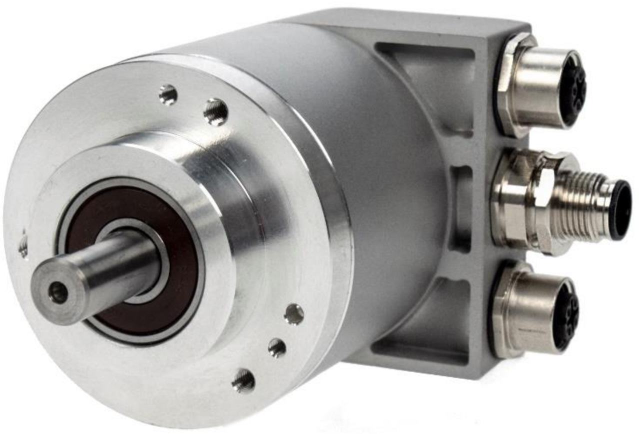
HENGSTLER

Technical Manual



ACURO® AC58/AC61 Profinet

2 565 737

HENGSTLER GmbH
Uhlandstr. 49
78554 Aldingen / Germany
Tel. +49 (0) 7424-89 0
Fax +49 (0) 7424-89 500
E-Mail: info@hengstler.com
www.hengstler.com

HENGSTLER GmbH
Uhlandstr. 49
78554 Aldingen / Germany
Tel. +49 (0) 7424-89 0
Fax +49 (0) 7424-89 500
E-Mail: info@hengstler.com
www.hengstler.com

General Manager: Jochen Feiler, Commercial Register Tuttlingen HRB 604 Sp, ID-Nr. DE 811194298

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 About this Document

This manual describes the ACURO® AC58/AC61 Profinet family of absolute shaft encoders, including specifications, dimensions, software, commissioning and parameterization.

1.2 Terms, Abbreviations and Definitions

For Abbreviations according Data coding, please see Appendix A

Term	Description
AC	PROFIdrive Application Class
AP	Application Process
API	Application Process Identifier
APO	Application Process Object
ASE	Application Service Element
CO	Communication Object
CR	Communication Relationship
CRC	Cyclic Redundancy Check
C-LS	Controller's Sign-Of-Life
CM	Context Management
DAP	Device Access Point
DO	Drive Object
DO-LS	Drive Object Sign-Of-Life
DP	Decentralized (distributed) periphery
DSC	Dynamic Servo Control
DU	Drive Unit
DX	Dada Exchange
FAL	Fieldbus Application Layer
FDL	Fieldbus Fata Link (Layer 2)
GC	Global Control Telegram
GSD	Device Data File (device description, input for a bus configuring tool)
GSDML	GSD Markup Language
HWR	Hardware Requirements Specification
ID	Identifier

IOAR	IO Application Relation
IOCS	IO Consumer Status
IOPS	IO Consumer Status
IOXS	IOCS and IOPS
IO Data	IO Data transmitted cyclically
IRT	Isochronous Real-Time Ethernet
IsoM	Isochroous Mode
I/O	Input/Output
I&M or IM	Identification and Maintenance
Ky'	Position closed loop control gain factor
Kpc	Position closed loop gain factor
LS	Sign-Of-Life
LSB	Least Significant Bit
MAP	Module Access Point
MS0	PROFIBUS MS0 AR (cyclically data exchange between master (class 1) and slave
MS0	PROFIBUS MS0 AR (cyclically data exchange between master (class 2) and slave
MSB	Most Significant Bit
M CR	Multicast CR
NC	Numerical Control system with a numeric control command set
NW	Network
Pxxx	Parameter (Identified by number xxx)
PAP	Parameter Access Point
PBE	Parameter description
P-Device	Peripheral Device (PROFIdrive Base Model)
PDU	Protocol Data Unit
PDS	Power Drive System
PI	PROFIBUS International
PLC	Programmable Logic Controller without a Motion Control command set
PLL	Phase Locked Loop (phase control loop)
PNO	PROFIBUS User Organization

PNU	Parameter Number
PPO	Parameter IO Data Object
PROFIBUS	Process Field Bus
PROFINET	Process Field Bus, Ethernet based
Prm	Parameter
PWE	Parameter Value
RES	Reserve (spare time in DP cycle)
RFG	Ramp Function Generator
RT	Real-Time Protocol
SDK	Software Development Kit
SDR	Software Design Requirements
SN	Sign
SRS	Safety Requirements Specification
STW	Control Word
SWR	Software Requirements Specification

All variables, parameters, and data used in this manual employ the LSB/MSB ("Intel") data format. All IP addresses in this document use host byte order.

1.3 Referenced Documents

This manual refers to the following documents:

2 565 725 Hengstler Installation Manual

Hengstler Technical Datasheet

CE Cert

PNO Certification

Technical Specification for PROFINET

Encoder Profile - Technical Specification for PROFIBUS and PROFINET Version 4.2 - March 2017

Encoder Profile - Technical Specification for PROFIBUS and PROFINET Version 4.1 – Dec. 2008

PROFIdrive Profile - Technical Specification for PROFIBUS and PROFINET Version 4.2 - Oct 2015

And referenced documents

2 Safety, Installation and Operation Hints

Care must be taken when installing and using this product. Please refer to Hengstler Installation Instructions Article No. 2 565 725 for detailed information on safety and installation. While much of this information is repeated below, the full, multilingual document should be used to ensure correct and safe installation.

2.1 Authorized personnel

This encoder should only be installed or uninstalled by a qualified technician, as the unit contains sensitive electronic circuitry.

2.2 Risk of injury due to rotating parts

Hair, jewelry or articles of clothing may become caught in rotating shafts or other parts! Prior to commencing any work, disconnect all power supplies and ensure that the working environment is Safe!

2.3 Risk of injury due to safety-critical applications

When the AC58/AC61 PROFINET encoder is used in safety-critical applications which could threat life or physical condition, it is required that position related data supplied by the encoder is checked on plausibility before being used within applications.

2.4 Risk of damage due to static electricity

The CMOS modules contained in this encoder are sensitive to high voltages, such as those that can arise due to friction in clothing or shoes.
Do not touch connector contacts or electronic components!

2.5 Risk of damage due to mechanical overload

Rigid mounting will cause constraining forces which will permanently overload and damage the bearings. *Never restrict the freedom of movement of the encoder! Use only the enclosed sheet metal springs or a suitable coupling when mounting the unit!*

2.6 Risk of damage due to mechanical shock

Violent shocks, e.g. hammer blows, can lead to damage of the optical sensing system and the ball bearings. *Never use force! Assembly is simple provided that correct procedures are followed.*

2.7 Risk of damage due to overloading

The unit may only be operated within the electrical, mechanical and other limits specified in the technical data.

2.8 Over-voltage

Over-voltage at the connecting terminals *must be limited to over voltage-class-II values (SELV)*.

2.9 Dragline mounting

Choose connecting cable carefully according the required conditions. The connectors are not suitable for moving cable applications. For moving cable applications, fix the cable close to the encoder. Standard Cables are *not rated for dragline mounting*, only for fixed mounting of the encoder.

2.10 Component

This encoder is a component intended for mounting to other equipment (motor, machine, etc.). It is not intended for direct sale to the end customer.

2.11 CE Mark

Manufacturers integrating this encoder into their products are responsible for compliance with CE guidelines, and for proper use of the CE mark.

3 General Information

3.1 Introduction

We all know that no two industrial applications are alike. That's why the Hengstler AC58/AC61 absolute encoder is available with an incredible array of different options and features, including 22-bit single-turn resolution. But what if our standard variants don't meet your needs? Then Hengstler is able to offer custom versions to fulfill your requirements! Extended temperature range, greatly enhanced shock and vibration ratings, and custom shaft sizes and shapes are just a few of the features we've provided our customers recently. This flexibility makes the Hengstler AC58/AC61 one of the most versatile encoders on the market, in addition to being one of the most robust.

Now the AC58/AC61 product line has been expanded by the addition of the popular PROFINET interface. Use of systems employing this open, high performance Ethernet-based system continues to grow rapidly. By offering the AC58/AC61 with PROFINET, users can now integrate one of the best absolute encoders on the market with virtually any system using PROFINET. This simplifies the design process and ensures reliable communications.

3.2 Fields of Application

The field of application for this encoder is industrial processes and controls. Examples of applications include packaging machines, injection molding machines, wood processing machine, assembly and handling technology, conveyor technology, printing and paper machines.

4 Installation and Operating

4.1 Installation Overview

Initial installation of the Hengstler ACURO AC58/AC61 PROFINET absolute encoder should be accomplished by following Hengstler Installation Instructions, Article No. 2 565 725, available for download from the Hengstler GmbH web site at www.hengstler.com. This document provides valuable information regarding safety, encoder mounting, wiring, etc. Please be sure to keep operation of the encoder within the electrical and mechanical limits specified by Hengstler GmbH, as shown in this and other Hengstler documentation.

4.2 Initial Setup

Your Hengstler ACURO AC58/AC61 PROFINET absolute encoder comes from the factory ready to install with the default PROFINET parameters and settings (6.5.1.2). Should you wish to change these parameters and/or settings, please refer to the following portions of this manual. Please note, however, that certain reset commands will reset any special parameters and/or settings that were changed at the factory back to their default values. Care must be taken to avoid accidentally changing desired settings while programming.

4.3 Bus/Power Connections

The Hengstler ACURO AC58 PROFINET absolute encoder supports the 100BASE-TX variant of the IEEE 802.3 standard. 100BASE-TX uses shielded twisted-pair copper cables with two pairs of wires. Cables of categories CAT 5, 6 or 7 can be used. M12 connectors are used, in keeping with the industrial applications in which the encoder will be used, and due to the connectors' excellent environmental protection provided by this connector type.
The maximum distance between two nodes is limited to 100 meters.
Note, however, that IP67 and higher protection ratings are contingent upon the use of an appropriately rated mating connector/cable assembly which has been properly installed by the user.
Connections are made with three (3) M12 connectors. The connectors have the following pinouts.

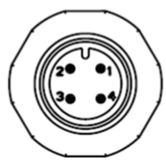
	Connector		
Pin	**Bus Port 2**	**Power (Supply Voltage)**	**Bus Port 1**
1	TxD+	UB in	TxD+
2	RxD+	N.C.	RxD+
3	TxD-	0 V in	TxD-
4	RxD-	N.C.	RxD-
Shield	Shield [1]	Shield [1]	Shield [1]
	 M12 connector, D-coded	 M12 connector, A-coded	 M12 connector, D-coded
1 shield connected to encoder housing			

Table 1: Connector Pinouts

4.4 LED Interpretation

The table “Communication and Power LEDs” explain the states of the Communication LEDs (MOD, NET, P1 and P2) and Power LED of the ACURO PROFINET. The location of the LEDs is as shown in Figure 1.

The LED are not visual with Explosion proof (AX65/70/71/73) or diverse customer-specific enclosures

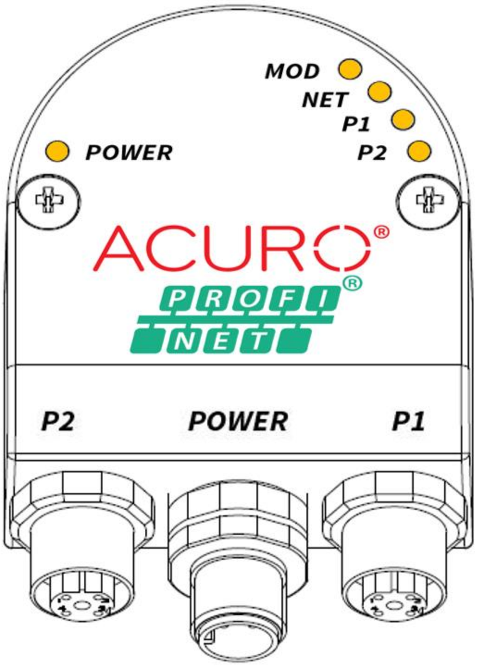


Figure 1: ACURO AC58/AC61 PROFINET Connector/LED Locations

The LED Indicator States are defined as follows. (Green indicates LED lit, white indicates LED off.)

- **Off:** LED is not illuminated at all.
- **On:** LED is illuminated continually.
- **Blinking:** LED flashes on and off continually but slowly with even spacing.

Name	Indicator State	Color	Meaning
MOD (Module Status)	Off		Module inactive
	Blinking	Red	Module-Configuration missing
	On	Red	Module-Error
	Blinking	Orange	Firmware-Update running
	On	Green	Module active
NET (Network Status)	Off		Network inactive
	Blinking	Red	Network Connection missing
	On	Red	Network active, PROFINET inactive
	Blinking	Orange	Firmware-Update running
	On	Green	PROFINET active

Name	Indicator State	Color	Meaning
P1 / P2 (Port 1 / Port 2)	Off		No Network connection to other Ethernet-Component
	Blinking	Green	Ethernet-Component connected on this port, Communication running
	On	Green	Ethernet-Component connected on this port
Power	Off		Power supply insufficient
	On	Green	Power supply sufficient

Table 2: Communication and Power LEDs

5 Specification

5.1 Base Model

The Encoder profile is based on the PROFIdrive profile [1] and uses therefore the PROFIdrive Base Model. Included in this profile are some general parts of the Base Model for a basic understanding. Please refer to the PROFIdrive profile for a full description of the Base Model.

5.1.1 Communication Devices

The PROFIdrive Base Model, which is used for the Encoder Profile also, defines as basic elements the following three classes of devices:

- **Controller**: The Controller is a controlling device which is associated with one or more Encoders. Related to the automation system, the Controller is the host for the overall automation.
- **P-Device**: The P-Device (peripheral device) is a field device and the host device for the Encoder Unit. The P-Device typically is associated with one or more Controller devices.
- **Supervisor**: The Supervisor typically is an engineering device which manages provisions of configuration data (parameter sets) and collections of diagnosis data from P-Devices and/or Controllers.

5.1.2 Communication Relationship

The Base Model defines the following types of communication relationships between the Devices in an automation system:

Following figure shows the Automation Devices and the defined relationships between them in the context of the Base Model used for the Encoder profile.

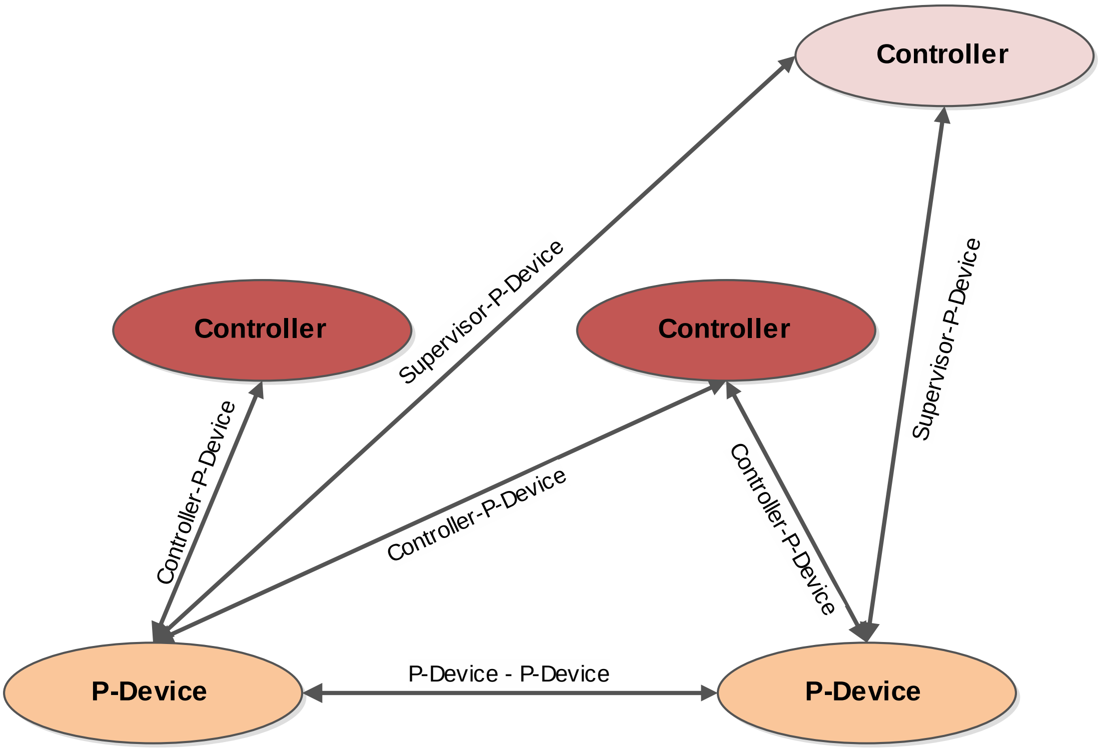


Figure 2: Automation Devices and their relationship

6 Encoder Object 3D00h

API	Object Name	Description
0x3D00	Encoder-Object (EO)	Encoder-Object (EO): implements PROFINET/PROFIBUS PROFIdrive Encoder profile.

6.1 EO IO Data

EO IO Data contains all data transmitted cyclically between the EO and other devices (controller, supervisor and device). The EO IO Data is structured in the same way than DO IO Data in the PROFIdrive profile [1] in Signals and Telegrams. The EO telegram is an aggregation of one or multiple Signals. For further details see the PROFIdrive profile definition and the encoder profile specific definitions and extensions in this clause

6.2 Standard Signals

A series of standard signals are defined to configure the IO data. The signal numbers are taken from the PROFIdrive profile [1]. See Table 13 for the list of Signals used in this profile.

No	Sinificance	Abbrevication	Data Type	Impl.
6	Speed acual value 16 Bit	NIST_A	INT16	Class 4
8	Speed acual value 32 Bit	NIST_B	INT32	Class 2
9	Sensor 1 control word	G1_STW	WORD	Class 3,4
10	Sensor 1 status word	G1_ZSW		Class 3,4
11	Sensor 1 position actual value 1	G1_XIST1		Class 3,4
12	Sensor 1 position actual value 2	G1_XIST2		Class 3,4
39	Sensor 1 position actual value 3	G1_XIST3		Optional
40..79	PROFIdrive profile specific			Reserved
80	Encoder control word 2	STW2_ENC		Class 1,2,3,4
81	Encoder status word 2	ZSW2_ENC		Class 1,2,3,4
82	Encoder preset control word 31 bit with trigger bit included	G1_XIST_PRESET _B		Class 1,2
83	Encoder preset control word 64 bit	G1_XIST_PRESET _C		Optional
84	Encoder preset control word 32 bit	G1_XIST_PRESET _B1		Class 1,2
85..88	Encoder profile specific			Reserved
89..59999				Reserved
60000..60999	Encoder vendor specific			Optional
61000..65535				Reserved

Table 3: Standard Signal definitions

6.3 Standard Telegrams

No	Name	Implementation				IO Data length		Semantic
		Class	Class	Class	Class	Outpu	Input	
81	Standard Telegram 81	M	M	M	M	4B /	12B/6W	
82	Standard Telegram 82	O	O	O	M	4B /	14B/7W	
83	Standard Telegram 83	O	O	O	O	4B /	16B/8W	
84	Standard Telegram 84	O	O	O	O	4B /	20B/5W	
86	Standard Telegram 86	O	M	O	O	4B /	8B/4W	
87	Standard Telegram 87	M	M	O	O	4B /	4B/2W	
88	Standard Telegram 88	O	O	O	O	8B /	12B/6W	
89	Standard Telegram 89	M	M	O	O	6B /	10B/5W	
▪ **M**: Mandatory, **O**: Optional, **B**: Byte, **W**: Word								

Table 4: Standard Telegrams

6.3.1 Standard Telegram 81

This telegram contains dictionary order possibilities making stati accessible and position data up to 32 Bit, which is mating to encoders up to resolution 12 Bit MT + 20 Bit ST or 22 Bit only ST

6.3.1.1 Setpoint

Output	Word Offset	Signal	Bit	Function	Implementation	
					Class 3	Class 4
Setpoi	0	STW2_ENC	15-12	Controller Sign-Of-Life	M	M
			11	Reserved		
			10	Control By PLC	M	M
			9,8	Reserved		
			7	Fault Acknowledge	O	O
			6-1	Reserved		
			0	Preset trigger		
	1	G1_STW	15	Acknowledge sensor error	M	M
			14	Activate parking	M	M
			13	Request absolute value cyclicylly	M	M
			12	Request set/shift of home position	O	M
			11	Home position mode	O	M
			10-8	Reserved		
			7-0	Function Request	O	O
▪ **M**: Mandatory, **O**: Optional						

