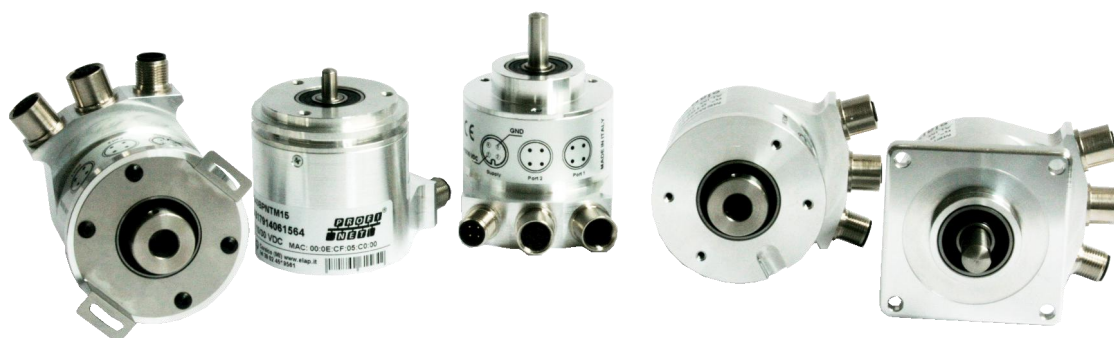


MEM-BUS ABSOLUTE ENCODER



With PROFINET Interface Operating Manual



ELAP VIA VITTORIO VENETO, 4 • I-20094 CORSICO (MI) • TEL. +39.02.4519561
FAX +39.02.45103406 • E-MAIL INFO@ELAP.IT • SITE WWW.ELAP.IT

CONTENTS

1	Generals.....	3	8	Acyclic Parameter Data.....	32
1.1	PROFINET Technology	3	8.1	Acyclic data exchange	32
1.2	The GSDML File.....	3	8.2	Identification and Maintenance (I&M functions)	32
1.3	The Encoder Profile	4	8.3	Base mode parameter access	32
1.4	MAC Address	4	8.4	Changing the preset value, parameter 65000.....	32
1.5	References	4	8.5	Reading the Preset value, parameter 65000	33
2	Installation	5	8.6	Supported parameters	33
2.1	Safety	5	9	Functional description of the encoder	36
2.2	Transport and storage.....	5	9.1	Code sequence	36
2.3	Mechanical assembly.....	5	9.2	Class 4 functionality	36
2.4	Electrical supply	5	9.3	G1_XIST1 Preset control	36
2.5	Connectors	6	9.4	Scaling function control	37
2.6	Status LEDs	7	9.5	Alarm channel control	37
3	Configuration with TIA PORTAL V13.....	8	9.6	Compatibility mode.....	37
3.1	Device description file installation (GSDML).....	8	9.7	Preset value	37
3.2	Setting the encoder configuration	9	9.8	Scaling function parameters.....	38
3.3	Setting the encoder device name.....	11	9.9	Maximum Master Sign-of-Life failures	38
3.4	Setting the encoder parameters.....	12	9.10	Velocity measuring units	38
3.5	Isochronous Real Time Setting (IRT).....	13	9.11	Encoder profile version	39
4	Configuration with STEP7.....	16	9.12	Offset value	39
4.1	Device description file installation (GSDML).....	16	9.13	Acyclic Data.....	39
4.2	Setting the encoder configuration	17	9.14	Identification and Maintenance (I&M Function).....	41
4.3	Setting the encoder device name.....	18	10	Encoder replacement using LLDP	42
4.4	Setting the encoder parameters.....	19	10.1	Encoder replacement with TIA PORTAL V13 ...	42
4.5	Isochronous Real Time Setting (IRT).....	20	10.2	Encoder replacement with STEP7	43
5	PROFINET IO data description	22	APPENDIX A: Reading position with TIA PORTAL		
5.1	Application Class definition	22	V13	45	
5.2	Standard signals	22	APPENDIX B: Reading / Writing parameters in		
5.3	Telegrams	22	record data block 0xB02E.....	46	
5.4	Format of Position Values in G1_XIST1 and		B.1	Writing Preset value (P65000).....	48
	G1_XIST2	23	B.2	Writing parameters P9 and P80.....	50
5.5	Format of the Position Value in G1_XIST3	24	APPENDIX C: Using Telegram 81 in IRT mode.....	52	
5.6	Control Word 2 (STW2_ENC)	24	APPENDIX D: Reading / Writing configuration		
5.7	Status Word 2 (SZW2_ENC)	24	record 0xBF00.....	54	
5.8	Control Word G1_STW	25	APPENDIX E: Operation with Technology Objects	56	
5.9	Status Word G1_SZW.....	25	TECHNICAL SPECIFICATIONS.....	57	
5.10	Preset function	25	ORDERING INFORMATION	57	
5.11	Real-Time communication	26			
6	IRT Communication and synchronization.....	28			
6.1	Controller Sign-of-Life (C-LS).....	28			
6.2	Device Sign-of-Life (DO-LS)	29			
6.3	Counting strategy for the Sign-of-Life failure				
	counter	30			
7	Alarms and warnings	31			
7.1	Diagnostics and Alarms	31			
7.2	Channel diagnostics.....	31			
7.3	Sensor status word	31			

DIMENSIONAL DRAWINGS ENCLOSED

1 Generals

The encoder is a precision measurement device used to determine angular positions and revolutions, and to provide them as electrical output signals to the follow-on device systems.

