this manual?

Did you find any mistakes in this manual? **Page**

Sent from:

Customer number:

Name:

Company:

Address:

Send to:

SYS TEC electronic AG
Am Windrad 2
D - 08468 Heinsdorfergrund
GERMANY
Fax: +49 (0) 37 65 / 38600-4100
Email: info@systec-electronic.com