Table 5: Standard Telegram 81 Setpoint

6.3.1.2 Actual value

Input	Word	Signal	Bit	Function	Implementation	
					Class 3	Class 4
Actual value	0	ZSW2_ENC	15-12	Encoder Sign-Of-Life	M	M
			11	Reserved		
			10	Control requested	M	M
			9,8	Reserved		
			7	Warning present / no warning	M	M
			6-4	Reserved		
			3	Fault present / no Fault	M	M
			2	NIST_VALID		
			1	XIST_VALD		
			0	XIST_PRESET_ACK		
	1	G1_ZSW	15	Sensor error	M	M
			14	Parking sensor active	M	M
			13	Transmit absolute value cyclicylly	M	M
			12	set/shift of home position executed	O	M
			11	Error acknowledement detected	M	M
			10	Reserved, set to zero		
			9	Probe 2 deflected	O	O
			8	Probe 1 deflected	O	O
			7-0	Function status	O	O
	2	G1_XIST1 (MSW)	31-16	Sensor 1 position actual value 1 High-Word		
	3	G1_XIST1 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		
	4	G1_XIST2 (MSW)	31-16	Sensor 1 position actual value 2 High-Word		
	5	G1_XIST2 (LSW)	15-0	Sensor 1 position actual value 2 Low-Word		
▪	**M**: Mandatory, **O**: Optional					

Table 6: Standard Telegram 81 Actual value

6.3.2 Standard Telegram 82

This telegram adds to telegram 81 rough speed information (16 Bit)

6.3.2.1 Setpoint

For more detail please refer to Standard Telegram 81 Setpoint

6.3.2.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 3	Class 4
Actual value	0	ZSW2_ENC	15-12	Encoder Sign-Of-Life	M	M
			11	Reserved		
			10	Control requested	M	M
			9,8	Reserved		
			7	Warning present / no warning	M	M
			6-4	Reserved		
			3	Fault present / no Fault	M	M
			2	NIST_VALID		
			1	XIST_VALD		
			0	XIST_PRESET_ACK		
	1	G1_ZSW	15	Sensor error	M	M
			14	Parking sensor active	M	M
			13	Transmit absolute value cyclicylly	M	M
			12	set/shift of home position executed	O	M
			11	Error acknowledement detected	M	M
			10	Reserved, set to zero		
			9	Probe 2 deflected	O	O
			8	Probe 1 deflected	O	O
			7-0	Function status	O	O
	2	G1_XIST1 (MSW)	31-16	Sensor 1 position actual value 1 High-Word		
	3	G1_XIST1 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		
	4	G1_XIST2 (MSW)	31-16	Sensor 1 position actual value 2 High-Word		
	5	G1_XIST2 (LSW)	15-0	Sensor 1 position actual value 2 Low-Word		
	6	NIST_A	15-0	Speed acual value 16 Bit		
▪ **M**: Mandatory, **O**: Optional						

Table 7: Standard Telegram 82 Actual Value

6.3.3 Standard Telegram 83

This telegram differs from telegram 82 in offering fine speed information (32 Bit)

6.3.3.1 Setpoint

For more detail please refer to Standard Telegram 81 Setpoint

6.3.3.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation Class 3	Implementation Class 4
Actual value	0	ZSW2_ENC	15-12	Encoder Sign-Of-Life	M	M
			11	Reserved		
			10	Control requested	M	M
			9,8	Reserved		
			7	Warning present / no warning	M	M
			6-4	Reserved		
			3	Fault present / no Fault	M	M
			2	NIST_VALID		
			1	XIST_VALD		
			0	XIST_PRESET_ACK		
	1	G1_ZSW	15	Sensor error	M	M
			14	Parking sensor active	M	M
			13	Transmit absolute value cyclicylly	M	M
			12	set/shift of home position executed	O	M
			11	Error acknowledement detected	M	M
			10	Reserved, set to zero		
			9	Probe 2 deflected	O	O
			8	Probe 1 deflected	O	O
			7-0	Function status	O	O
	2	G1_XIST1 (MSW)	31-16	Sensor 1 position actual value 1 High-Word		
	3	G1_XIST1 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		
	4	G1_XIST2 (MSW)	31-16	Sensor 1 position actual value 2 High-Word		
	5	G1_XIST2 (LSW)	15-0	Sensor 1 position actual value 2 Low-Word		
	6	NIST_B (LSW)	31-16	Speed acual value 32 Bit High-Word		
	7	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low-Word		
▪ **M**: Mandatory, **O**: Optional						

Table 8: Standard Telegram 83 Actual Value

6.3.4 Standard Telegram 84

This telegram differs from telegram 83 in offering position data 64 Bit, means, it is needed for resolutions above 12 Bit MT + 20 Bit ST, i.e. 12 Bit MT + 22 Bit ST

6.3.4.1 Setpoint

For more detail please refer to Standard Telegram 81 Setpoint

6.3.4.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 3	Class 4
Actual value	0	ZSW2_ENC	15-12	Encoder Sign-Of-Life	M	M
			11	Reserved		
			10	Control requested	M	M
			9,8	Reserved		
			7	Warning present / no warning	M	M
			6-4	Reserved		
			3	Fault present / no Fault	M	M
			2	NIST_VALID		
			1	XIST_VALD		
			0	XIST_PRESET_ACK		
	1	G1_ZSW	15	Sensor error	M	M
			14	Parking sensor active	M	M
			13	Transmit absolute value cyclicylly	M	M
			12	set/shift of home position executed	O	M
			11	Error acknowledement detected	M	M
			10	Reserved, set to zero		
			9	Probe 2 deflected	O	O
			8	Probe 1 deflected	O	O
			7-0	Function status	O	O
	2	G1_XIST3 (W3)	63-48	Sensor 1 position actual value 3 Word-3		
	3	G1_XIST3 (W2)	47-32	Sensor 1 position actual value 3 Word-2		
	4	G1_XIST3 (W1)	31-16	Sensor 1 position actual value 3 Word-1		
	5	G1_XIST3 (W0)	15-0	Sensor 1 position actual value 3 Word-0		
	6	G1_XIST2 (MSW)	31-16	Sensor 1 position actual value 2 High-Word		
	7	G1_XIST2 (LSW	16-0	Sensor 1 position actual value 2 Low -Word		
	8	NIST_B (LSW)	31-16	Speed acual value 32 Bit High		

	9	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low		
▪ **M**: Mandatory, **O**: Optional						

Table 9: Standard Telegram 84 Actual Value

6.3.5 Standard Telegram 86

This telegram offers direct preset function and position data up to 32 Bit, which is mating to encoders up to resolution 12 Bit MT + 20 Bit ST or 22 Bit only ST as well as fine speed information (32 Bit)

This is equal to the former Siemens-telegram 860

6.3.5.1 Setpoint

Output Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 1	Class 2
Setpoint	0	G1_XIST_PRESET_B (MSW)	31	Preset trigger		M
			30-16	Sensor preset value High-Word		
	1	G1_XIST_PRESET_B (LSW)	15-0	Sensor preset value Low-word		M
▪ **M**: Mandatory, **O**: Optional						

Table 10: Standard Telegram 86 Setpoint

6.3.5.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 1	Class 2
Actual value	0	G1_XIST1 (MSW)	31-16	Sensor 1 position actual value 1 High-Word		M
	1	G1_XIST1 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		M
	2	NIST_B (LSW)	31-16	Speed acual value 32 Bit High-Word		M
	3	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low-Word		M
▪ **M**: Mandatory, **O**: Optional						

Table 11: Standard Telegram 86 Actual value

6.3.6 Standard Telegram 87

This telegram differs from telegram 86 in not offering speed information

This is equal to the former Siemens-telegram 870

6.3.6.1 Setpoint

Output Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 1	Class 2
Setpoint	0	G1_XIST_PRESET_B (MSW)	31	Preset trigger	M	M
			30-16	Sensor preset value High-Word	M	M
	1	G1_XIST_PRESET_B (LSW)	15-0	Sensor preset value Low-word	M	M
▪ **M**: Mandatory, **O**: Optional						

Table 12: Standard Telegram 87 Setpoint

6.3.6.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 1	Class 2
Actual value	0	G1_XIST1 (MSW)	31-16	Sensor 1 position actual value 1 High-Word	M	M
	1	G1_XIST1 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word	M	M
▪ **M**: Mandatory, **O**: Optional						

Table 13: Standard Telegram 87 Actual value

6.3.7 Standard Telegram 88

This telegram differs from telegram 86 in offering 64 Bit direct Preset instead of 32 Bit
And offering 64 Bit position instead of 32 Bit – good for higher resolutions above 1220

6.3.7.1 Setpoint

Output	Word Offset	Signal	Bit	Function	Implementation	
					Class	Class
Setpoint	0	G1_XIST_PRESET_C (W3)	63	Preset trigger		
			62-48	Sensor preset value Word-3		
	1	G1_XIST_PRESET_C (W2)	47-32	Sensor preset value Word-2		
	2	G1_XIST_PRESET_C (W1)	31-16	Sensor preset value Word-1		
	3	G1_XIST_PRESET_C (W0)	15-0	Sensor preset value Word-0		
M: Mandatory, **O**: Optional						

Table 14: Standard Telegram 88 Setpoint

6.3.7.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation	
					Class	Class
Actual value	0	G1_XIST3 (W3)	63-48	Sensor 1 position actual value 1 Word-3		
	1	G1_XIST1 (W2)	47-32	Sensor 1 position actual value 1 Word-2		
	2	G1_XIST1 (W1)	31-16	Sensor 1 position actual value 1 Word-1		
	3	G1_XIST1 (W0)	15-0	Sensor 1 position actual value 1 Word-0		
	4	NIST_B (MSW)	15-0	Sensor 1 position actual value 1 High-Word		
	5	NIST_B (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		
▪ **M**: Mandatory, **O**: Optional						

Table 15: Standard Telegram 88 Actual value

6.3.8 Standard Telegram 89

This telegram adds to telegram 86 dictionary order possibilities

6.3.8.1 Setpoint

Output Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 1	Class 2
Setpoint	0	STW2_ENC	15-12	Controller Sign-Of-Life	M	M
			11	Reserved		
			10	Control By PLC	M	M
			9,8	Reserved		
			7	Fault Acknowledge	O	O
			6-1	Reserved		
			0	Preset trigger		
	1	G1_XIST_PRESET_B1 (MSW)	31-16	Encoder preset control word 32 bit High-Word	M	M
	2	G1_XIST_PRESET_B1 (LSW)	15-0	Encoder preset control word 32 bit Low-Word		
▪ **M**: Mandatory, **O**: Optional						

6.3.8.2 Actual value

Input Data	Word Offset	Signal	Bit	Function	Implementation	
					Class 1	Class 2
Actual value	0	ZSW2_ENC	15-12	Encoder Sign-Of-Life	M	M
			11	Reserved		
			10	Control requested	M	M
			9,8	Reserved		
			7	Warning present / no warning	M	M
			6-4	Reserved		
			3	Fault present / no Fault	M	M
			2	NIST_VALID		
			1	XIST_VALD		
			0	XIST_PRESET_ACK		
	1	G1_XIST1 (MSW)	31-16	Sensor 1 position actual value 1 High-Word		
	2	G1_XIST1 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		
	3	NIST_B (LSW)	31-16	Speed acual value 32 Bit High-Word		
	4	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low-Word		
▪ **M**: Mandatory, **O**: Optional						

Table 16: Standard Telegram 89 Actual Value

6.4 OEM Telegrams

No	Name	Implementation				IO Data length		Semantic
		Class 1	Class 2	Class 3	Class 4	Output	Input	
100	OEM Telegram 100	O	O	O	O	4B / 2W	12B/6W	
101	OEM Telegram 101	O	O	O	O	4B / 2W	14B/7W	
860	OEM Telegram 860	O	O	O	O	4B / 2W	14B/7W	
▪	**M**: Mandatory, **O**: Optional, **B**: Byte, **W**: Word							

6.4.1 OEM Telegram 100

This telegram adds to telegram 86 acceleration information (32 Bit)

6.4.1.1 Setpoint

Output Data	Word Offset	Signal	Bit	Function
Setpoint	0	G1_XIST_PRESET_B (MSW)	31	Preset trigger
			30-16	Sensor preset value High-Word
	1	G1_XIST_PRESET_B (LSW)	15-0	Sensor preset value Low-word

Table 17: OEM Telegram 100 Setpoint

6.4.1.2 Actual Value

Input Data	Word Offset	Signal	Bit	Function
Actual value	0	G1_XIST2 (MSW)	31-16	Sensor 1 position actual value 1 High-Word
	1	G1_XIST2 (LSW)	15-0	Sensor 1 position actual value 1 Low-Word
	2	NIST_B (LSW)	31-16	Speed acual value 32 Bit High-Word
	3	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low-Word
	4	ACC_B (MSW)	31-16	Acceleration actual value 32 Bit High-Word
	5	ACC_B (LSW)	15-0	Acceleration acual value 32 Bit Low-Word

Table 18: OEM Telegram 100 Actual value

6.4.2 OEM Telegram 101

This telegram adds to telegram 88 acceleration information (32 Bit)

6.4.2.1 Setpoint

Output Data	Word Offset	Signal	Bit	Function	Implementation Class	Class
Setpoint	0	G1_XIST_PRESET _C (W3)	63	Preset trigger		
			62-48	Sensor preset value Word-3		
	1	G1_XIST_PRESET _C (W2)	47-32	Sensor preset value Word-2		
	2	G1_XIST_PRESET _C (W1)	31-16	Sensor preset value Word-1		
	3	G1_XIST_PRESET _C (W0)	15-0	Sensor preset value Word-0		
▪ **M**: Mandatory, **O**: Optional						

Table 19: OEM Telegram 101 Setpoint

6.4.2.2 Actual Value

Input Data	Word Offset	Signal	Bit	Function
Actual value	0	Position (W3)	63-48	Sensor 1 position actual value 1 Word-3
	1	Position (W2)	47-32	Sensor 1 position actual value 1 Word-2
	2	Position (W1)	31-16	Sensor 1 position actual value 1 Word-1
	3	Position (W0)	15-0	Sensor 1 position actual value 1 Word-0
	4	NIST_B (LSW)	31-16	Speed acual value 32 Bit High-Word
	5	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low-Word
	6	ACC_B (MSW)	31-16	Acceleration actual value 32 Bit High-Word
	7	ACC_B (LSW)	15-0	Acceleration acual value 32 Bit Low-Word

Table 20: OEM Telegram 101 Actual value

6.4.3 OEM Telegram 860

This telegram is similar to telegram 86 and represents the former Siemens OEM type

6.4.3.1 Setpoint

Output Data	Word Offset	Signal	Bit	Function	Implementation Class	Class
Setpoint	0	G1_XIST_PRESET_B (MSW)	31	Preset trigger		
			30-16	Sensor preset value High-Word		
	1	G1_XIST_PRESET_B (LSW)	15-0	Sensor preset value Low-word		
▪ **M**: Mandatory, **O**: Optional						

Table 21: Standard Telegram 86 Setpoint = OEM Telegram 860

6.4.3.2 Actual Value

Input Data	Word Offset	Signal	Bit	Function	Implementation Class	Class
Actual value	0	Position (MSW)	31-16	Sensor 1 position actual value 1 High-Word		
	1	Position (LSW)	15-0	Sensor 1 position actual value 1 Low-Word		
	2	NIST_B (LSW)	31-16	Speed acual value 32 Bit High-Word		
	3	NIST_B (LSW)	15-0	Speed acual value 32 Bit Low-Word		
▪ **M**: Mandatory, **O**: Optional						

Table 22: Standard Telegram 86 Actual value = OEM Telegram 860

6.5 Encoder Function Requirements

The encoder functions defined in this profile are divided upon four Encoder device classes, Class 1, Class 2, Class 3 and Class 4 with the implementation requirements of the different functions according to following Table:

Function	Implementation				Semantic
	C1	C2	C3	C4	
BMP parameter access channel	M	M	M	M	
Parameter control	M	M	M	M	
Code sequence	O	M	O	M	
Alarm channel	M	M	M	M	
NV-RAM	O	O	O	O	
Reset control	O	M	O	M	
PROFIdrive fault buffer	O	O	O	O	
Operating time	O	O	O	O	
Scaling function	O	M	O	M	
Preset control	O	M			
PROFIdrive position feedback interface			M	M	
Clock synchronous operation	O	O	M	M	
Set/shift home position			O	M	
V3.1 compatibility mode			M	M	
Velocity actual value (NIST_x)	O	M	O	O	
▪ **M**: Mandatory, **O**: Optional, **C**: Class					

Table 23: Encoder Function Requirements

6.5.1 Base Mode Parameter access channel

6.5.1.1 Encoder Standard Parameter

BMP	Byte Offset	Bit Offset	Name	Data Type	Value	Description
Encoder Standard parameter	0	0	Code Sequence	Bit	0	CW, default
					1	CCW
	0	1	Class 4 Functionality	Bit	0	Disable
					1	Enable, default
	0	2	G1_XIST1 Preset Control	Bit	0	Enable
					1	Disable, default
	0	3	Scaling Control	Bit	0	Disable, default
					1	Enable
	0	4	Alarm Channel Control	Bit	0	Disable, default
					1	Enable
	0	5	V3.1 Compatibility Mode	Bit	0	Enable
					1	Disable, default
	0	6	Encoder Type	Bit	0	Rotary, default
					1	Linear
	1	0	Measuring Unit per Revolution High	UINT32	0-..	Not used / visible
	5	0	Measuring Unit per Revolution Low	UINT32	1-x	Encoder specific
	9	0	Total Measuring Range High	UINT32	0-x	Encoder specific
	13	0	Total Measuring Range Low	UINT32	2-x	Encoder specific
	17	0	Max. Controller Sign-Of-Life Failure	UINT8	0-255	Default: 1
	18	0	Velocity Unit	UINT8	0	Steps / second, default
					1	Steps / 100ms
					2	Steps / 10ms
					3	RPM
					4	N”N4
	19	0	Velocity Reference	Float32	0-x	Encoder specific

Table 24: BMP Encoder Standard Parameter

6.5.1.1.1 Addressing

- Interface: PROFINET Record Read/Write,
- API: 0x3D00
- Slot: 1
- Subslot: 1/2
- Record Index: 0xFFFF
- Record length: 31 Byte

6.5.1.2 Encoder Profile Parameter

BMP	Byte Offset	Bit Offset	Name	Data Type	Value	Description
Encoder Profile Parameter Control (65005)	0	0	Parameter Init control	BitArea	0	PRM
					1	NV-RAM (deafult)
					2	Not used
					3	Not used
	0	2	Parameter Write protect	BitArea	0	Write all (default)
					1	Read only
					2	Write Controller
					3	Write Supervisor
	0	5	Parameter 65005 write protect	Bit	0	Write all (default)
					1	Read only
	0	6	Reset control write protect	Bit	0	Write all (default)
					1	Read only

Table 25: Encoder BMP Profile Parameter

6.5.1.2.1 Addressing

- Interface: PROFINET Record Read/Write,
- API: 0x3D00
- Slot: 1
- Subslot: 1/2
- Record Index: 0x1001
- Record length: 2 Byte

6.5.1.3 Encoder OEM Parameter

If specific resolution is required in a project, this could be defined here, if different encoder will be used, which resolution differs from this value, PLC will go to Error

BMP	Byte Offset	Bit Offset	Name	Data Type	Value	Description
Encoder OEM Parameter Sensor Configuration	0	0	Singleturn Bits	UINT8	10	10 Bit
					11	11 Bit
					12	12 Bit
					13	13 Bit
					14	14 Bit
					15	15 Bit
					16	16 Bit
					17	17 Bit
					18	18 Bit
					19	19 Bit
					20	20 Bit
					21	21 Bit
					22	22 Bit
					33	Real Configuration (default)
	1	0	Multiturn Bits	UINT8	0	0 Bit
					4	4 Bit
					8	8 Bit
					12	12 Bit
					33	Real Configuration (default)

Table 26: BMP Encoder OEM Parameter

6.5.1.3.1 Addressing

- Interface: PROFINET Record Read/Write,
- API: 0x3D00
- Slot: 1
- Subslot: 1/2
- Record Index: 0x1000
- Record length: 2 Byte