ELAP encoder has a resolution of 13 bit, corresponding to 8192 steps per revolution. Its integrated optical/magnetic sampling system makes it suitable for standard applications. All requirements for PROFINET IO devices are satisfied, both for RT (Real Time) and IRT (Isochronous Real Time) classes.

1.1 PROFINET Technology

PROFINET is the open industrial Ethernet standard of PROFIBUS & PROFINET International (PI) for automation. PROFINET uses TCP/IP and IT standards, and is in effect, real-time Ethernet.

There are two different net types relating to PROFINET: PROFINET CBA and PROFINET IO.

PROFINET CBA is suitable for the component-based communication via TCP/IP and the real-time communication for real-time requirements in modular systems engineering. Both communication options can be used in parallel.

PROFINET IO was developed for real time (RT) and isochronous real time (IRT) communication with the decentralized periphery. The designations RT and IRT essentially describe the real-time properties for the communication within PROFINET IO.

To achieve these functions, three different protocol levels are defined:

- TCP/IP for PROFINET CBA for the commissioning of plants with reaction times in the range of 100 ms
- RT (Real-Time) protocol for PROFINET CBA and PROFINET IO applications up to 1 ms cycle times
- IRT (Isochronous Real-Time) for PROFINET IO applications in drive systems with cycle times of less than 1 ms

Interfacing the peripheral devices, such as encoders, is implemented by PROFINET IO. Its basis is a cascading real-time concept. PROFINET IO defines the entire data exchange between controllers (devices with "master functionality") and the devices (devices with "slave functionality"), as well as parameter setting and diagnosis.

PROFINET IO is designed for the fast data exchange between Ethernet-based field devices, and follows the provider-consumer model. The configuration of an IO-System has been kept nearly identical to the "look and feel" of PROFIBUS.

A PROFINET IO system consists of the following devices:

- The IO Controller, which contains the automation program and controls the automation task.
- The IO Device, which is a field device such as an encoder, monitored and controlled by an IO Controller.
- The IO Supervisor is software typically based on a PC, for setting parameters and diagnosing individual IO Devices.

An application relation (AR) is established between an IO Controller and an IO Device. These ARs are used to define communication relations (CR) with different characteristics for the transfer of parameters, cyclic exchange of data and handling of alarms.

1.2 The GSDML File

In order to start using an absolute encoder with PROFINET interface, a device description file needs to be downloaded and imported into the configuration software. The device description file is called a **Generic Station Description Markup Language** file and contains the necessary implementation parameters needed for a PROFINET IO device.

Process data and alarms are always transmitted in real time (RT). PROFINET IO Real-Time is based on IEEE and IEC definitions, so that all Real-Time Functions must be executed in a specific time lag during the Bus cycle. RT communication is the base for PROFINET IO data exchange and Real-Time data are always managed with a higher priority than TCP (UDP)/IP data.

1.3 The Encoder Profile

Profiles are pre-defined configurations of the functions and features available from PROFINET for use in specific devices or application such as encoders. They are specified by PI (PROFIBUS & PROFINET International) working groups and published by PI. Profiles are important for openness, interoperability and interchangeability, so that the end user can be sure that similar equipments from different vendors perform in a standardized way.

ELAP complies with the definitions in the encoder profile 3.162, version 4.1. PROFINET is generally defined by PROFIBUS & PROFINET International (PI) and backed by the INTERBUS Club and, since 2003, is part of the IEC 61158 and IEC 61784 standards.

1.4 MAC Address

PROFINET IO field devices are addressed using MAC addresses and IP addresses. All field devices have got three MAC addresses.

The first MAC address of the encoder is printed on its label for commissioning purposes.

1.5 References

- Profile Encoder For PROFIBUS and PROFINET, v. 4.1, n. 3.162
- Profile Drive Technology PROFIdrive, v. 4.1, n. 3.172
- Profile Guidelines Part 1: Identification & Maintenance Functions, v.2.0, n. 3.502
- Diagnosis for PROFINET IO, v.1.0, n.7.142
- PROFINET IRT Engineering, 1.3, n.7.172
- PROFINET Cabling and Interconnection Technology, v3.1, n.2.252
- PROFINET Installation Guideline for Cabling And Assembly, v1.0, n.8.072



Certification No. Z10989

2 Installation

- The encoder must be installed by qualified experts only, with net voltage off and standstill shaft.
- Always observe the operating instructions of the machine manufacturer.

2.1 Safety

- Always observe prevention and safety norms during the installation and operation of the device.
- Use the encoder exclusively for its intended purpose.
- High voltage, current and rotating parts may cause serious or fatal injuries.
- Encoders must not be operated outside the specified limited values (see detailed product documentation).

2.2 Transport and storage

- Always transport or store encoders in their original packaging.
- Never drop encoders or expose them to major vibrations.

2.3 Mechanical assembly

- Do not open the device.
- Do not carry out mechanical changes on the device.
- Avoid impacts or shocks on the housing and shaft.
- Operate the device within its environmental specifications.

2.4 Electrical supply

- Carry out the wiring operations exclusively with unplugged voltage supply.
- Do not operate on the electrical plant while the encoder is on.
- Ensure that the entire plant complies with EMC requirements. The installation environment and wiring affect the electromagnetic compatibility of the encoder. In particular:
 - before handling and installing the encoder, eliminate any electrostatic charge from your body and from any tool that will get in contact with the device;
 - supply the encoder with steady voltage free from electrical noise; if necessary, install EMC filters for the supply input;
 - always use shielded and, if possible, twisted cables
 - do not use longer cables than necessary
 - the device cable path should be away from power cables
 - install the device away from possible interference sources, or shield it effectively
 - connect the cable shield and the connector case to a protective earth and make sure that the protective earth is free from electrical noise; the connection to earth can be carried out at the encoder side and/or at the user side; it is up to the user to evaluate which is the best solution to keep the electrical interference as low as possible.
- To achieve the highest possible noise immunity the PROFINET cable screen must be connected to ground on both ends.
- In certain cases current might flow over the screen, therefore it is recommended to use equipotential connections.

2.5 Connectors

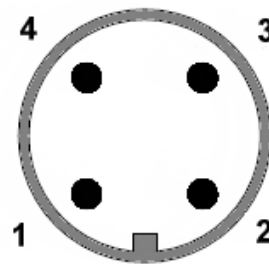
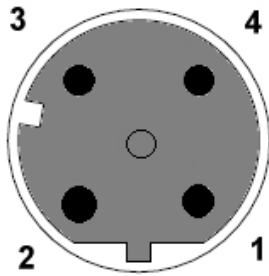
PROFINET connectors of the encoder are M12 female type for communication ports, and M12 male type for supply.

PROFINET connectors port 1 and port 2 M12 female type, D code

Pin	Signal	Colour
1	Tx +	Yellow
2	Rx +	Orange
3	Tx -	White
4	Rx -	Blue

SUPPLY CONNECTOR M12 male type, A code

Pin	Signal
1	Supply voltage (10-30 V DC)
2	N.C.
3	GND (0V)
4	N.C.



If PLC PROFINET connectors are RJ45 type, compatible to IEC 60603-7, the contact arrangement and the color coding of the cable is specified as follows:

Signal	M12 pin	Color	RJ45 pin
Tx+	1	Yellow	1
Rx+	2	White	3
Tx-	3	Orange	2
Rx-	4	Blue	6

For further information, see the document

PROFINET Cabling and Interconnection Technology, v3.1, n.2.252

which is available at address

<http://www.profibus.com/nc/download/installation-guide/downloads/profinet-cabling-and-interconnection-technology/display/>

2.6 Status LEDs



READY (green)

- OFF: Device is not ready to work.
- Flashing: Device is initializing its own data.
- ON: Device is ready to work.

ERROR (red)

- ON: Device is not connected to the BUS.
- Flashing: Device is connected to the BUS, but no communication with the IO Controller is present.
- OFF: Communication with the IO Controller is active.

LINK1 / LINK2 (green or yellow)

- OFF: Port 1 / 2 is not connected to the BUS.
- Green ON: Port 1 / 2 is connected to the BUS, but there is no data exchange.
- Yellow ON: Data exchange is active on Port 1 / 2.

LEDs LINK1 and LINK2 are also used as indicators by the “Search Device/ Flashing” function.

3 Configuration with TIA PORTAL V13

This chapter illustrates how to setup and configure a PROFINET encoder to work in RT and IRT modes, using TIA PORTAL V13 programming tool.

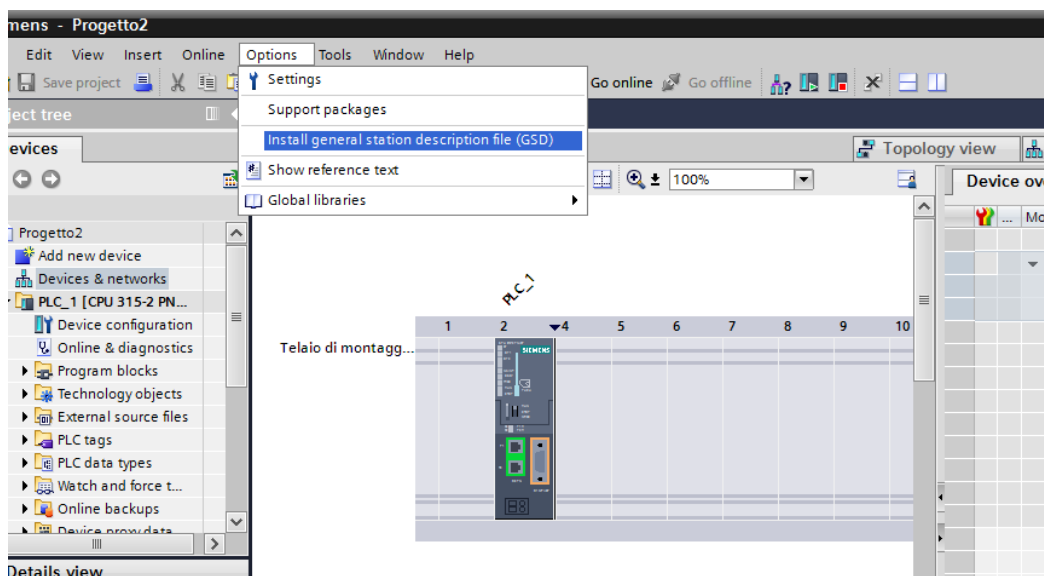
The following examples were generated with:

- Siemens TIA PORTAL V13 programming tool
- Siemens CPU315-2PN/DP 315-2EH14-0AB0 V3.2
- PROFINET ELAP Encoder MEM540BPNTM10

These examples are about a project in which the main CPU was already inserted. More details about the programming tool can be found in the examples provided directly by Siemens, at link: <http://support.automation.siemens.com/WW/view/it/40263542/0/it>

3.1 Device description file installation (GSDML)

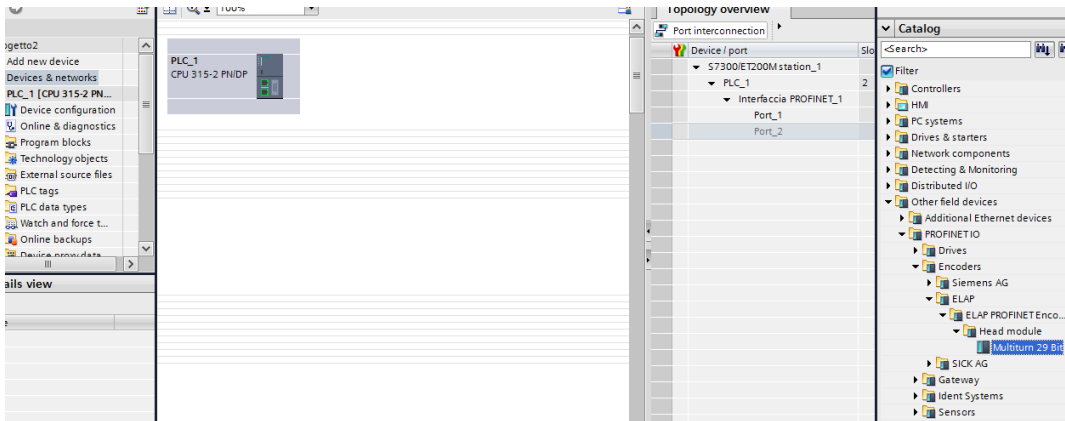
In order to start using an absolute encoder with PROFINET interface, a device description file needs to be imported into the configuration software. The device description file is a Generic Station Description Markup Language file, which contains the necessary implementation parameters needed for a PROFINET IO device.



1. Select *Options* → *Install GSD File* and click the *Browse button* to navigate to the location of the GSD file. If a bitmap picture representing the encoder is requested, make sure that the bitmap file is located in the same folder as the GSDML file. A bitmap file is provided with the GSDML file by ELAP.
2. Select the GSD file and click the *Install button* to start installing the file selected.

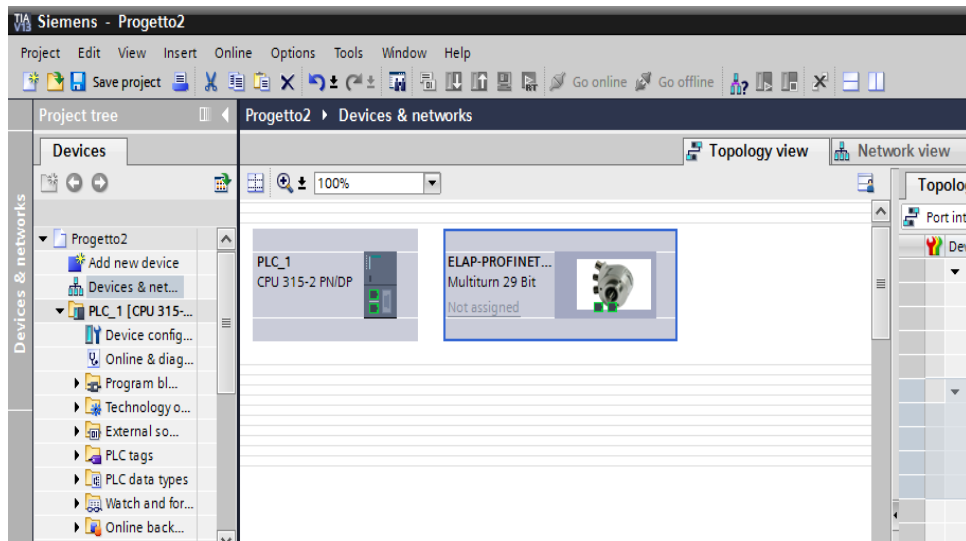
3.2 Setting the encoder configuration

When the GSD file has been installed, browse to *network view*: the supported encoder types can be found in the HW Configuration under *Other Field devices* → *PROFINET IO* → *Encoders* → *ELAP* → *ELAP PROFINET Encoders* → *Head Module* → *Multiturn 29 bit*