6.6 Encoder Parameters

6.6.1 PROFIdrive Profile Parameter

PNU	ROC	Access	NV	Name	Data type	Description	Semantic
922	O	Get	NV	u16TlgSel	UINT16	Telegram Selection	
925	O	Get	NV	u16MasterSolCtrl	UINT16	Master Sign-Of-Life	
964	R	Get	NV	**tDuIdent**	**ARRAY of**	**Drive-Unit Ident**	**DU-specific**
				u16Duype	*UINT16*	Drive-Unit Type	
				u16SwVersion	*UINT16*	*Software Version*	
				u16FwDateYear	*UINT16*	*Firmware Date: Year*	
				u16FwDateDDMM	*UINT16*	*Firmware Date: DDMM*	
				u16NumOfDo	*UINT16*	*Number of Drive-Objects*	
965	R		NV	tProfileID	UINT16	Profile ID	
970	O	Set	V	u16Load	UINT16	Load command	
971	O	Set	V	u16Store	UINT16	Store command	
972	O	Set	V	u16Reset	UINT16	Reset command	DU-specific
974	O	Get	NV	tAccessServiceId	ARRAY of		PRM-974
975	O	Get	NV	**tEoIdent**	**ARRAY of**	**Encoder-Ident**	
				u16VendorID	*UINT16*	*Vendor-ID*	
				u16DoType	*UINT16*	*Drive-Object Type*	
				u16SwVersion	*UINT16*	*Software Version*	
				u16FwDateYear	*UINT16*	*Firmware Date: Year*	
				u16FwDateDDMM	*UINT16*	*Firmware Date: DDMM*	
				u16DoTypeClass	*UINT16*	*Drive-Object Type Class*	
				u16DoSubClass	*UINT16*	*Drive-Object Sub-Class*	
				u16DoID	*UINT16*	*Drive-Object-ID*	
976	O	Set	V	u16Load	UINT16	DU-Load command	DU specific
977	O	Set	V	u16Store	UINT16	DU-Store command	DU-specific
978	O	Get	NV	tDoList	ARRAY of	List of Drive-Objects	DU-specific
				DO-ID (1)	UINT16	Drive-Object ID	
979	R	Get	NV	tSensorFormat	ARRAY of	Sensor Format	SENSOR-FORMAT
				tHeader	UINT32	Header	Subindex: 0
				u32SensorType	UINT32	Sensor type	Subindex: 1
				u32SensorResolution	UINT32	Sensor Resolution	Subindex: 2
				u32Shiftfactor_G1_XIST1	UINT32	Shift factor G1_XIST1	Subindex: 3
				u32Shiftfactor_G1_XIST2	UINT32	Shift factor G1_XIST2	Subindex: 4

				u32Determinable Rev	UINT32	Determinable Revolution	Subindex: 5
				u32Reservd	UINT32	reserved	Subindex: 6
				u32Reservd	UINT32	reserved	Subindex: 7
				u32Reservd	UINT32	reserved	Subindex: 8
				u32Reservd	UINT32	reserved	Subindex: 9
				u32Reservd	UINT32	reserved	Subindex: 10
980	R	Get	V	ParameterList	ARRAY of UINT16	Contains List of Parameter	
60000	R	Set	NV	fVelRef	FLOAT32	Velocity Reference	
60001	R	Set	NV	u16VelNorm	UINT16	Velocity Norm (Unit)	
61000	O	Get	NV	tStationName	SHORT-STR	Station Name	DU-specific
61001	O	Get	NV	u32StationIp	UINT23	Station IP-Address	DU-specific
61002	O	Get	NV	abStationMac	ARRAY of BYTE	Station MAC Address	DU-specific
61003	O	Get	NV	u32StationGatew ay	UINT32	Station Gateway	DU-specific
61004	O	Get	NV	u32StationSubne t	UINT32	Station Subnet	DU-specific
65000	R	Set	NV	tPreset	INT32	32 Bit Preset	
65001	R	Get	V	tOpStatus	ARRAY of	Operating Status	
				u32BlkHeader	UINT32	Block Header	Subindex: 0
				32OpStatus	UINT32	Operating status	Subindex: 1
				u32Faults	UINT32	Faults	Subindex: 2
				u32SuppFaults	UINT32	Supported Faults	Subindex: 3
				u32Warnings	UINT32	Warnings	Subindex: 4
				u32SuppWarning s	UINT32	Supported warnings	Subindex: 5
				u32EoProfileVer	UINT32	Encoder Profile Version	Subindex: 6
				u32OperatingTim e	UINT32	Operating time	Subindex: 7
				u32Offset	UINT32	Offset	Subindex: 8
				u32MeasurUnit	UINT32	Measuring Unit per Rev.	Subindex: 9
				u32TotalRes	UINT32	Total Resolution	Subindex: 10
				u32VelocityNorm	UINT32	Velocity Norm	Subindex: 11
				u32VelocityRef	UINT32	Velocity Ref	Subindex: 12
				u32MeasurUnitM sw	UINT32	MSW Measuring Unit	Subindex: 13
				u32MeasurUnitLs w	UINT32	LSW Measuring Unit	Subindex: 14
				u32TotalResMsw	UINT32	MSW Total Resolution	Subindex: 15
				u32TotalResLsw	UINT32	LSW Total Resolution	Subindex: 16
				u32OffsetMsw	UINT32	MSW Offset	Subindex: 17
				u32OffsetLsw	UINT32	LSW Offset	Subindex: 18
65002	R	Set	NV	tPreset	INT64	64 Bit Preset	
65004	R	Set	NV	tFunCtrl	UINT32	Function Control	
65005	R	Set	NV	tPrmCtrl	UINT32	Parameter Control	

65006	R	Set	NV	tMeasurUnit	UINT32	32 Bit Measuring Unit per Revolution	
65007	R	Set	NV	tTotalRes	UINT32	32 Bit Total Resolution	
65008	R	Set	NV	tMeasurUnit	UINT64	64 Bit Measuring Unit per Revolution	
65009	R	Set	NV	tTotalRes	UINT64	64 Bit Total Resolution	
65010	R	Get	NV	u32OpTime	UINT32	Operating Time	
ROC: **R**: Required, **O**: Optional, **C**: Conditional **NV**: Non-Volatile, **V**: Volatile							

Table 27: BMP Profidrive Profile Parameter

6.6.2 OEM parameter

PNU	ROC	Access	NV	Name	Data type	Description	Semantic
100	R	Get	NV	tOffset	UINT64	Offset value changed by preset function.	
101	R	Set	NV	**tFormat**	**STRUCT of**	**Sensor Format**	**User Sensor Format**
				u8Single	*UINT8*	*Singleturn Bits*	
				u8Multi	*UINT8*	*Multiturn Bits*	
102	R	Get	NV	**tFreescale Nvd**	**STRUCT of**		
				u32Multiturn Pos	*UINT32*	*Multiturn Position*	
				i64CorrFactor	*INT64*	*Correction Factor*	
104	R	Set	V	u16FwUpdate	UINT16	Firmware Update: activates Firmware Loader.	
• **ROC**: **R**: Required, **O**: Optional, **C**: Conditional • **NV**: Non-Volatile, **V**: Volatile							

Table 28: BMP OEM Parameter

6.7 Measured Value Processing

6.7.1 Position Scaling

This section contains the processing chain to calculate the position value (G1_XIST*) from raw sensor data. Raw position in case of multi-turn encoder is defined in Formula 1.

Please notice, that the term "RawPosition" used within the following description describes the unscaled Position w/o Offest

```
RawPosition = (MultiturnValue * SingleTurnResolution) +
SingleturnValue
```
Formula 1: Calculation of raw position

6.7.2 Preset Function

The position value of an ACURO AC58 PROFINET encoder can be changed by the use of a preset value. This allows the encoder to be placed in any position when installed, and a preset value used to match the position output to the desired machine position. For example, the encoder can be made to read "0" when a machine is at its home position.

If it is required to store a preset value, one of the Telegrams including Preset value transmission needs to be chosen (Telegram 86, 87, 88, 89, 100, 101 or 860)
the encoder saves the offset between the actual encoder position value and the position value when the preset is applied. This offset value is then used if power is lost and then restored to ensure that the correct position value is reported even if the encoder shaft has been moved while power was off.

6.7.3 Residual Value Calculation

If the encoder is in continuous operation and the factor between Scaled Range and Physical Range is not a power of two, a residual error would occur if Physical Range is reached. This is shown in Figure 3. To compensate this error - which occurs at every zero crossing of Physical Range – an offset has to be added. The calculation of this offset is called the residual value calculation, often used within Round Axis Applications.

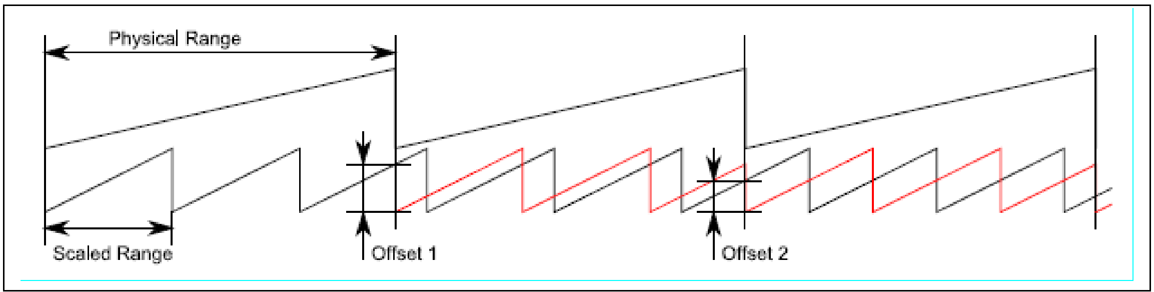


Figure 3: Errors due to non-integer multiples of Scaled Range to Physical Range

Residual value calculation is activated if "Scaling function control" enabled

The encoder saves the parameters which are needed for residual value calculation in the FRAM at every power off. This allows detecting a movement of up to 1020 rotations in powered-down state and correcting the residual value from the stored parameters after restarting the device.

NOTE! The residual value logic is only functional when any unpowered rotation of the multi-turn encoder does not exceed 1020 revolutions and when power is restored for a minimum of 11 continuous seconds while still within these 1020 rotations.

6.7.4 Speed Calculation

Raw speed is calculated as difference of the raw position values per integration time, as shown in Formula 5:

```
              RawPosition( t=0 ) - RawPosition( t = -IntegrationTime )
RawSpeed = ---------------------------------------------------------------
                                   IntegrationTime
```

Formula 5: Calculation of raw speed in unscaled measuring units

`IntegrationTime` will be automated chosen in respect of the actual speed

NOTE! The resulting scaled speed must fit into the data format, chosen with Telegram
Please choose Velocity measuring unit appropriate!

Due to quantization of position values, a leap in speed values is possible at slow speeds as shown in Figure 4.

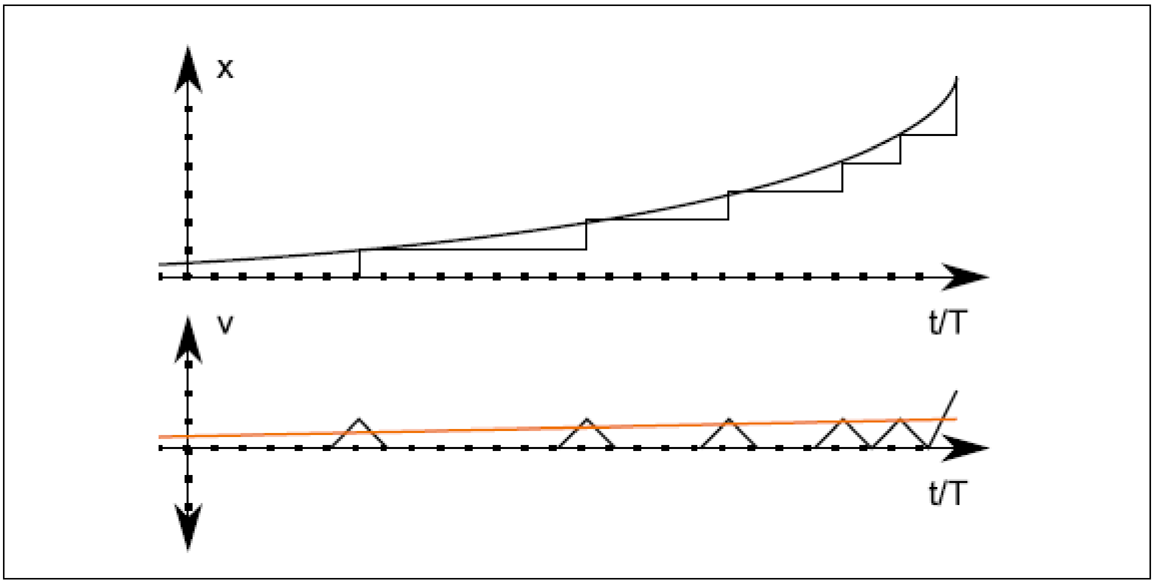


Figure 4: Leap in speed values

7 Startup procedure

As an aid for the use of standard ProfiNet-tools, there is a so-called GSD file (electronically data sheet) with device master information. It is available by download from our Internet homepage. The GSD(ML) file contains the available encoder parameters.

7.1 Example for configuration with STEP7/TIA v15

7.1.1 Startup RT-Mode

7.1.1.1 Startup program

⇨ Start Program

Figure 5: TIA-Portal-Icon

7.1.1.2 Select or create a project

For adding a new device, first open an existing Project or create a new project:
=> Click "create new project"

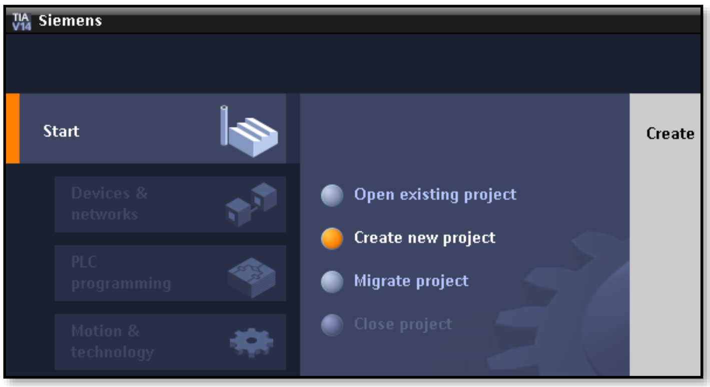


Figure 6: Dialog for to select a project

7.1.1.3 Add and configure device

Within a chosen project, select the Devices:

⇨ Select desired S7-CPU

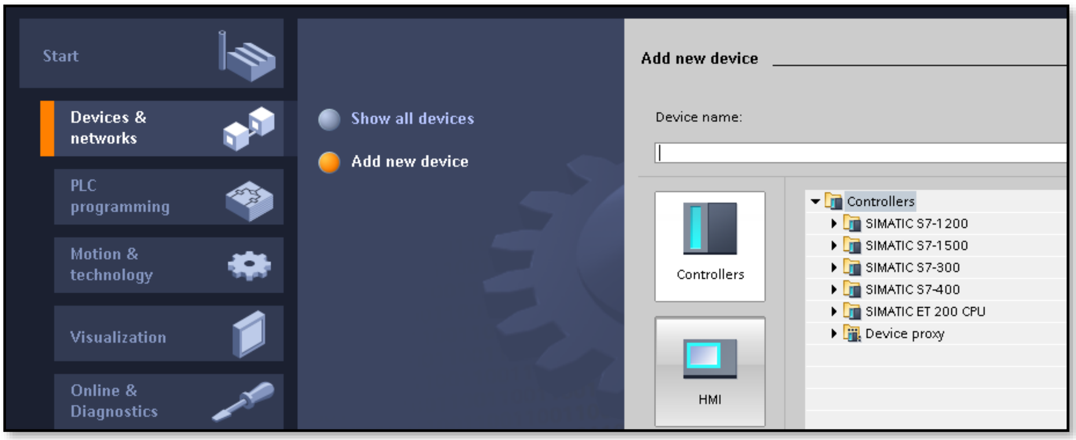


Figure 7: Dialog Select desired S7-CPU

⇨ choose S7-CPU according your available hardware

⇨ add S7-CPU in double-clicking on the right type

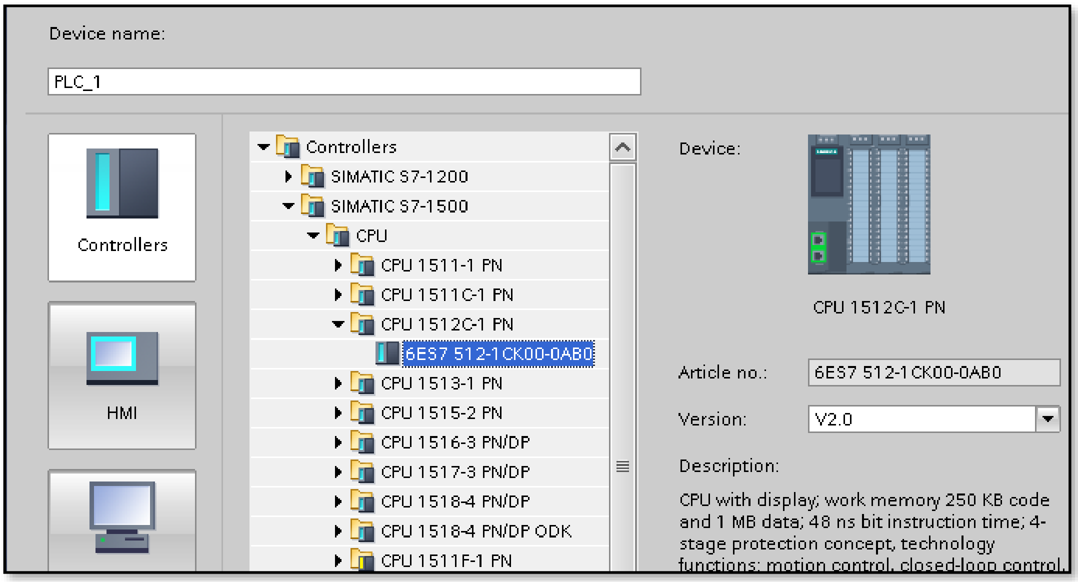


Figure 8: Selection Hierarchy

⇨ New CPU added!

⇨ Switch to CPU in Device-View

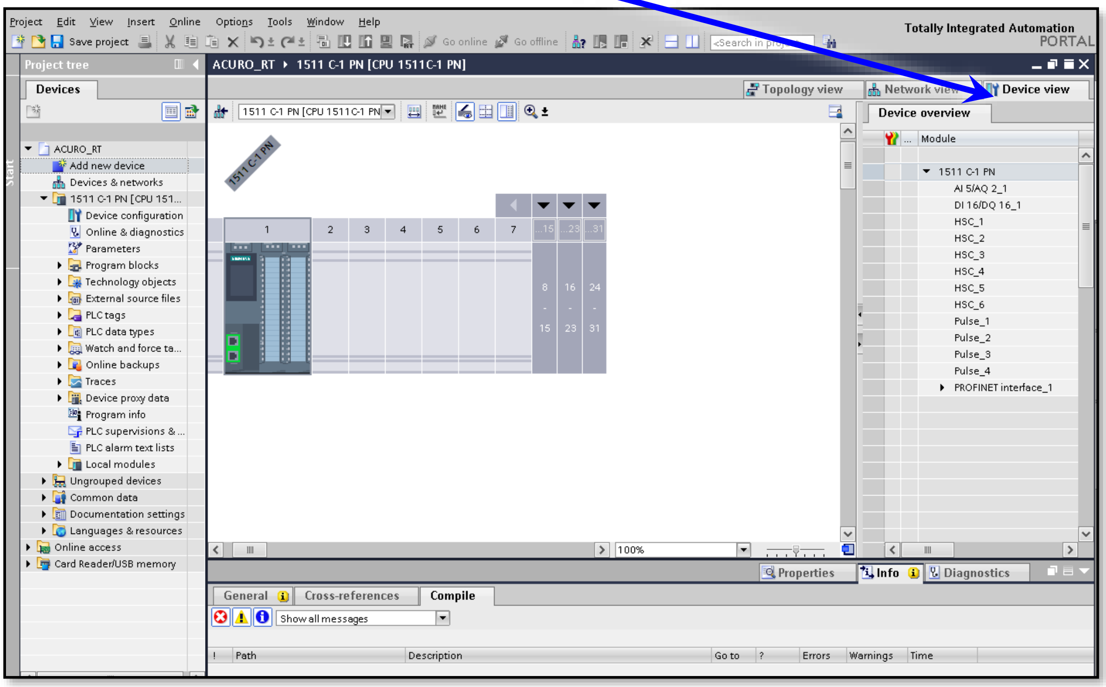


Figure 9: Device-View

7.1.1.4 Add new devices to device catalog by GSD(ML)

⇨ Select Manage GSD menu (install new GSDML)

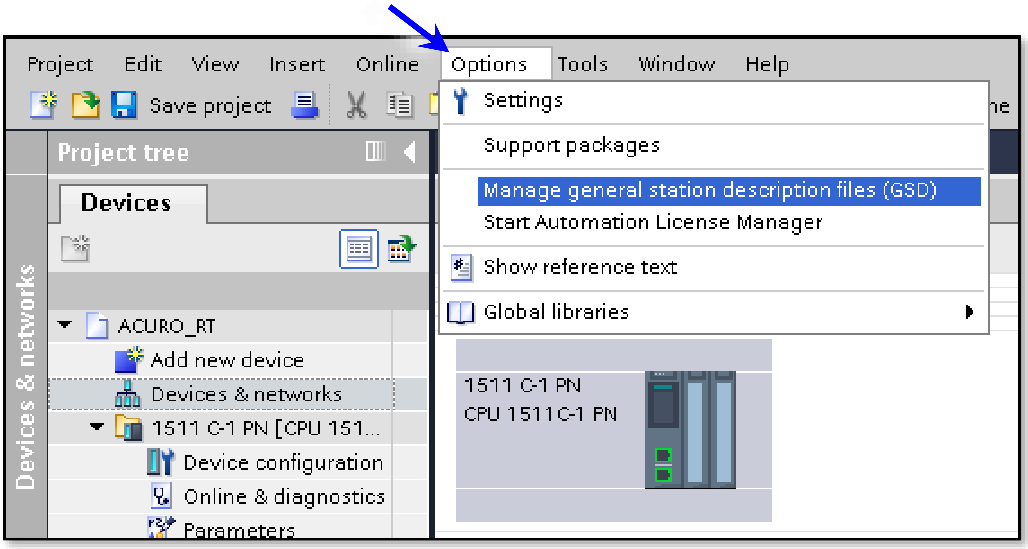


Figure 10: Options-Selection Box,

You need to have the Hengstler-GSD(ML) available, i.e. per download from our website, Unzip and store it to a chosen folder, our example shows folder “gsdml/GSDML-ACURO”

⇨ install ACURO GSDML in managing GSD files dialog

⇨ Select "..." to choose GSD(ML) folder/file for installation.

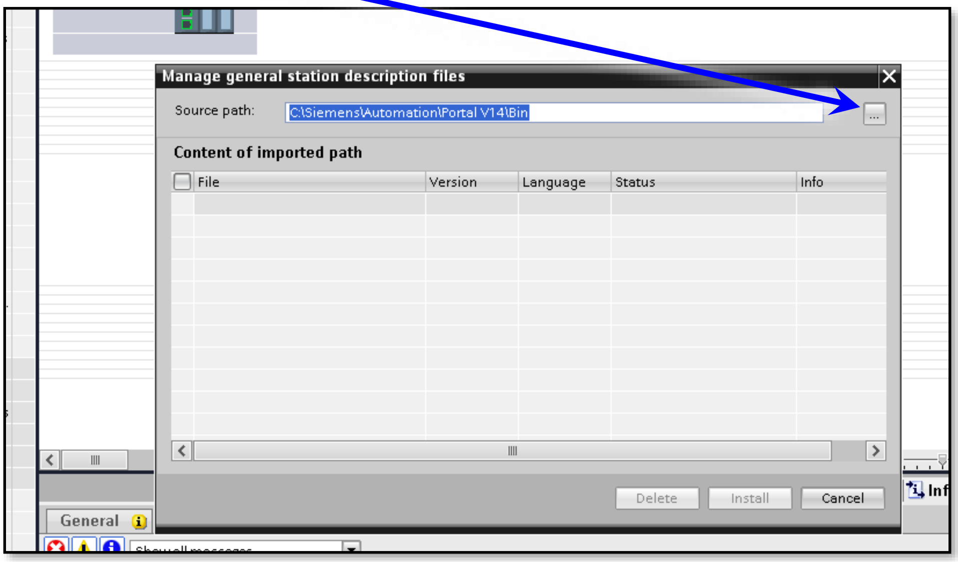


Figure 11: “Manage general station description files” - dialog-Box

⇨ Select GSDML directory:

⇨ Select directory where GSD(ML) is located (for installation).

Figure 12: Detail of Folder View

⇨ Select checkbox of GSD(ML) to install this file

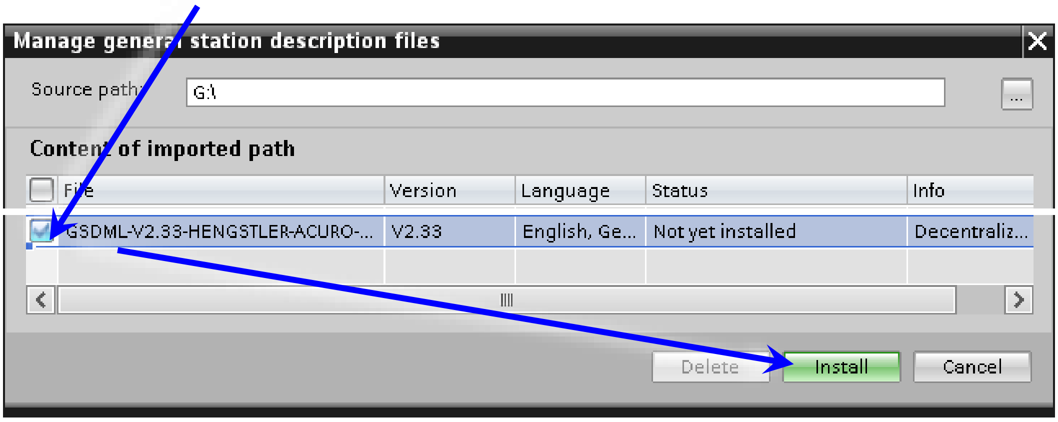


Figure 13: Visualization of Content of imported path

⇨ GSDML installation successful

⇨ Close dialog => Hardware Catalog will be updated

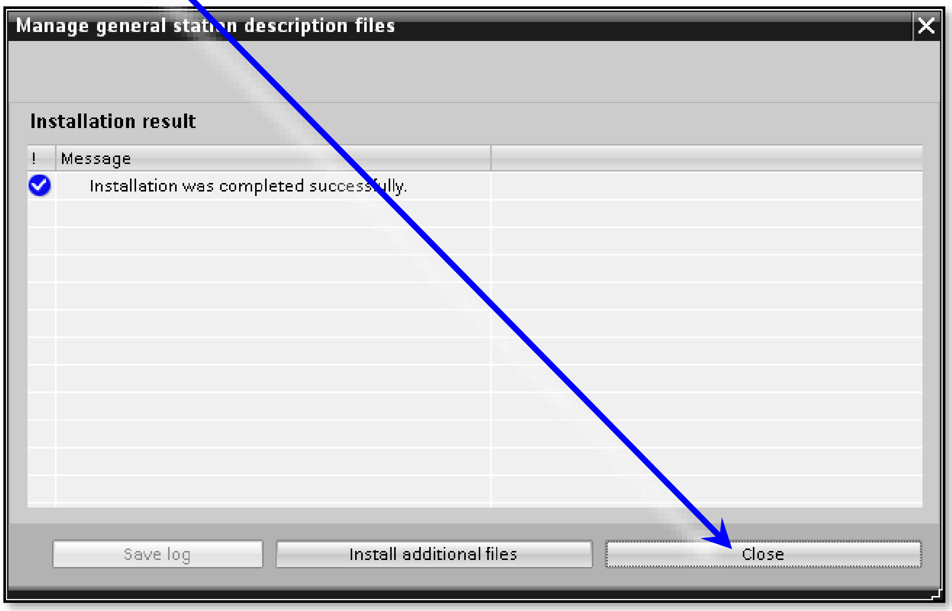


Figure 14: Installation result Dialog-Box - to be closed

⇨ Go back to Devices & networks

⇨ Open Hardware Catalog at the right edge of the main window (create encoder project)

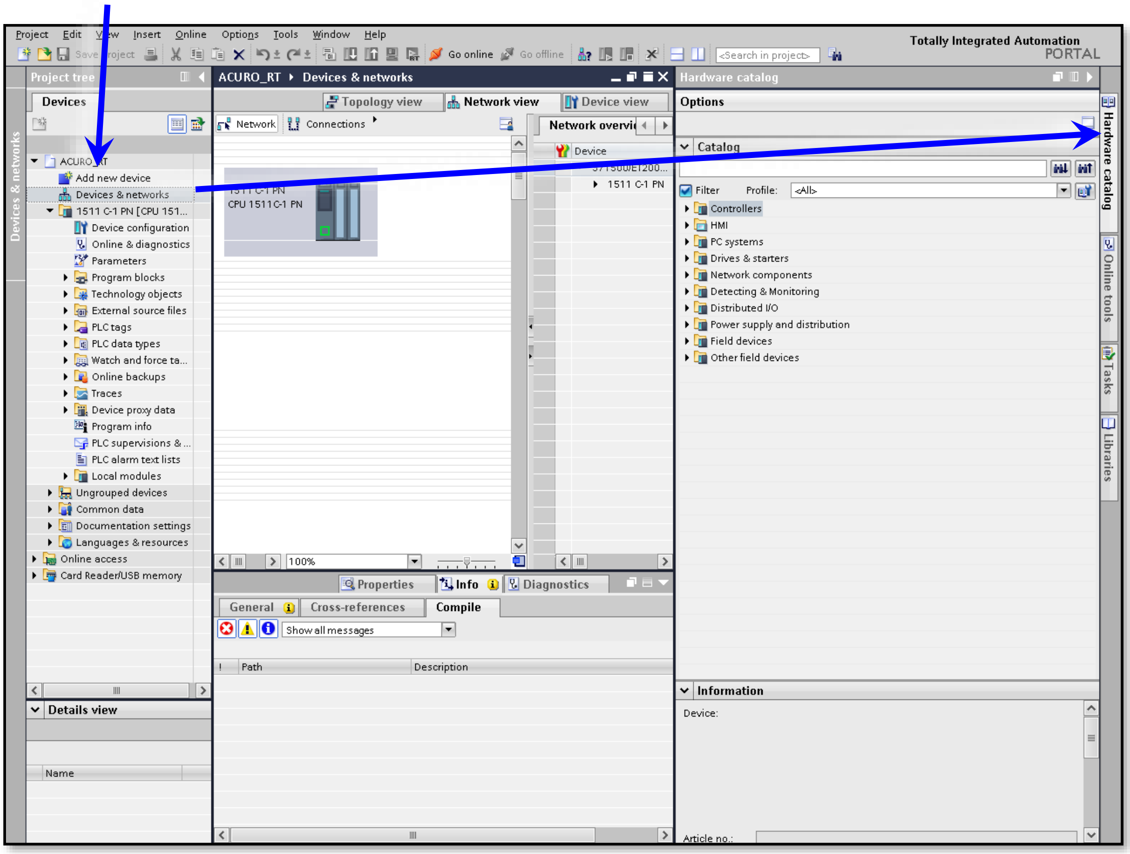


Figure 15: Main Window

HENGSTLER

⇨ Select encoder from hardware-catalog within "Other field devices"

⇨ Choose "Encoder Standard" for full version.
If not working, i.e. in case of older PLC-versions, choose "encoder No PDEV"

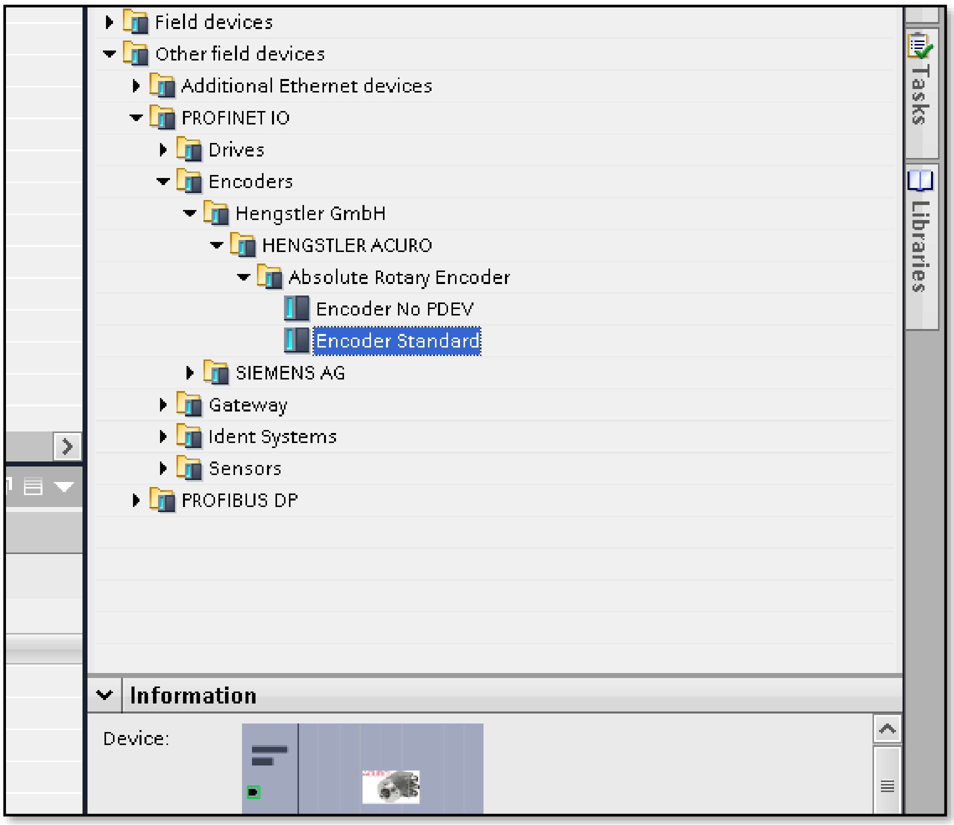


Figure 16: Path to be selected in Hardware Catalog

⇨ Paste Encoder into project by double-clicking on "Encoder Standard"
This draws the selected encoder (ACURO) into network-view

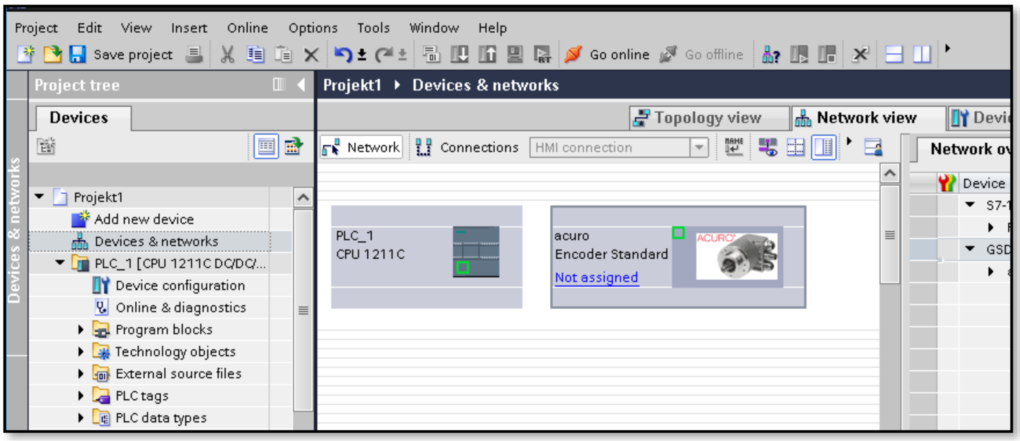


Figure 17: Network view

⇨ Assign PROFINET Master

⇨ Click on "Not Assigned" and select master for this device

Or just draw a line between the small green boxes

Figure 18: Detail in Network view with dialog box

⇨ Master assigned:

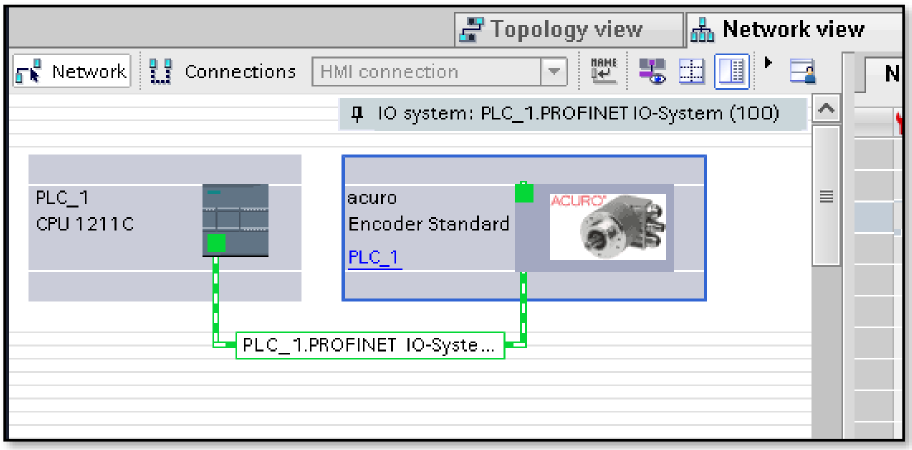


Figure 19: Network View showing Encoder assigned to Master (S7CPU)

⇨ Create Topology, in "Topology view", using the exact ports, to be used:

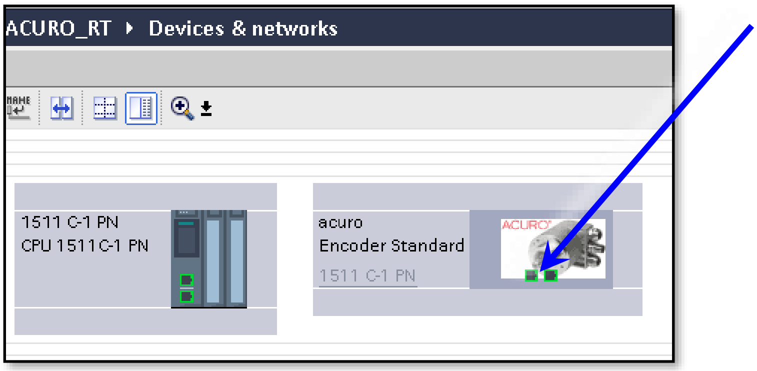


Figure 20: Topology-hierarchy shown in window

⇨ Connect ports in drawing line between meant ports

⇨ **Picture must be identical to real cabling**!

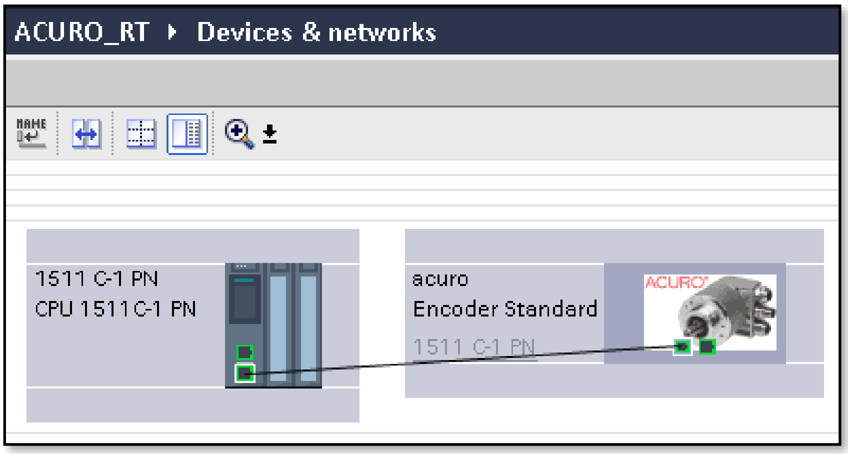


Figure 21: Graphical connection drawn

⇨ Ports connected, topology created:

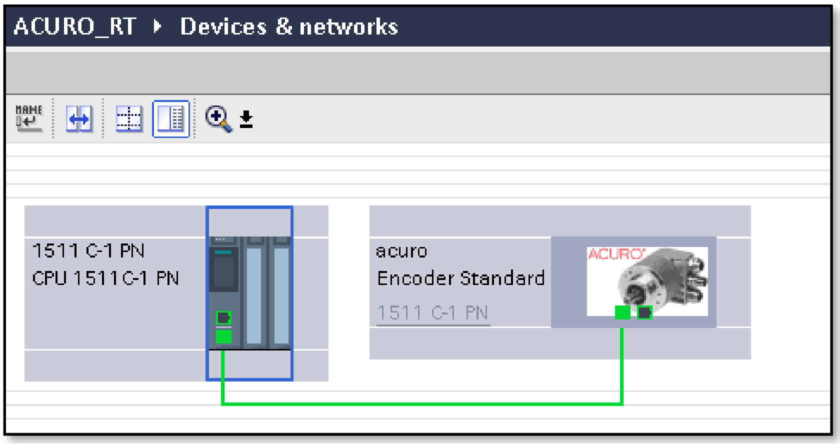


Figure 22: Window showing finished configuration

7.1.1.5 Configure Encoder

⇨ Switch to device-view (ACURO); by double-clicking ACURO Logo:

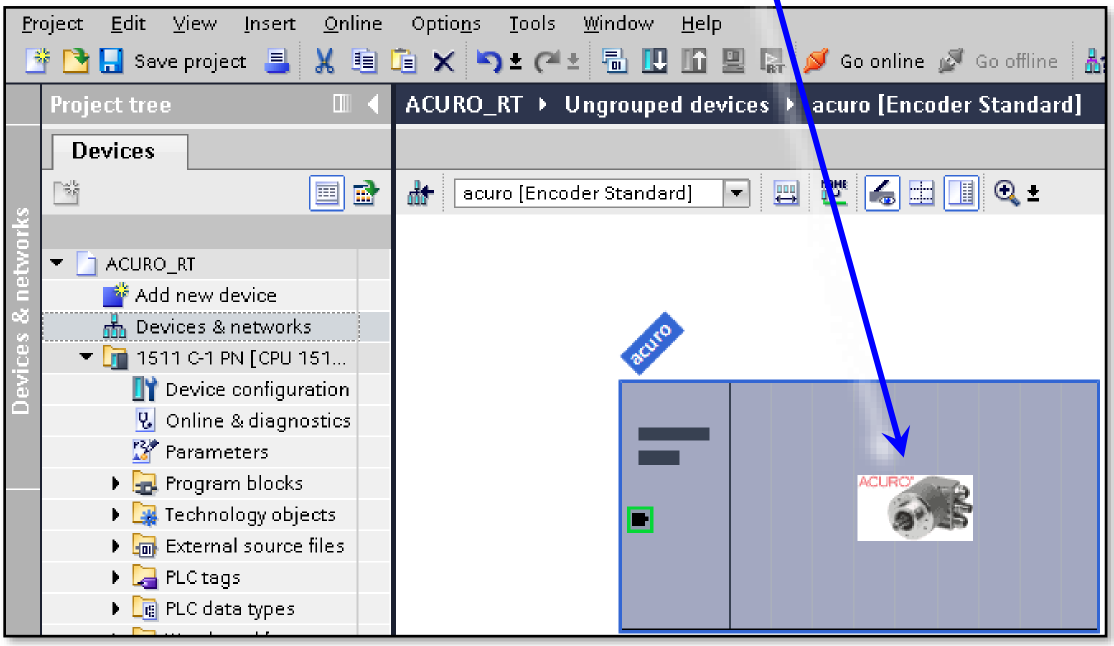


Figure 23: Device-View

⇨ Change device-name if necessary:

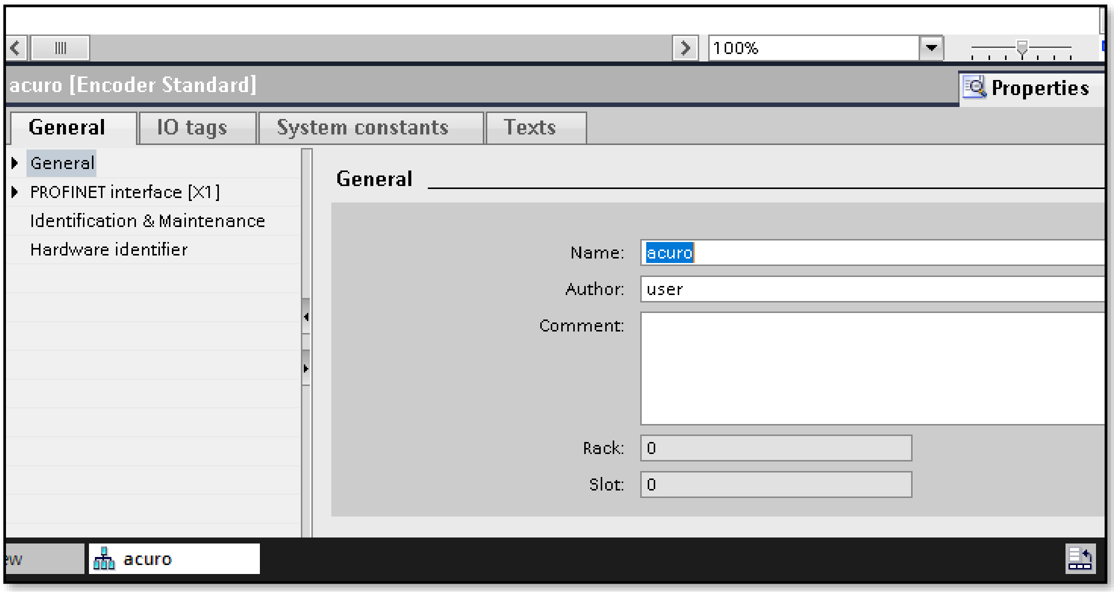


Figure 24: “General”-Tab in Device-view

⇨ Configure IP-address:

⇨ Select Configuration-window “Ethernet Addresses”

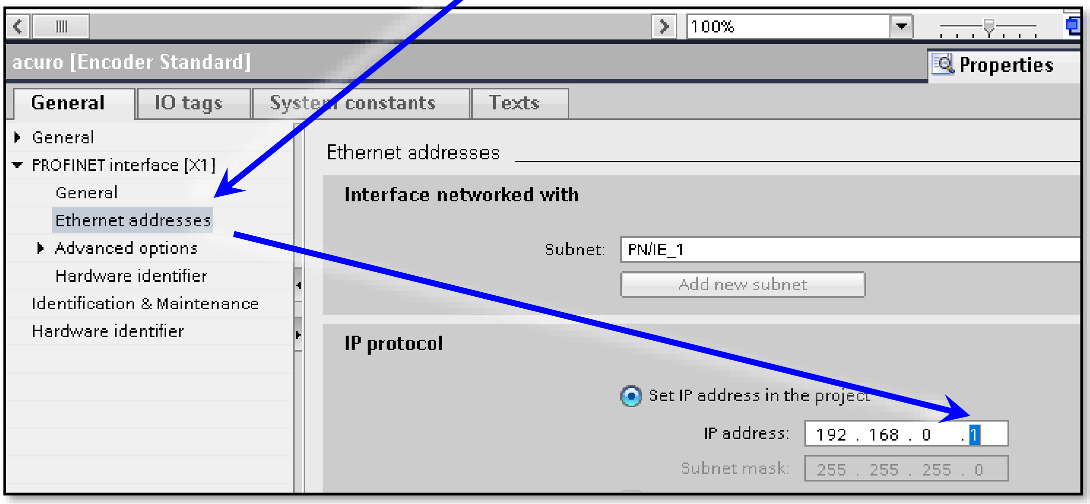


Figure 25: “General”-Tab in Device-view

⇨ Change Encoder-IP-Address to 192.168.0.**100** (range can be fit to your needs)
Usually Master gets low number, slaves high numbers:

Figure 26: “General”-Tab in Device-view – Dialog for IP-Address setting

7.1.1.6 Configure PLC

⇨ Switch back to tab "Network view" and double-click on S7-icon

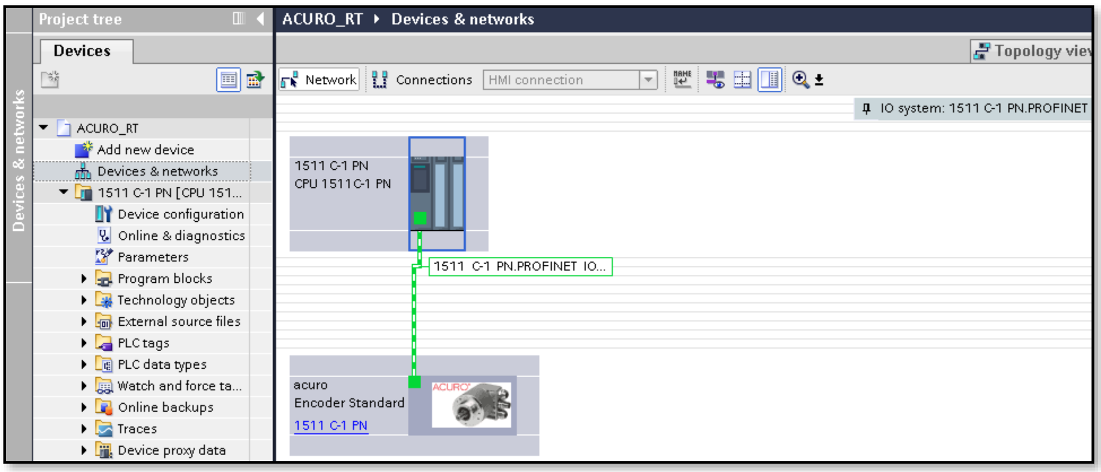


Figure 27: Network View showing Encoder assigned to Master (S7CPU)

Configure IP-address of S7-CPU:

⇨ Select "Ethernet-addresses" in PROFINET interface within dialog-tab "General" from "Properties"-dialog

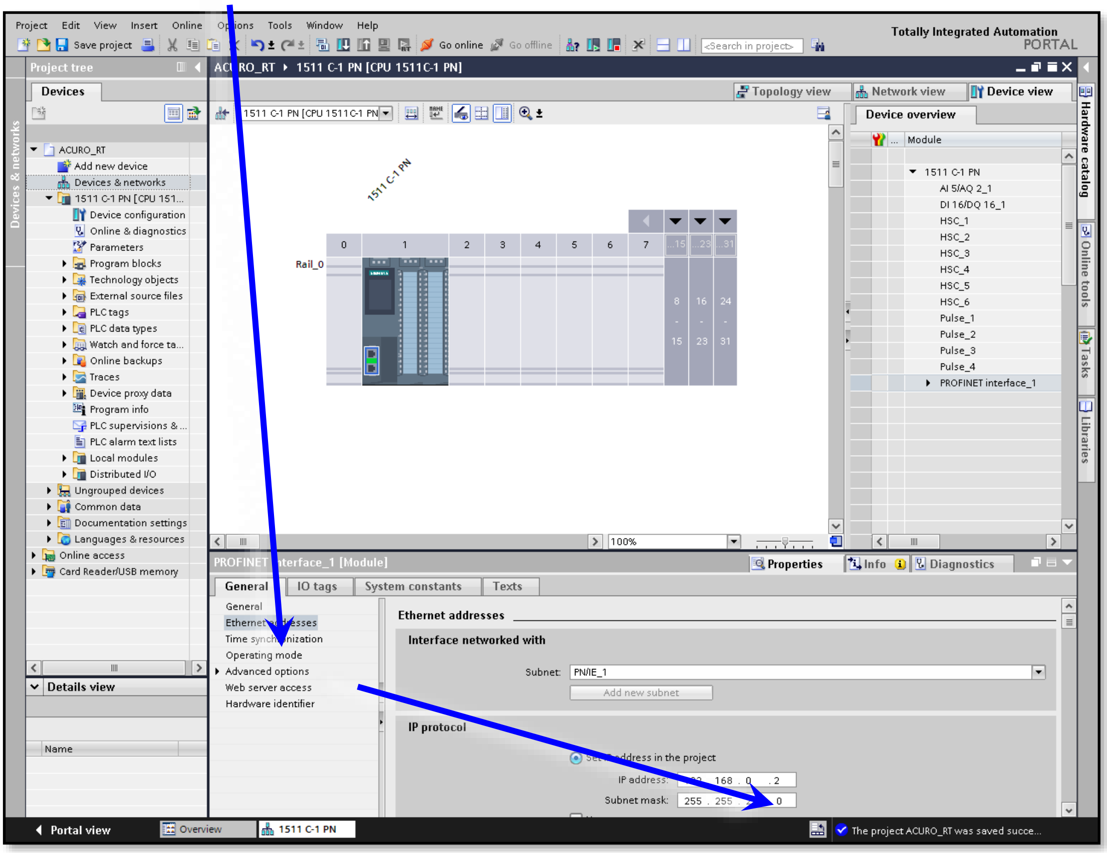


Figure 28: dialog-tab "General" from "Properties"-dialog

⇨ Set S7-CPU IP-address of S7-CPU to 192.168.0.**1**

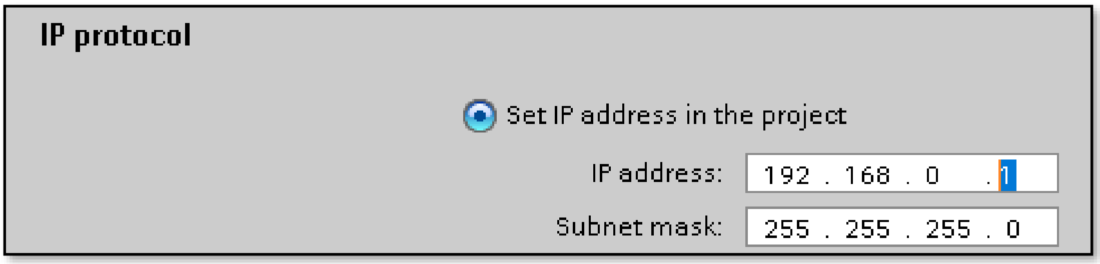


Figure 29: “General”-Tab in Device-view – Dialog for IP-Address setting

7.1.1.7 Load project onto PLC

⇨ Select menu "Online"-"Download to device"

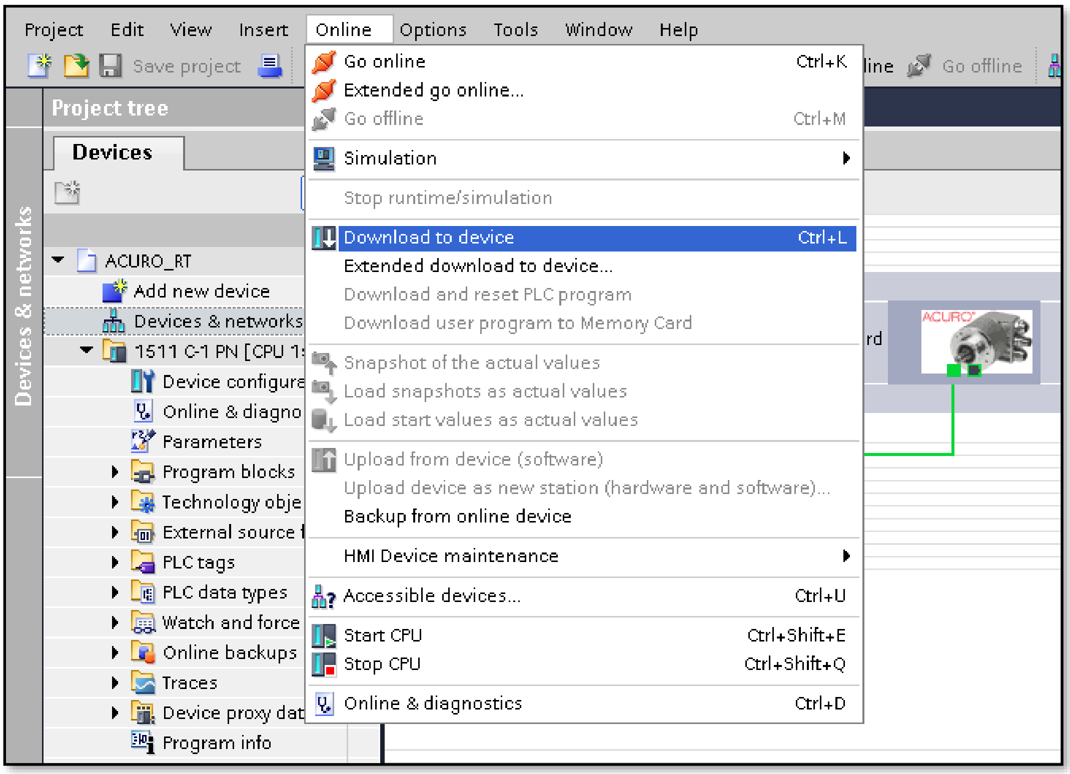


Figure 30: Dialog-window "Online"

⇨ Select network connection (typical "PN/IE") as well as PG/PC Interface (Your network card connected to S7 CPU), connection "Direct at slot '1 X1'"

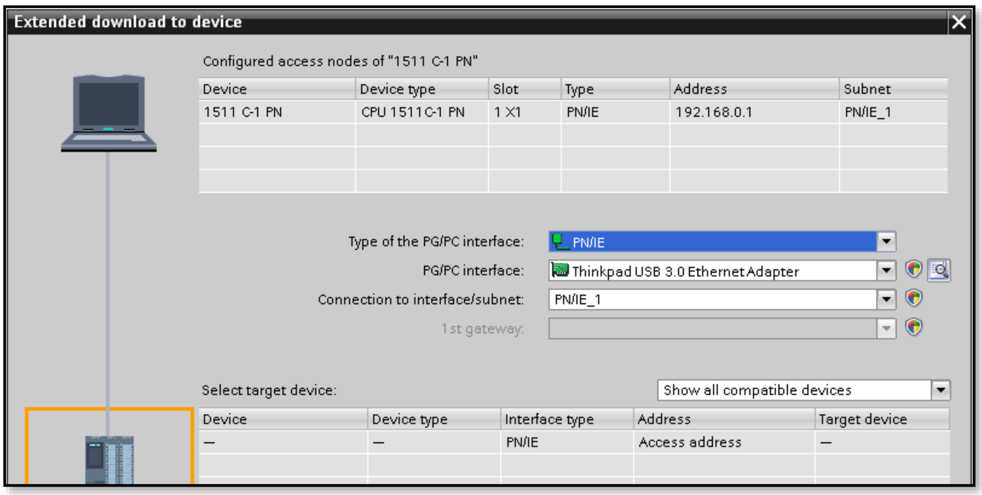


Figure 31: Dialog-window "Download to device"

⇨ Select target device by choosing “Show devices with same address”

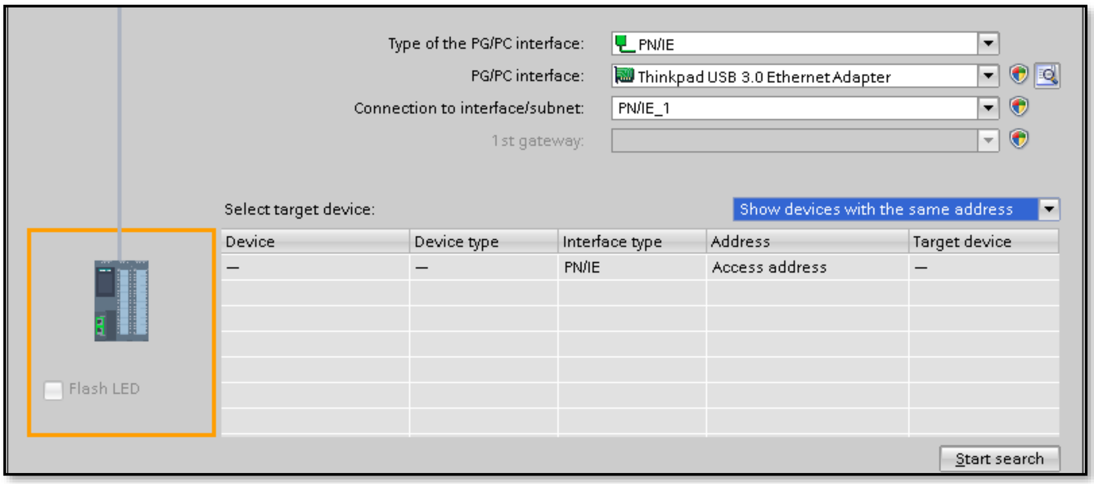


Figure 32: Dialog-window "Download to device"

⇨ when S7-CPU found, start download in selecting found S7-CPU and clicking "Load", if requested (and desired) store connection as default

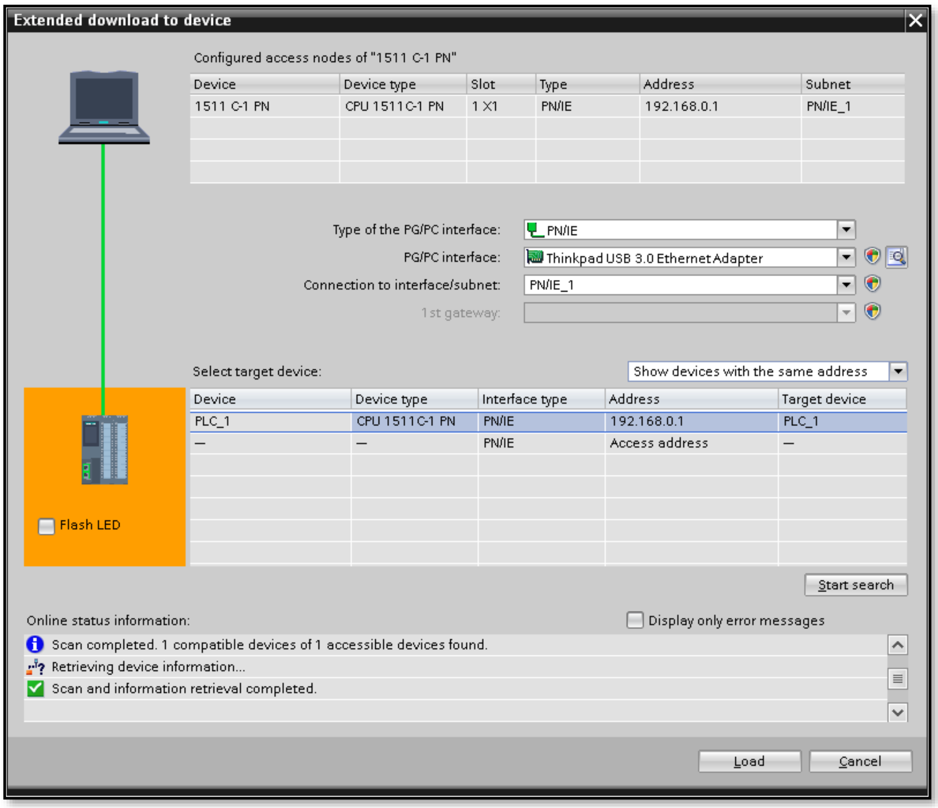


Figure 33: Dialog-window "Download to device"

⇨ Load preview:

⇨ Click checkbox "overwrite all", to overwrite all previous configurations

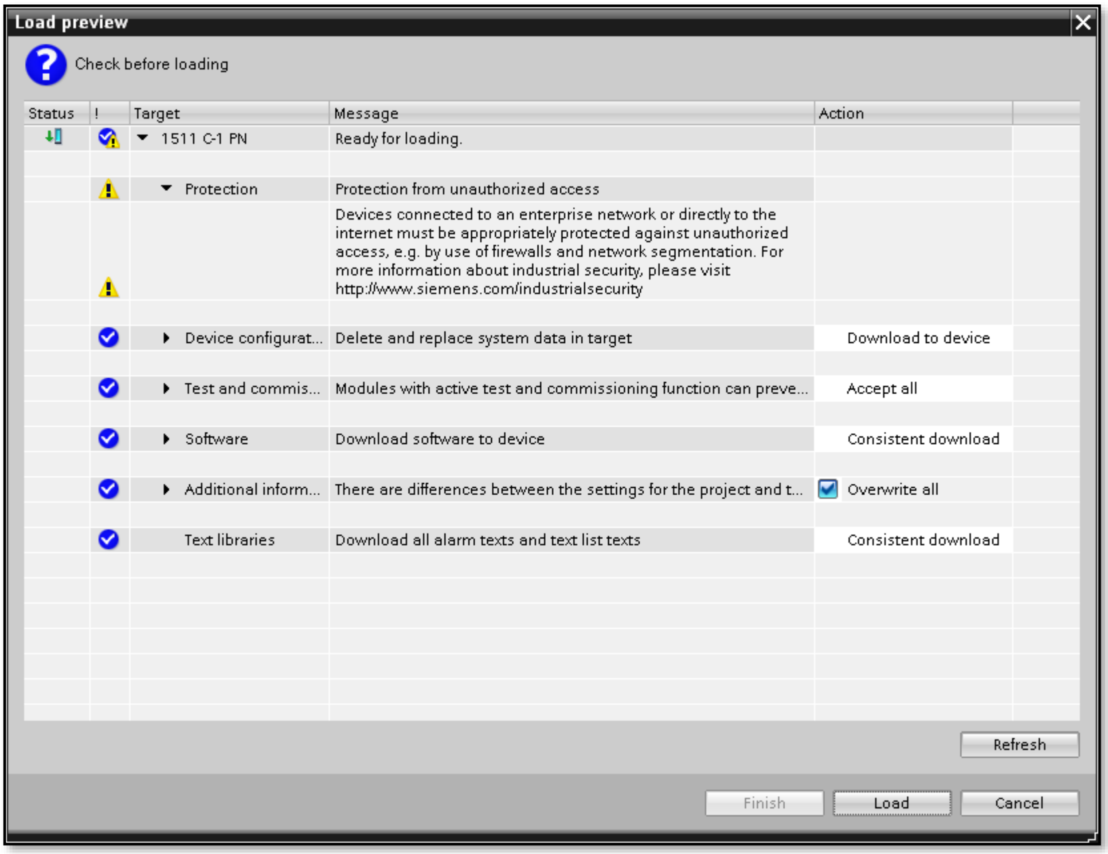


Figure 34: Dialog-window "Load preview"

⇨ Load results

⇨ Click checkbox for "Start all" and click on "Finish"

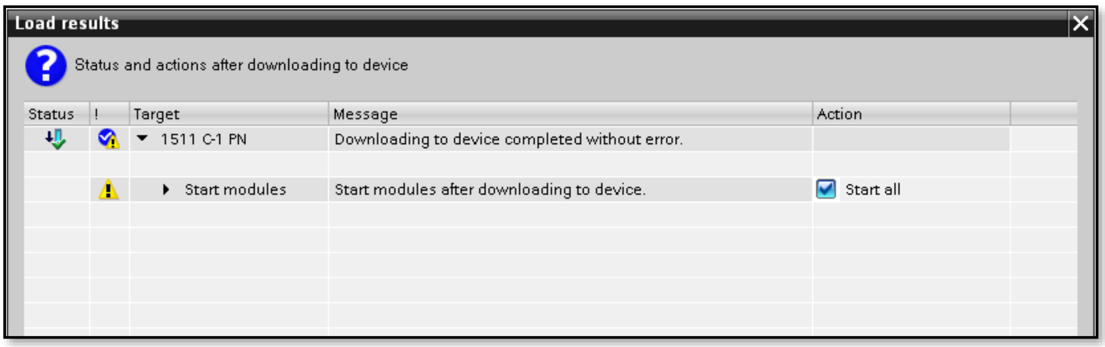


Figure 35: Dialog-window "Load results"

7.1.1.8 Create Online connection

⇨ click on "Go online"
and, if requested, configure your connection again like described above

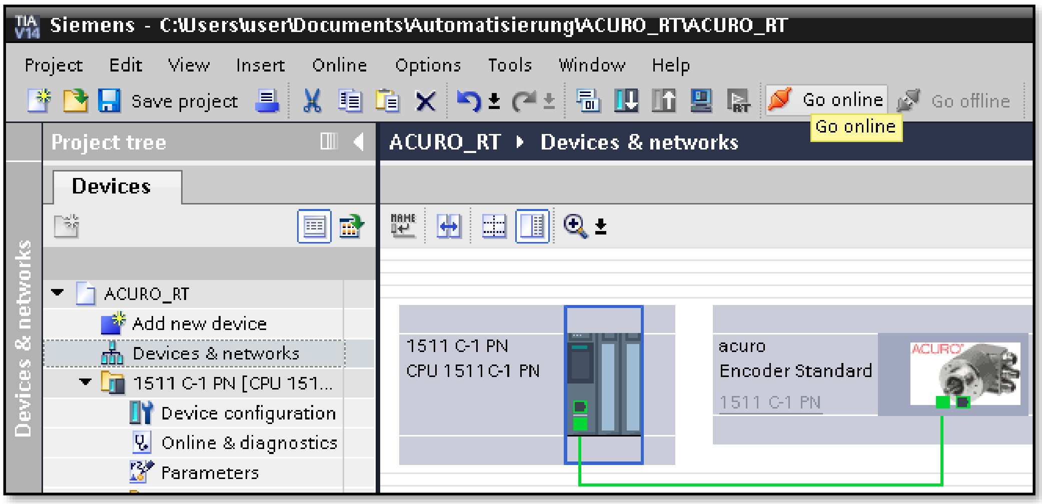


Figure 36: selected button “Go online”

⇨ Online connection established
⇨ Configuration successful, if all connections and devices are marked green

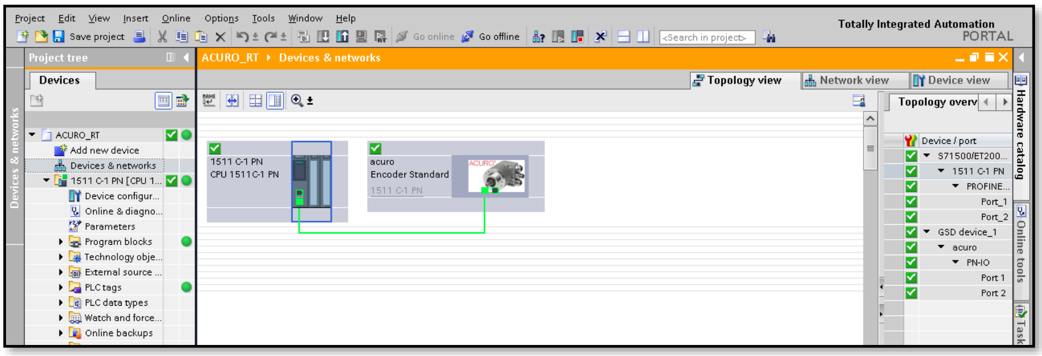


Figure 37: Main window

7.1.2 Configure Data-Format: Assign a Telegram

After Implementing the Encoder into a network, the data which would needed to be transmitted needs to be chosen.

Therefore, different Telegram styles are predefined (see Chapter 6.3 Standard Telegrams or 6.4 OEM Telegrams). Within these telegrams, it is defined, which data (Position, Speed, …) in which format would be submitted.

According the requirements of the application, one of the described telegrams would need to be selected. The following procedure describes, how to assign the chosen telegram within the project.

7.1.2.1 Configure Network

⇨ Open Network view after executing procedures described above (choose "Configure network" in Portal-view)

⇨ Double-Click ACURO-Symbol

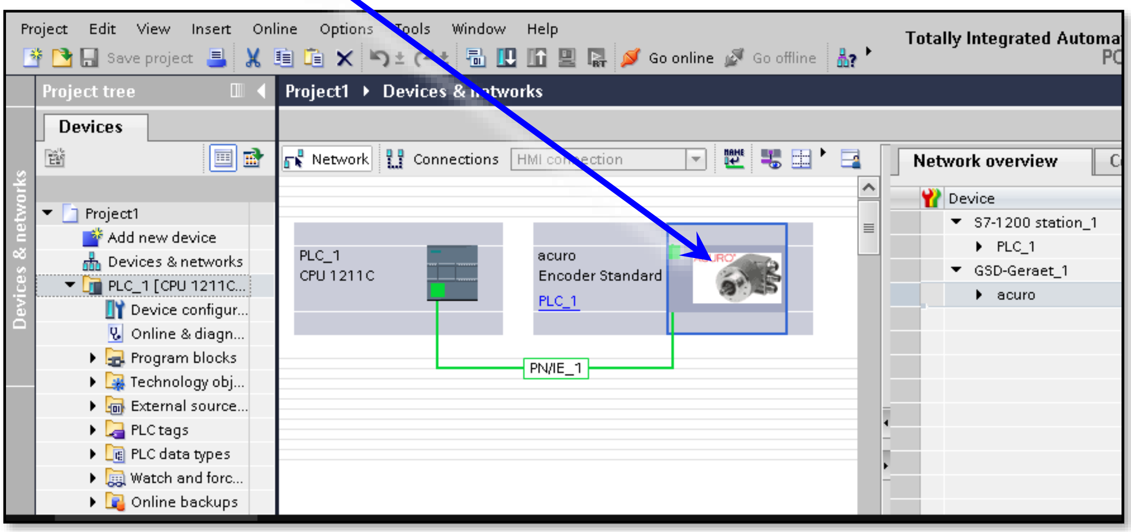


Figure 38: Network-View

⇨ Open up in Hardware Catalog – “Submodules”
⇨ Select the mating Telegram, i.e. here shown “Telegram 84”
⇨ Draw it into the device-view in empty line below “Modul Access Point”

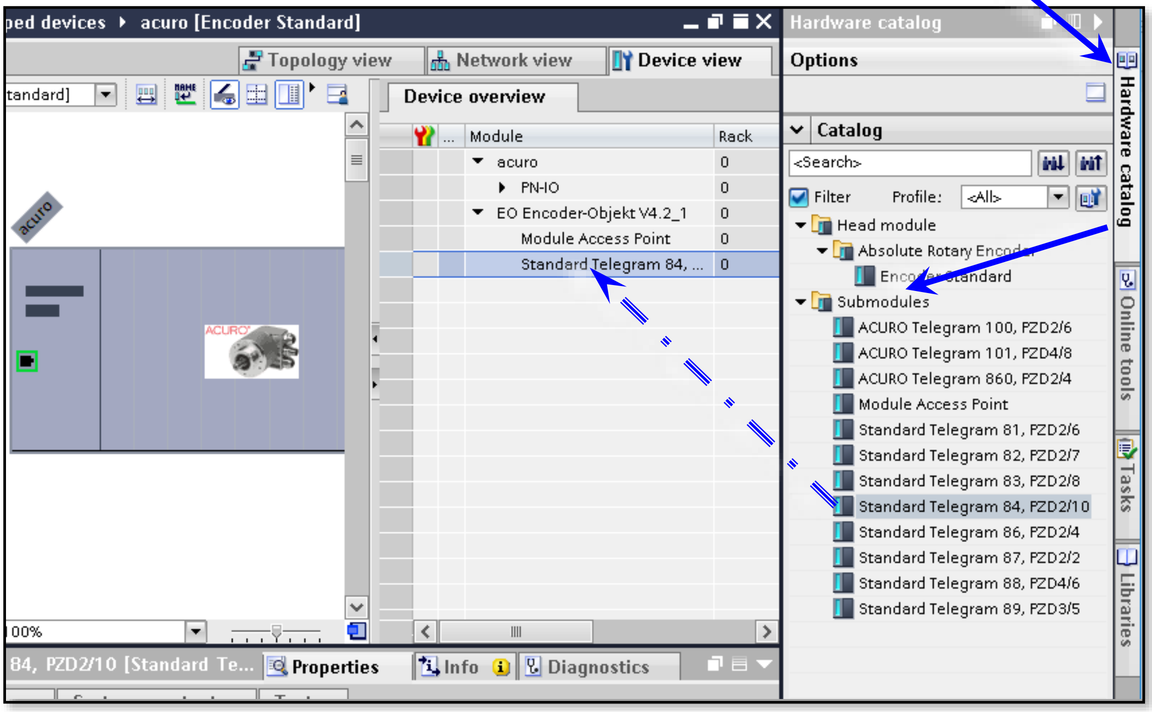


Figure 39: Device View - Hardware catalog

The addresses of chosen “Telegram 84” can be seen and modified in I/O addresses box
These addresses may be needed by user specific programming

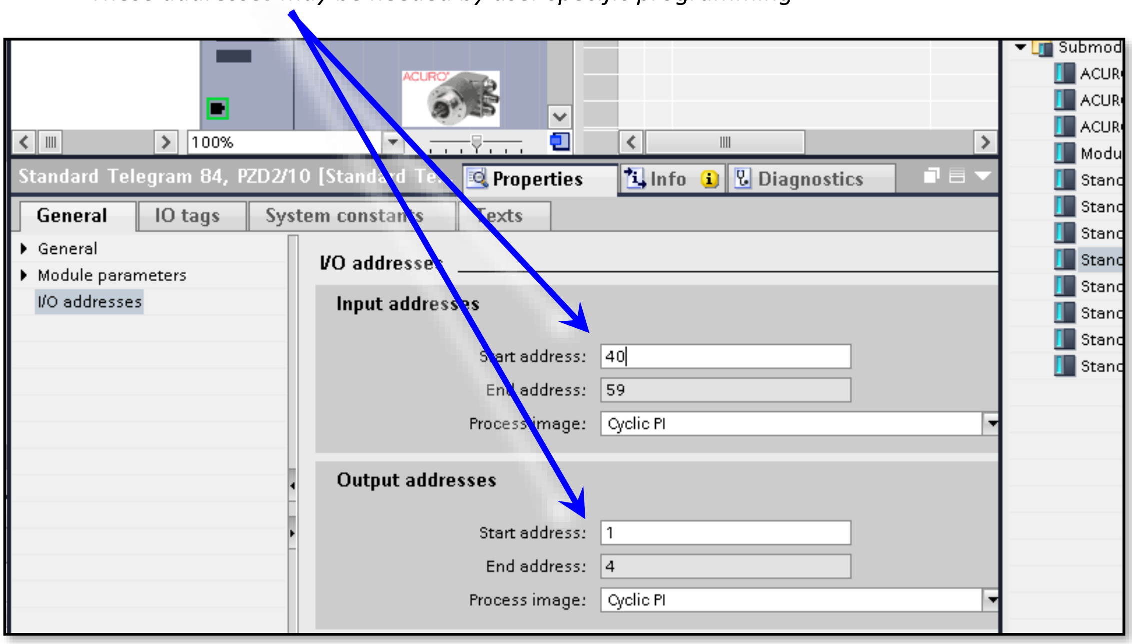


Figure 40: Network-View – Properties - I/O addresses

7.1.3 Configure ACURO Sensor Configuration

⇨ select "Module access point"

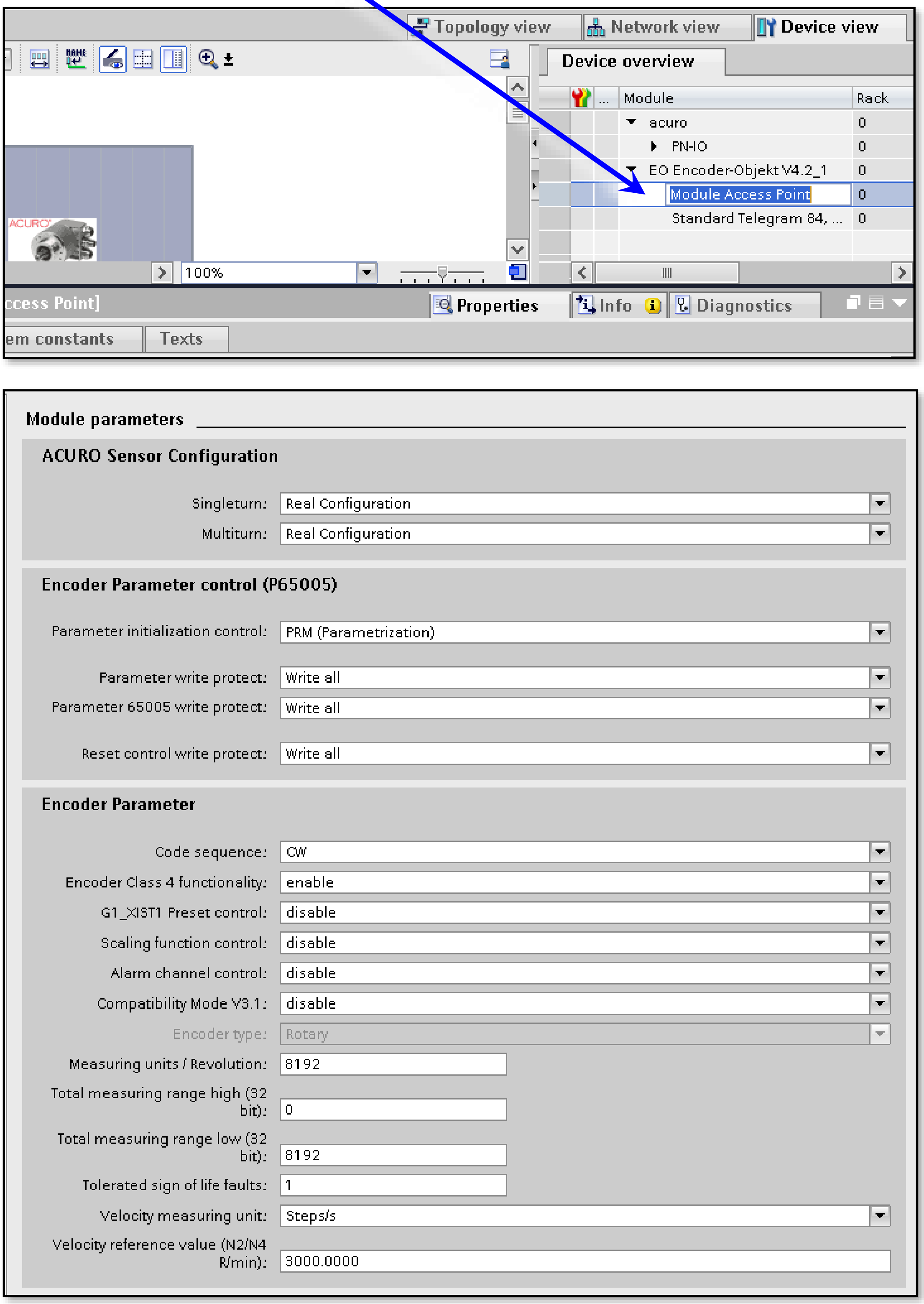


Figure 41: Network-View – Properties – Module parameters

7.1.4 Startup IRT-Mode

If Clock synchronous operation for Application is necessary, this can be realized with IRT Mode.
The following procedure describes, how to switch the RT-Installation into IRT (Isochronous Real Time) mode
Please notice, that your S7 version must offer this feature!