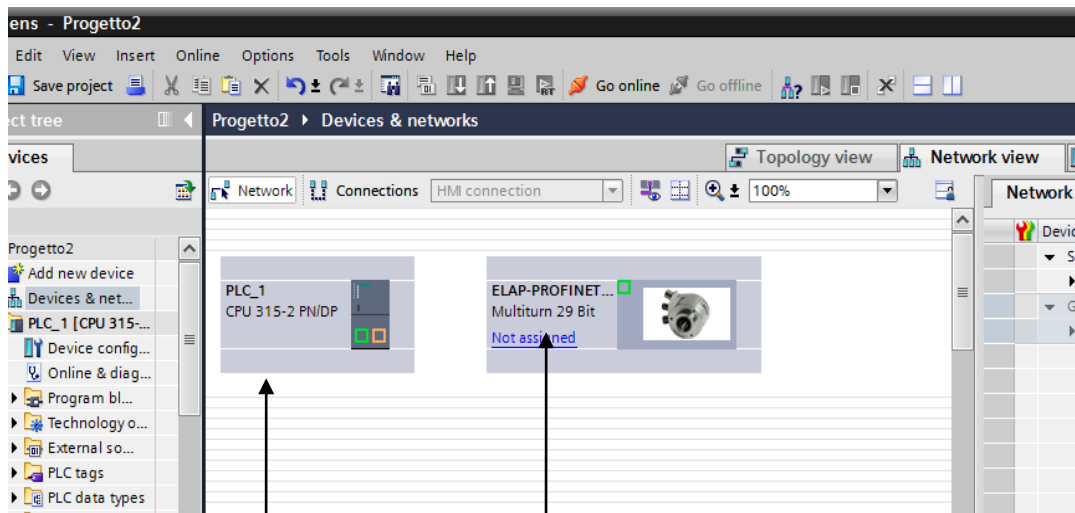


Select the encoder and double click it.

If different encoders must be configured, then the following steps need to be repeated for each device.

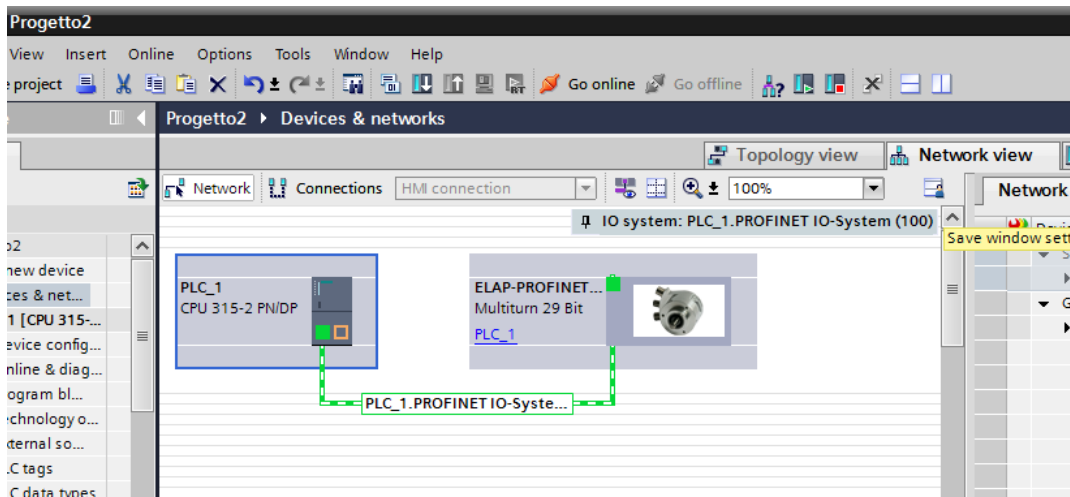


Now the device (encoder) must be assigned to the controller. Browse to *Network View*, and



Controller Interface Encoder Interface

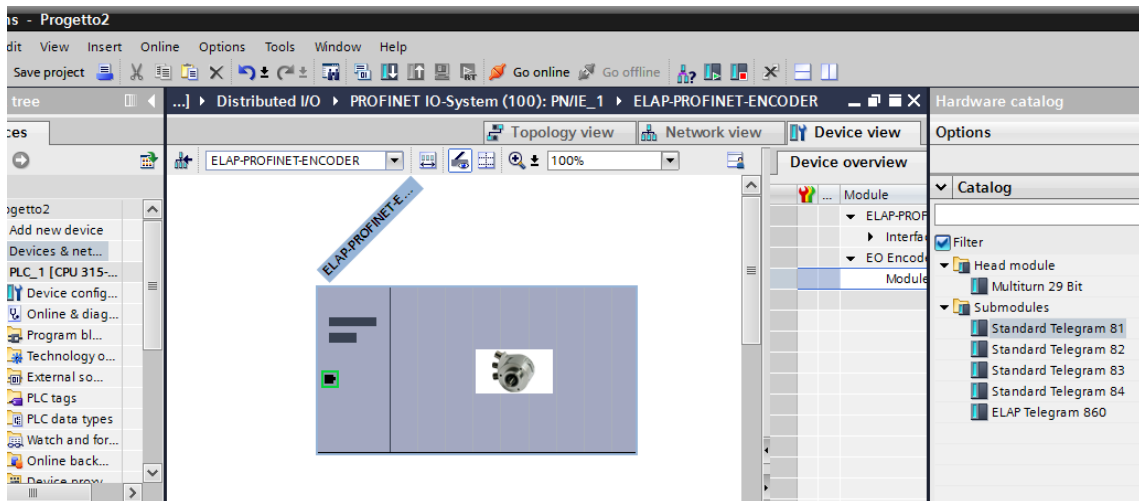
click on *Encoder Interface*; keep the left button clicked while dragging the mouse to the controller interface, then release the button (see previous image).



Now, the encoder is assigned to the IO controller.

The next step is to choose the type and length of data that should be sent to and from the IO controller. This is done by choosing one of different telegrams. The available telegrams for the Multi-turn 29 bit encoder can be found under *ELAP PROFINET Encoders* → *Submodules*

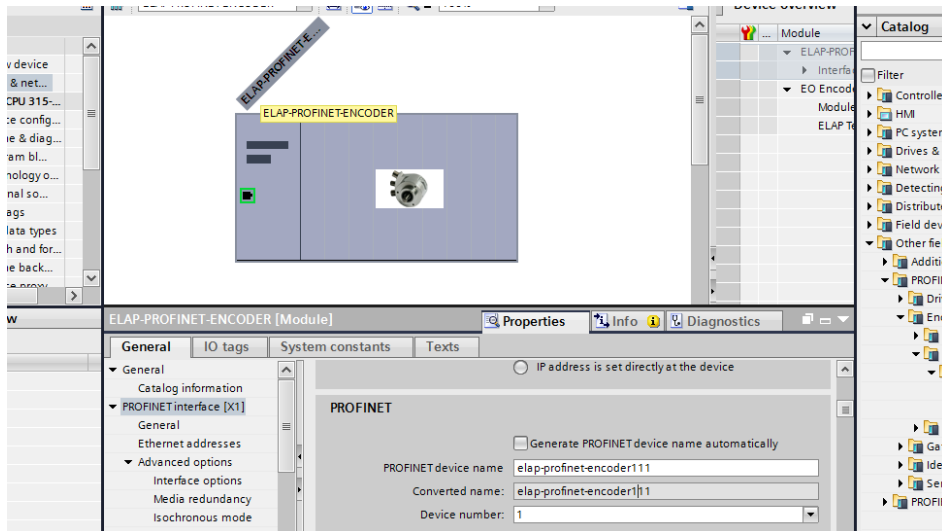
In this example, the standard telegram 81 is chosen. From the *Device View*, double click on the *Standard Telegram 81*.



NOTE: The steps above are to be performed for each device.

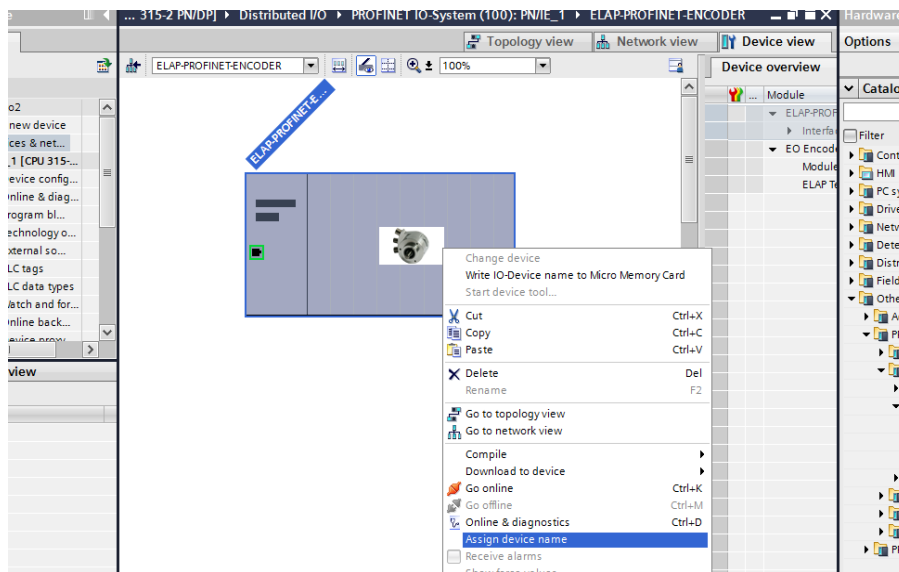
3.3 Setting the encoder device name

In a PROFINET network, all IO devices have got a unique device name. To set the encoder device name, double click on the encoder icon to open the *Properties* window. The default name for the encoder appears in the “Device name” field. Enter an appropriate device name (for example “ELAP-PROFINET-111”) and then disable the option “Generate PROFINET Device Name Automatically”.



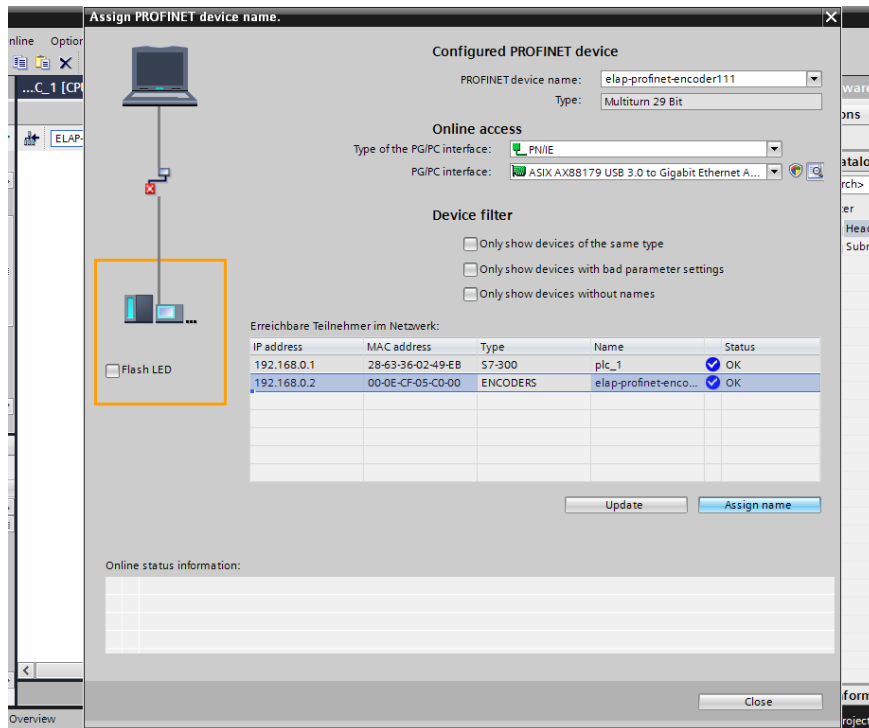
Right-click on the encoder image, then select “Assign Device Name”.