7.1.4.1 Configure connection to IRT

- ⇨ Open Network view after procedures described above
- ⇨ Select Network view segment "PN/IE_1"
- ⇨ Select Sync Domain

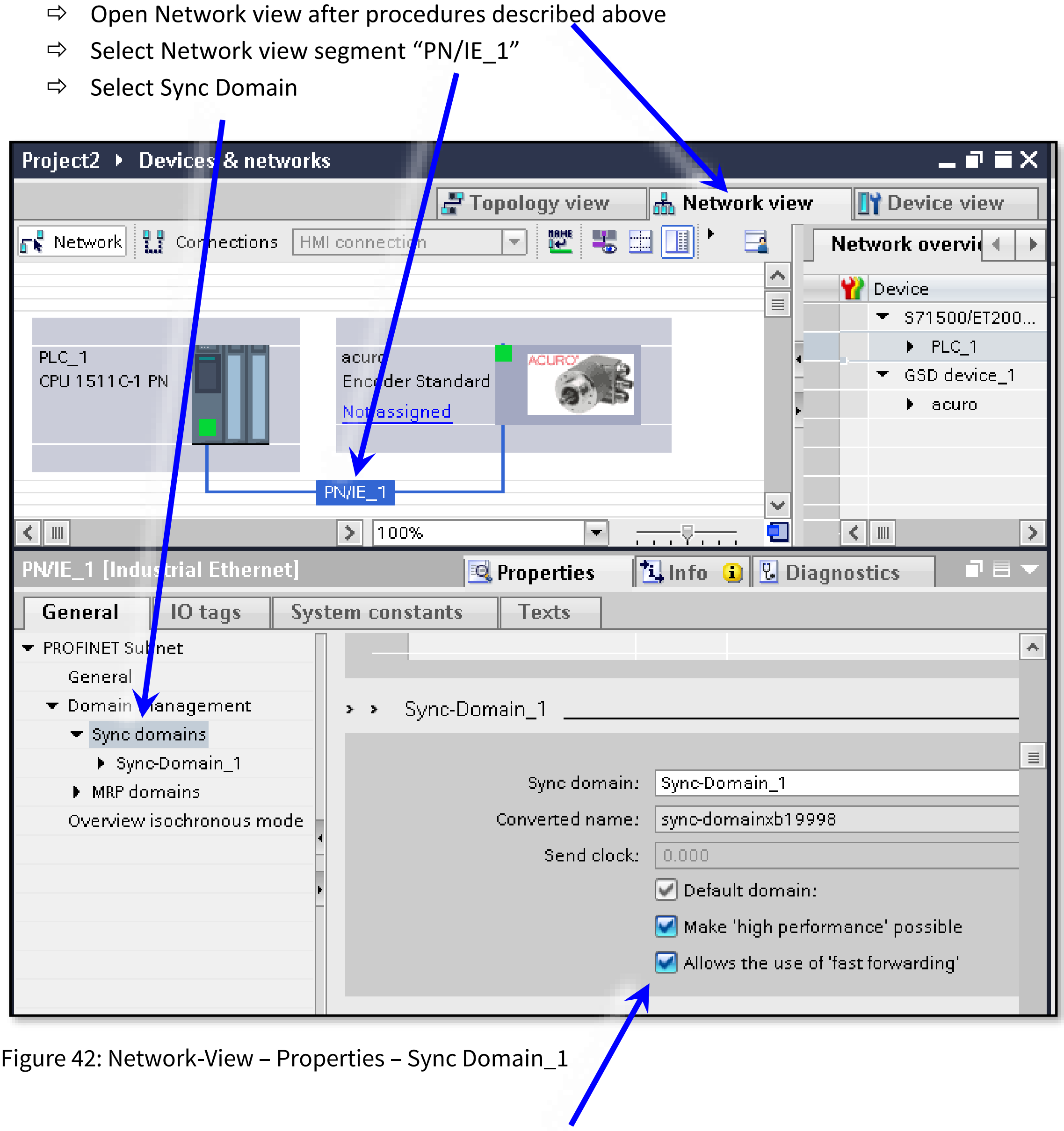


Figure 42: Network-View – Properties – Sync Domain_1

- ⇨ Activate "Make high performance possible" as well as "Allow the use of fast forwarding""
- ⇨ Select Sync Domain

7.1.4.2 Configure Devices to IRT

⇨ Select Device" Scroll down to "IO devices"

⇨ Change role of S7 to "Sync Master" and the role of Acuro to "Sync Slave"

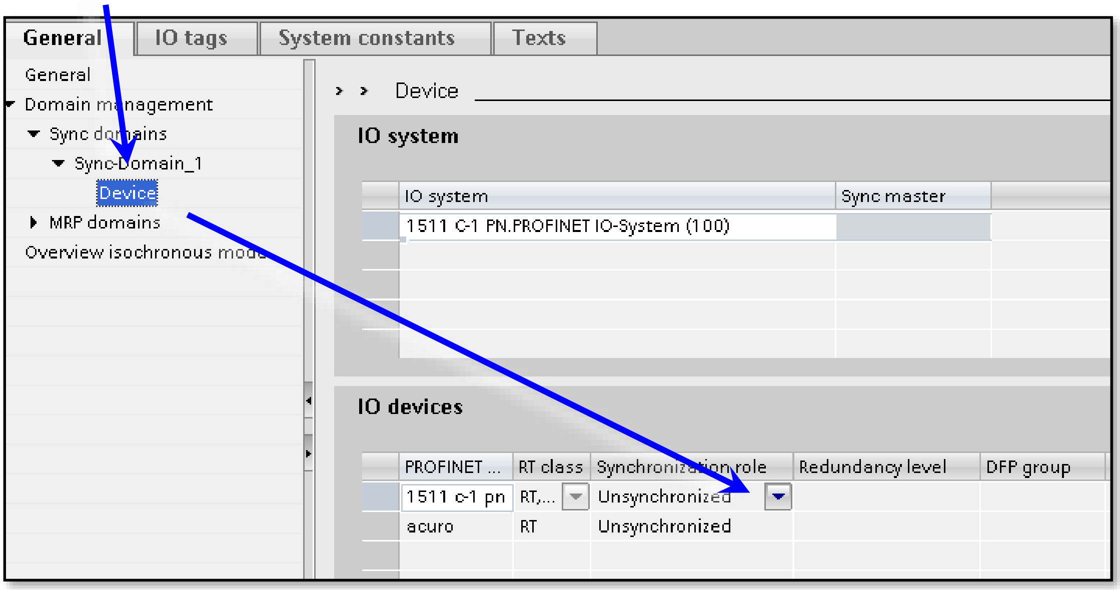


Figure 43: Network-View – Properties – IO devices

⇨ Now, both devices are in IRT-mode

7.1.4.3 Add IRT program block

⇨ Open Program blocks, double-click on "Add new block"

⇨ Open "Organization Block in opened dialog-box

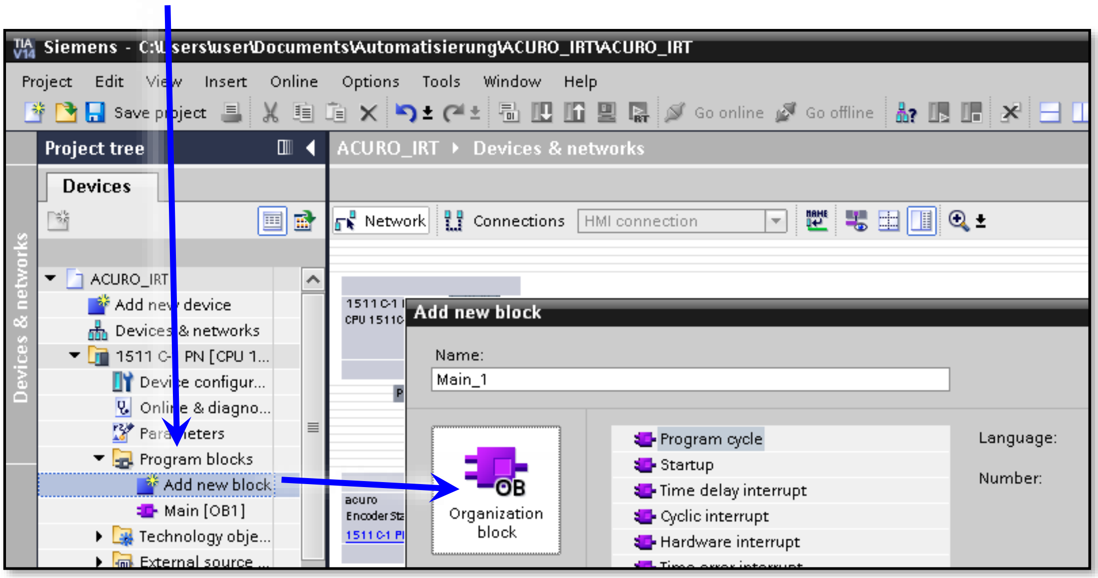


Figure 44: Project tree – Devices + Dialog-Box "Add new block"

⇨ Choose "Synchronous Cycle"-Block

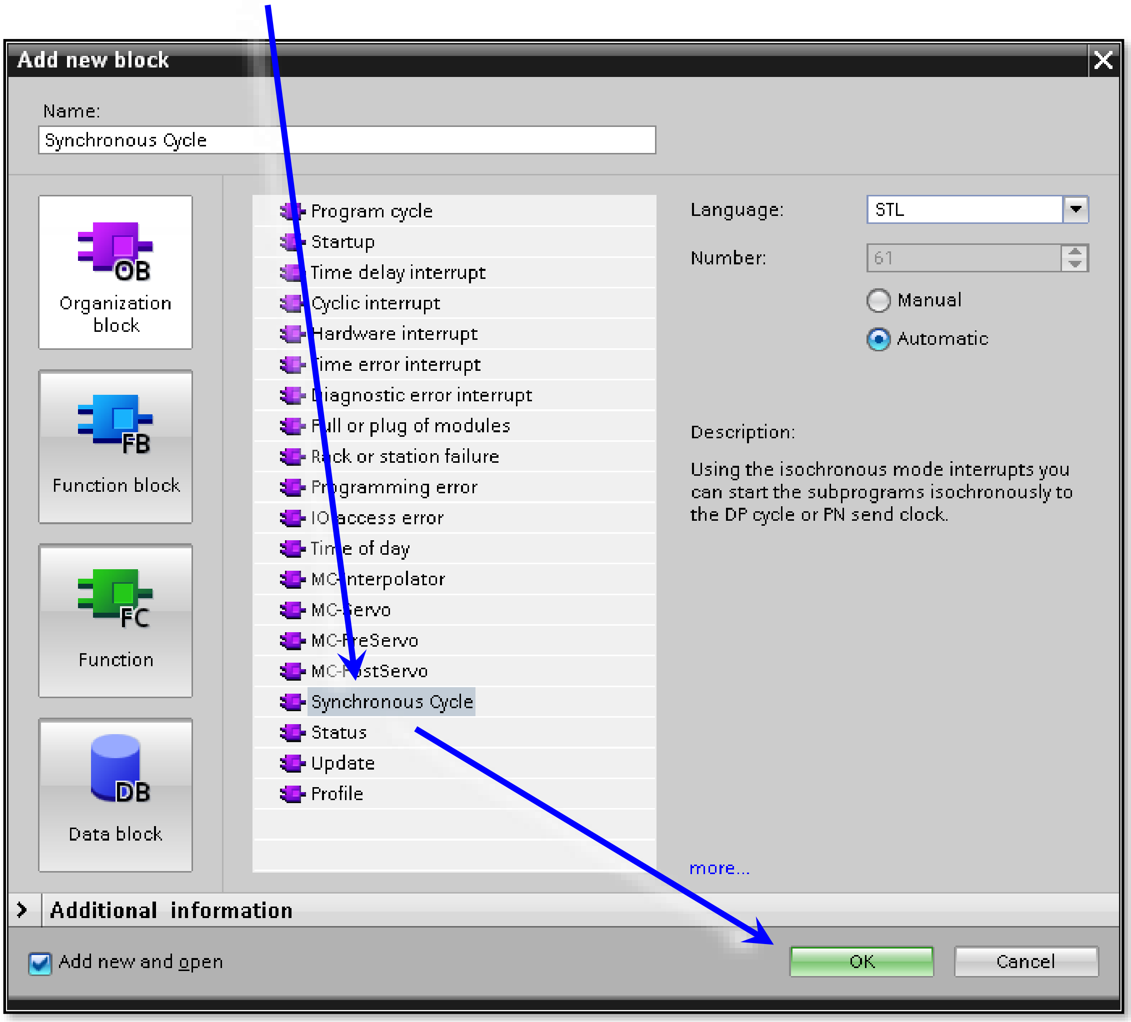


Figure 45: Dialog-Box "Add new block

⇨ Program-block inserted as STL-block

7.1.4.4 Configure port

⇨ Choose "Device configuration" in double-click on

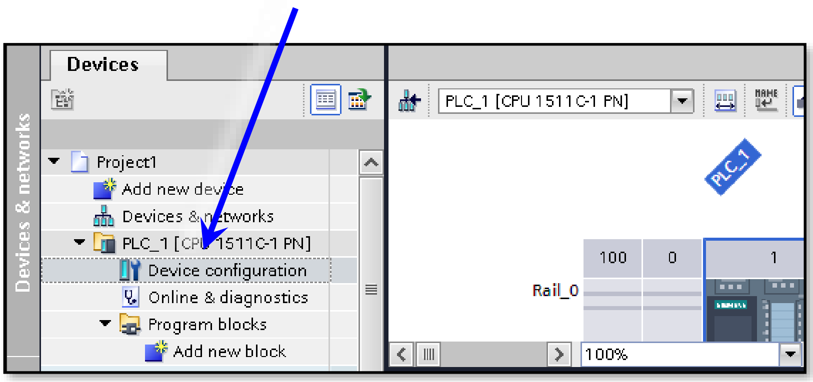


Figure 46: Project tree – Devices

⇨ Choose Port-X1 within "Advanced Options"

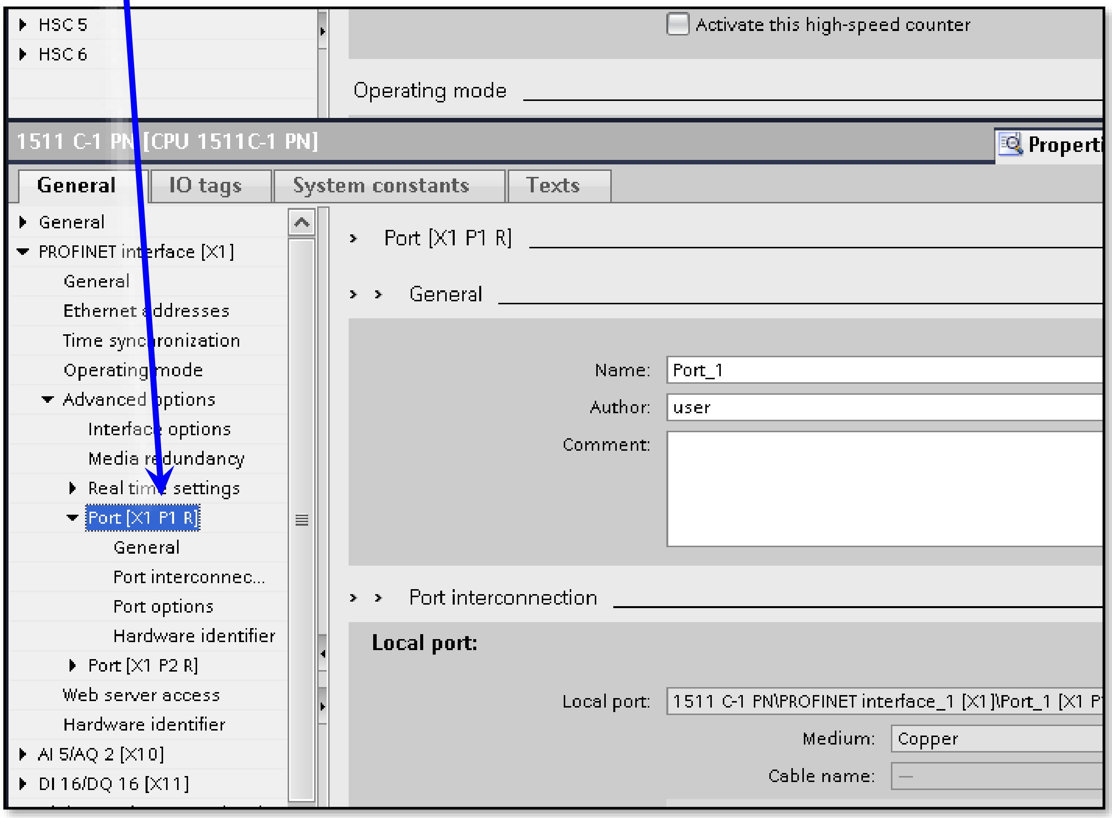


Figure 47: Device View – Advanced options - Local Port

⇨ If required, set Boundaries within Port Options

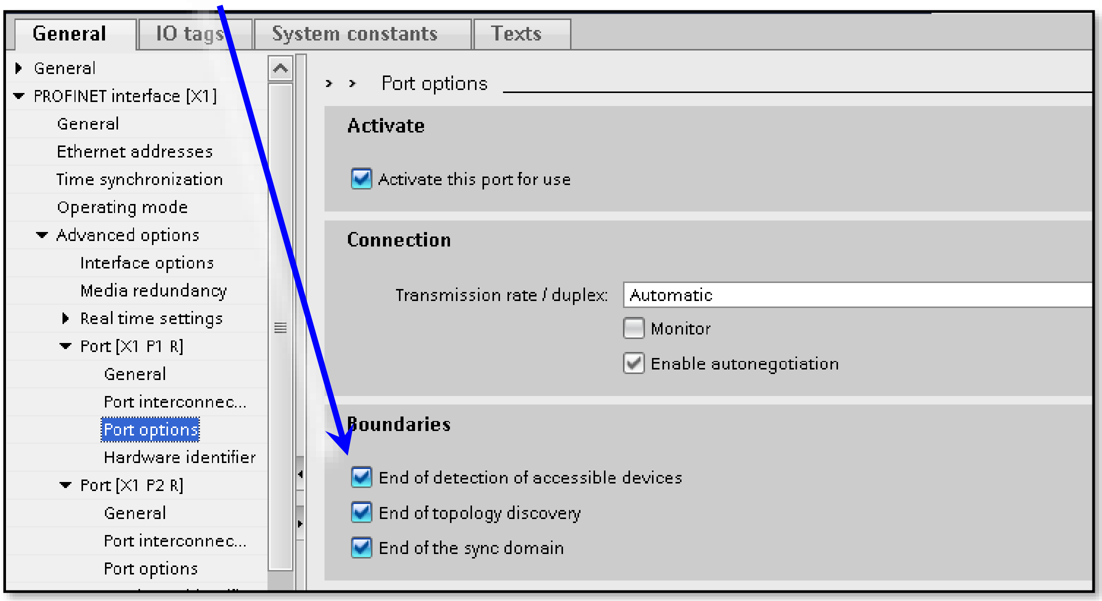


Figure 48: Device View – Port options – Boundaries

7.1.4.5 Verify Configuration

⇨ Verify, if S7 is assigned as Sync-Master

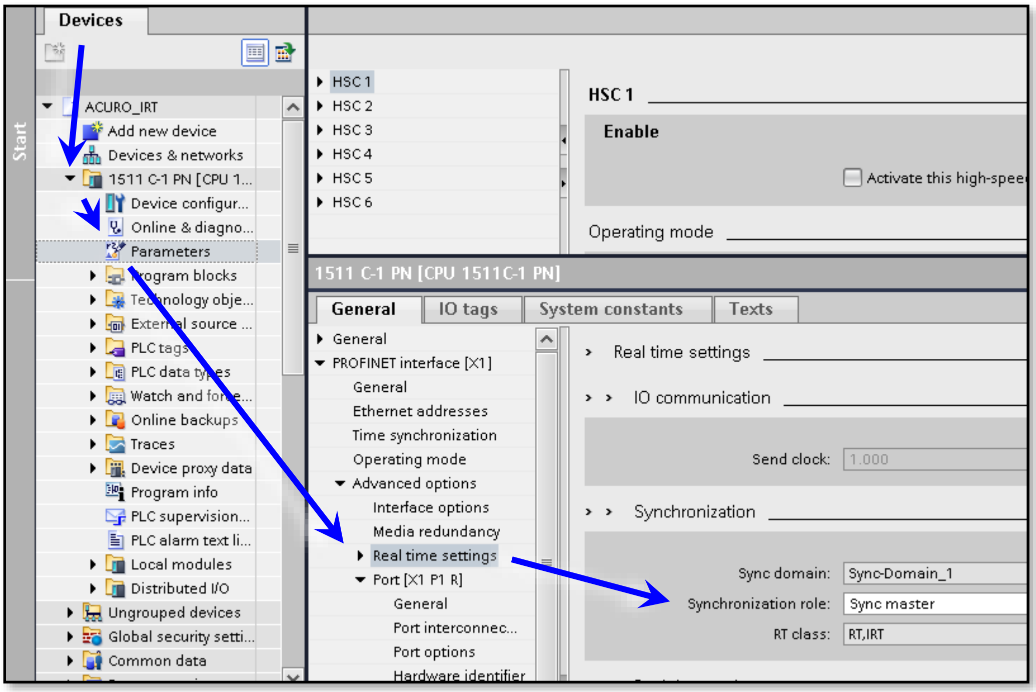


Figure 49: Device View – Parameters – Real time settings

7.1.4.6 Install Realtime setting for Encoder

⇨ Connect ACURO-Telegram with Synchronous-block “Synchronous Cycle”