NOTE: Devices must be supplied and connected during this procedure.



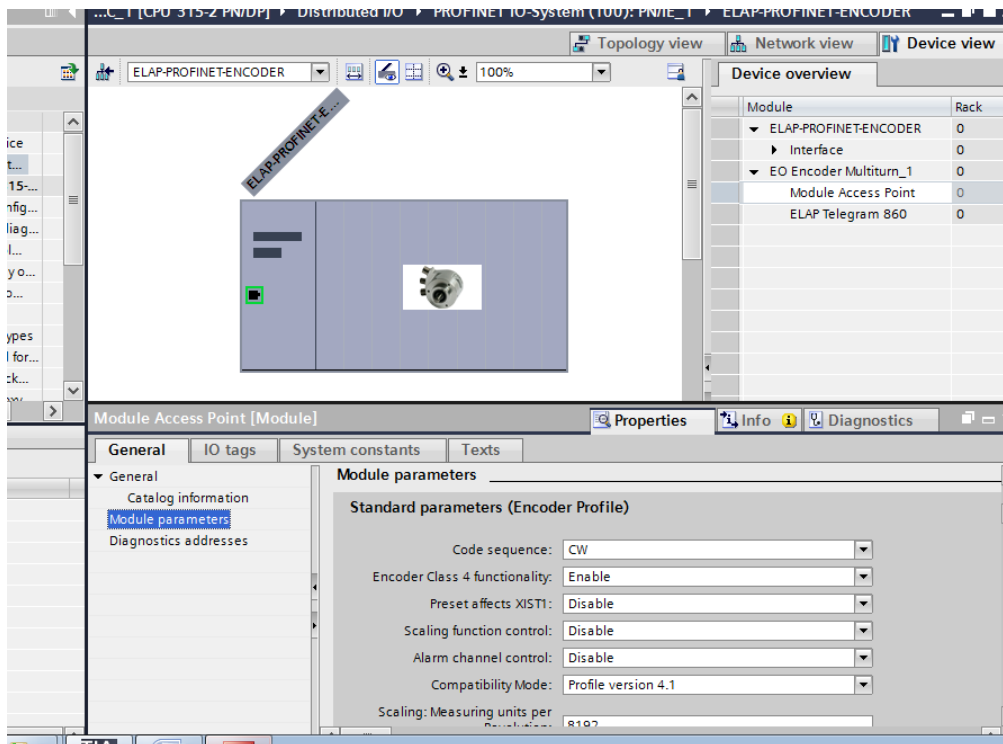
Choose the device to be renamed, then click on the *Assign name button* to adopt the changes.

After changing the device name, it is recommended to double check the name change by means of the *Update button*; then leave by the *Close button*. The MAC address of the encoder is written on the encoder label.

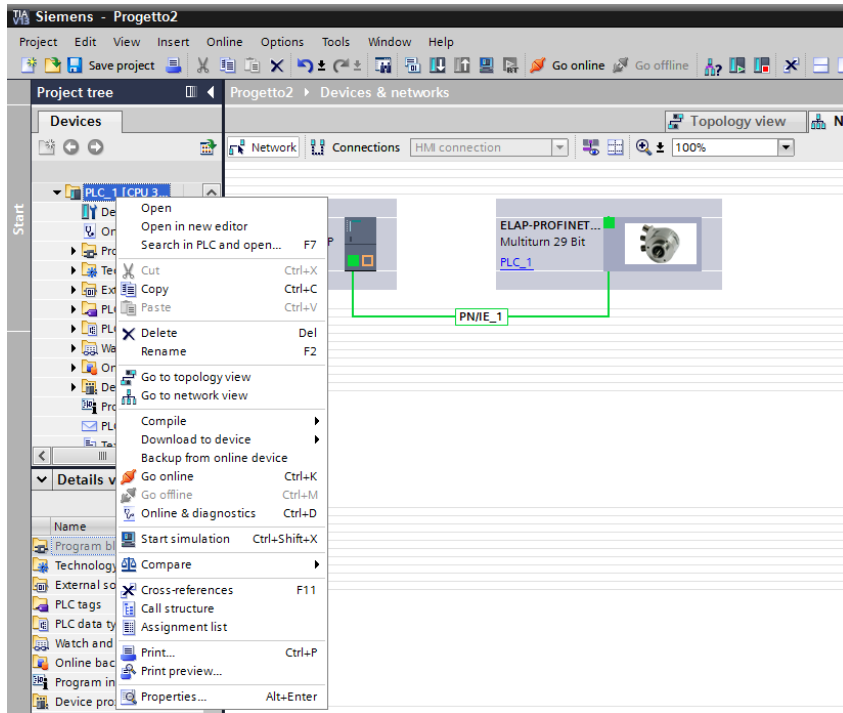


3.4 Setting the encoder parameters

To set the encoder user parameters select the *“Module Access Point”* field in *Device View*, then choose the *“Module Parameters”* tab. To set the parameter data, change the value of the different parameters by clicking on the drop down list in the *Value field* for the respective parameter.