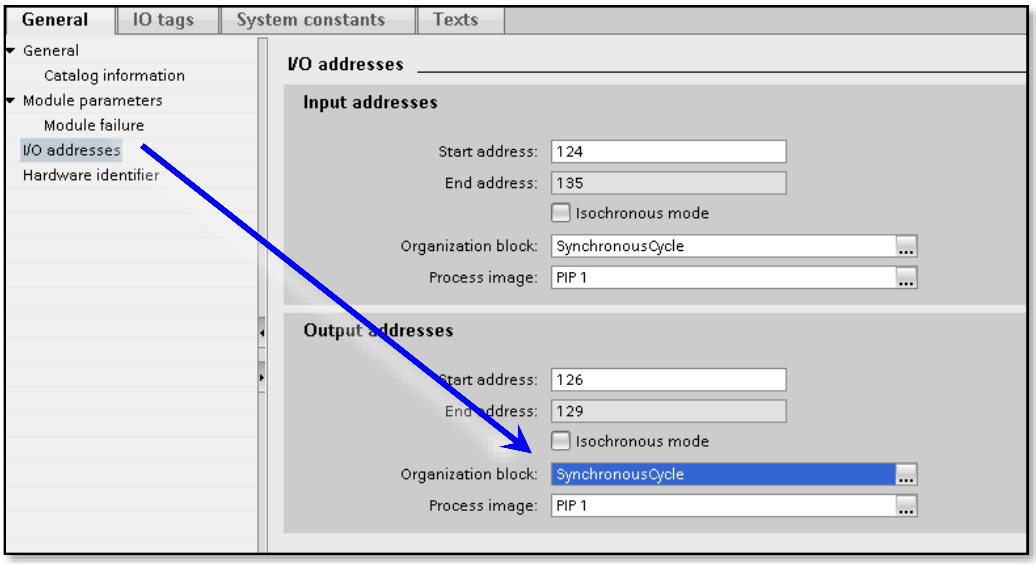


Figure 50: Device View – Module Parameters – Output addresses

⇨ Chose “Synchronous Cycle” in Dialog-Box

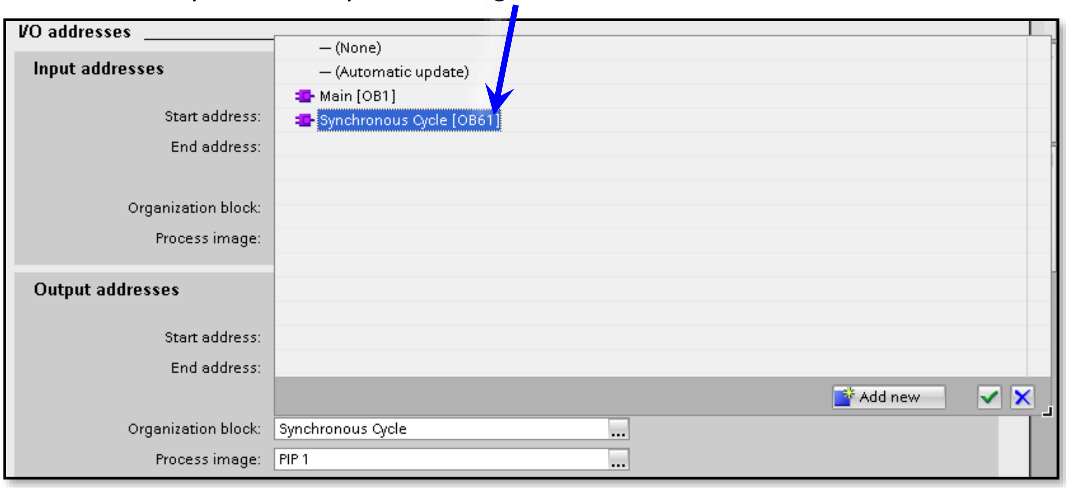


Figure 51: I/O addresses – organization block - dialog-box

⇨ activate synchronized actualization for Input and output

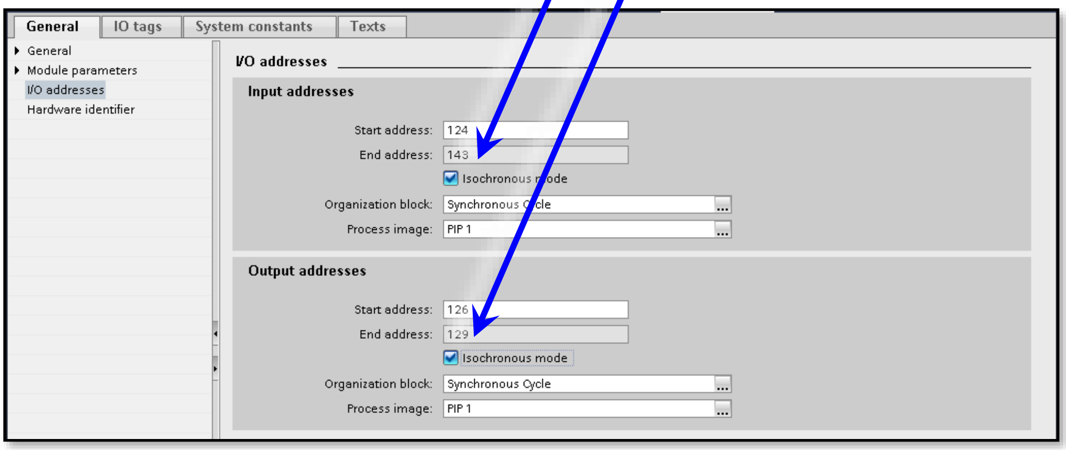


Figure 52: I/O addresses – addresses

⇨ Set both checkboxes “Isochronous Mode” for complete Slave

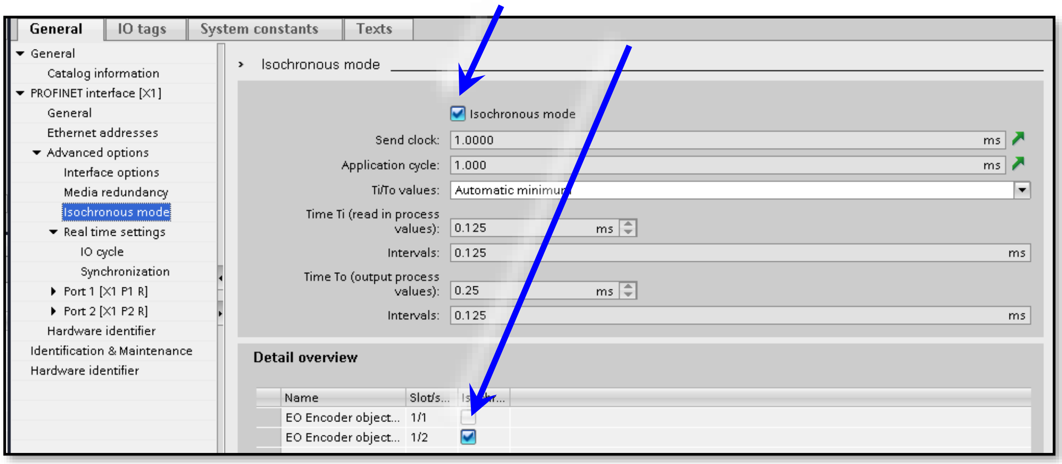


Figure 53: Advanced options – Isochronous mode

⇨ Verify, Realtime-settings: RT-class should be “IRT”

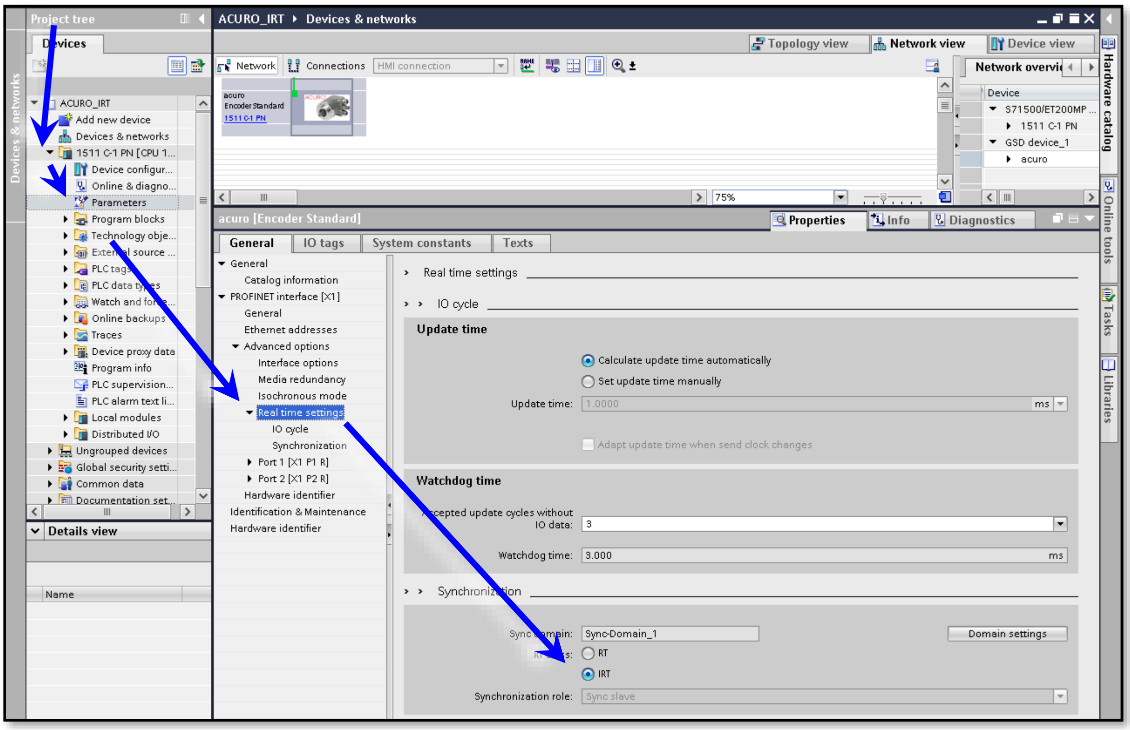


Figure 54: Advanced options – Real time settings - Synchronization

7.1.4.7 Start translation-procedure of hardware

⇨ Open menu by licking right mouse button”

⇨ Start translation procedure of hardware in selecting “Compile” => “hardware (rebuild all)”

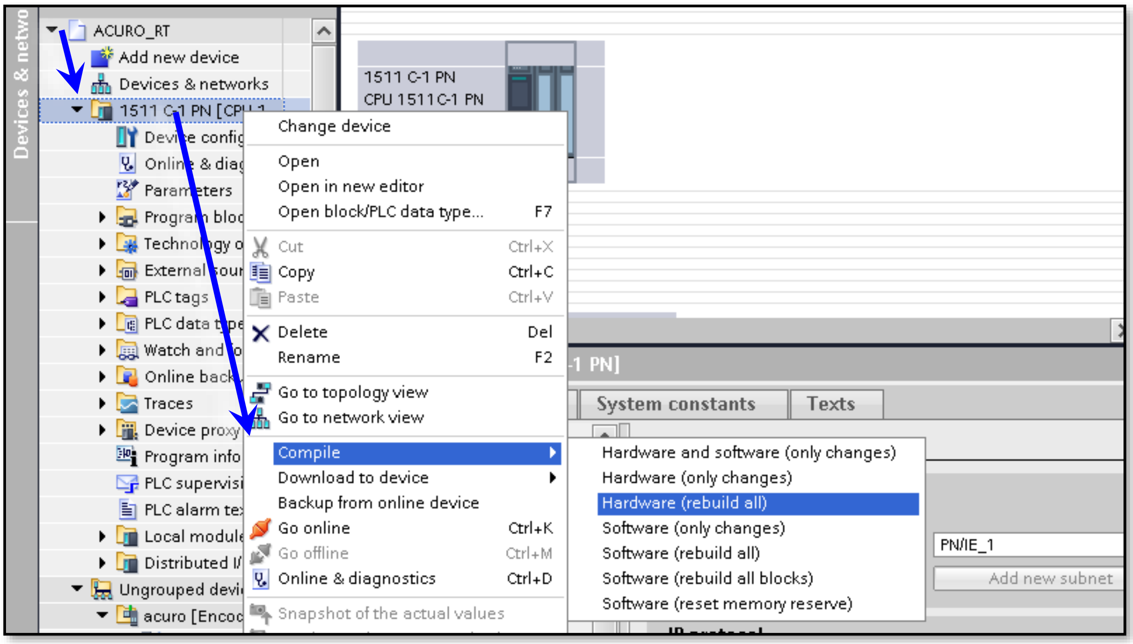


Figure 55: right mousebutton-menue

HENGSTLER

⇨ translation process done

⇨ Solve all warnings before getting productive!

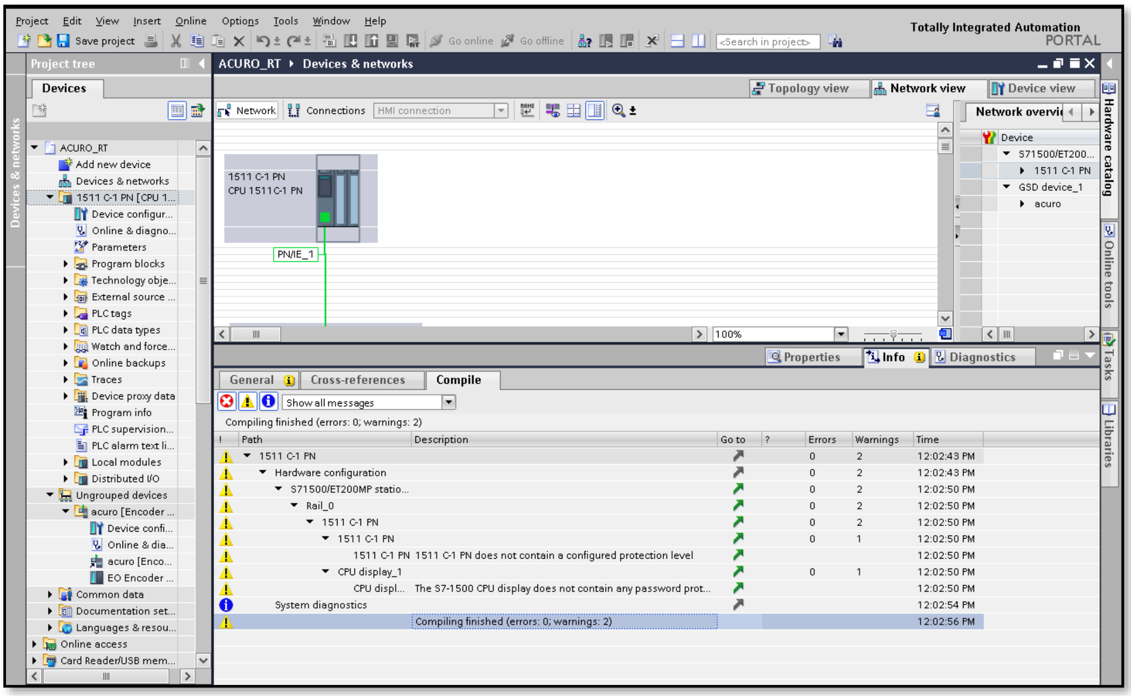


Figure 56: result view of translation process

⇨ Load project onto PLC (see procedure in former chapter)

8 Appendix A

8.1 Data Type Coding

8.1.1 PROFIdrive Data Type coding

Name	Code	Description
BOOL	0x01	Logical Boolean with values TRUE and FALSE
INT8	0x02	Signed 8–bit integer value
INT16	0x03	Signed 16–bit integer value
INT32	0x04	Signed 32–bit integer value
INT64	0x38	Signed 64–bit integer value
UINT8	0x05	Unsigned 8–bit integer value
UINT16	0x06	Unsigned 16–bit integer value
UINT32	0x07	Unsigned 32–bit integer value
UINT64	0x37	Unsigned 64–bit integer value
FLOAT32	0x08	32–bit floating point value
FLOAT64	0x0F	64–bit floating point value
VISIBILE-STRING	0x09	Visible String
OCTET-STRING	0x0A	Octet String
BINARY-DATE	0x0B	Binary Date
TIME_OF_DAY	0x0C	Time of day
TIME-DIFF	0x0D	Time Difference
OCTET-STRING2	0x1F	Octet String
ZERO	0x40	Zero Type
N2	0x71	Refer to PROFIdrive specification
N4	0x72	Refer to PROFIdrive specification
V2	0x73	Refer to PROFIdrive specification
L2	0x74	Refer to PROFIdrive specification
R2	0x75	Refer to PROFIdrive specification
T2	0x76	Refer to PROFIdrive specification
T4	0x77	Refer to PROFIdrive specification
D2	0x78	Refer to PROFIdrive specification
E2	0x79	Refer to PROFIdrive specification
C4	0x7A	Refer to PROFIdrive specification
X2	0x7B	Refer to PROFIdrive specification
X4	0x7C	Refer to PROFIdrive specification

8.1.2 CIP Data Type coding

Elementary data types are identified using the identification codes defined in the table below:

Name	Code	Description
BOOL	0xC1	Logical Boolean with values TRUE and FALSE
SINT	0xC2	Signed 8–bit integer value
INT	0xC3	Signed 16–bit integer value
DINT	0xC4	Signed 32–bit integer value
LINT	0xC5	Signed 64–bit integer value
USINT	0xC6	Unsigned 8–bit integer value
UINT	0xC7	Unsigned 16–bit integer value
UDINT	0xC8	Unsigned 32–bit integer value
ULINT	0xC9	Unsigned 64–bit integer value
REAL	0xCA	32–bit floating point value
LREAL	0xCB	64–bit floating point value
STIME	0xCC	Synchronous time information
DATE	0xCD	Date information
TIME_OF_DAY	0xCE	Time of day
DATE_AND_TIME	0xCF	Date and time of day
STRING	0xD0	character string (1 byte per character)
BYTE	0xD1	bit string 8bits
WORD	0xD2	bit string 16bits
DWORD	0xD3	bit string 32bits
LWORD	0xD4	bit string 64bits
STRING2	0xD5	character string (2 bytes per character)
FTIME	0xD6	Duration (high resolution)
LTIME	0xD7	Duration (long)
ITEM	0xD8	Duration (short)
STRINGN	0xD9	character string (N bytes per character)
SHORT_STRING	0xDA	character sting (1 byte per character, 1byte length indicator)
TIME	0xDB	Duration (milliseconds)
EPATH	0xDC	CIP path segments
ENGUNIT	0xDD	Engineering Units
STRINGI	0xDE	International Character String
STRUCT	0xA2	Struct data filed

Table 29: CIP Data Type Definitions

8.2 ISDK Data Types

8.2.1 SHORT STRING

8.2.1.1 TS_ISDK_STR32

Code	Name	Data type	Description	Data size
0xA2	TS_ISDK_STR32	STRUCT of	Device Information Parameters	33 = (1 + 32)
	u8Length	UINT8	Length of String	1
	str	ARRAY of CHAR	String Buffer	32: ISDK_STR32_LEN

8.2.1.2 TS_ISDK_STR256

Code	Name	Data type	Description	Data size
0xA2	TS_ISDK_STR256	STRUCT of	Device Information Parameters	256 = (1 + 255)
	u8Length	UINT8	Length of String	1
	str	ARRAY of CHAR	String Buffer	256: ISDK_STR256_LEN

8.2.2 TS_ISDK_SW_VER

Code	Name	Data type	Description	Data size
0xA2	TS_ISDK_SW_VER	STRUCT of	Firmware Information Parameters	4
	u8IC	UINT8	Internal Changes	BYTE-0
	u8BF	UINT8	Bug-Fix	BYTE-1
	u8FE	UINT8	Functional Enhancement	BYTE-2
	u8TR	UINT8	Type Recognition 'V'	BYTE-3

8.2.3 TS_ISDK_SPI_CFG

Code	Name	Data type	Description	Data size
0xA2	TS_ISDK_SPI_CFG	STRUCT of	SPI Config. parameters	TS_ISDK_SPI_CFG
	u8ChID	UINT8	Channel-ID	
	u8MsMode	UINT8	Master/Slave Mode	
	u8OutMode	UINT8	Master:0, Slave: output select	
	u8Format	UINT8	Frame Format	
	u8ClkPhaPol	UINT8	Clock / Phase / Polarity	
	u8DataSize	UINT8	Data size	
	u32Src	UINT8	Serial baud rate source	
	u32Div	UINT32	Serial baud rate divisor	
	u8IrqMask	UINT32	Interrupt handling	
	u8LoopBack	UINT8	Loopback mode	

8.2.4 TS_ISDK_DEVINFO

Code	Name	Data type	Description	Data size
0xDA	**tDevInfo**	**STRUCT of**	Device Information Parameters	**TS_ISDK_DEVINFO**
	u16VendorID	*UINT16*	*Vendor-ID*	
	u16DeviceID	*UINT16*	*Device-ID*	
	u16DevType	*UINT16*	*Device-Type*	
	u32ProdCode	*UINT32*	*Product-Code*	
	u32ArticleNr	*UINT32*	*Article-Number*	
	u32BoardType	*UINT32*	*Board-Type*	
	tName	*SHORT-STR*	*Product-Name*	*TS_ISDK_STR32*
	tType	*SHORT-STR*	*Product-Type*	*TS_ISDK_STR32*
	tFamily	*SHORT-STR*	*Product-Family*	*TS_ISDK_STR32*
	tSerial	*SHORT-STR*	*Serial-Number*	*TS_ISDK_STR32*
	tOrderID;	*SHORT-STR*	*Order-ID*	*TS_ISDK_STR32*
	tVendorName	*SHORT-STR*	*Vendor-Name*	*TS_ISDK_STR32*
	abMacAddrList	*2D ARRAY of BYTE*	*MAC Address List*	***MAC[CNT][LEN]*** *CNT: 3,* *LEN: 6 Bytes*

Table 30: ISDK Data Types