After configuration and parameterization of the device, the settings must be saved and compiled by clicking on the command *Save Project*. Then the settings are to be downloaded to the IO-controller by clicking on the command *PLC → Download to Device*.

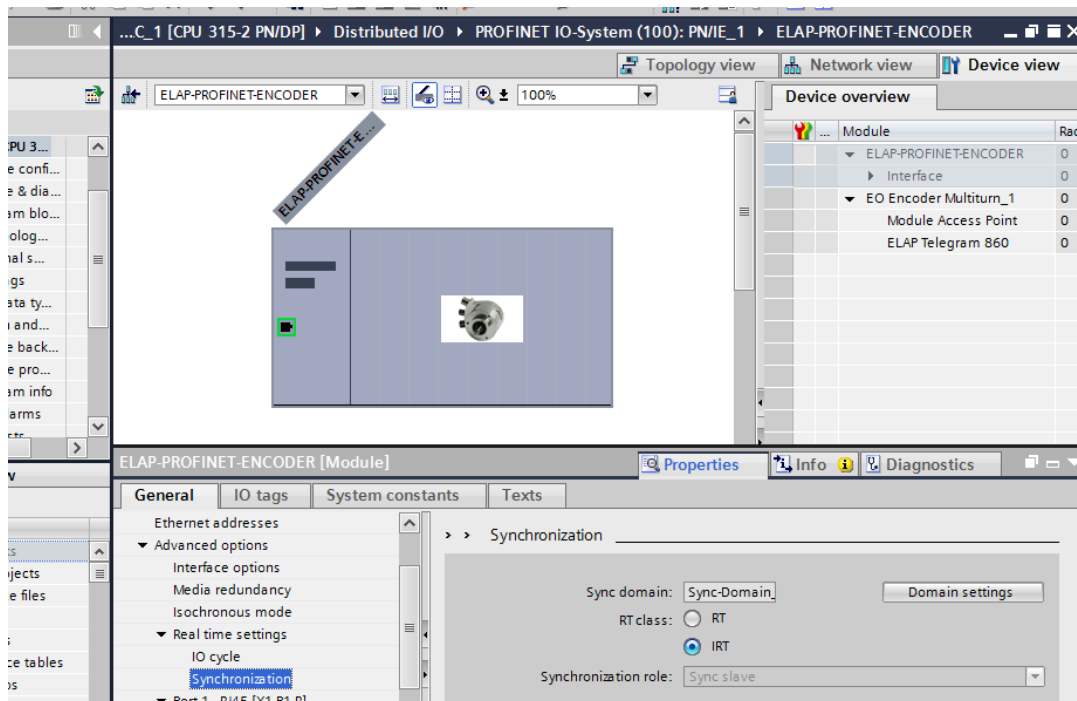


Configuration data are accessible run time, for read and write operations, at address 0xBF00.

3.5 Isochronous Real Time Setting (IRT)

The basic procedures for encoder configuration and parameterization are the same as described above.

To enter the IRT settings of the encoder, double click on the Encoder Image to open the *Properties* window.

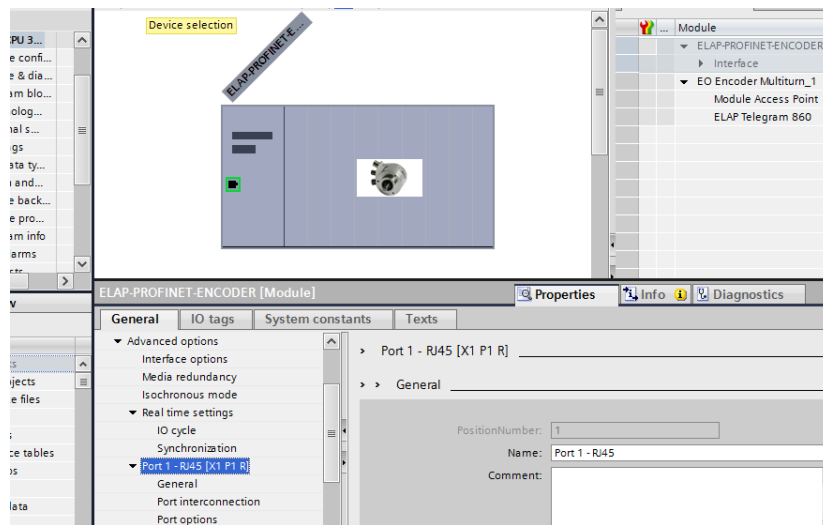


Select “Advanced Option → Real time Settings → Synchronization” tab to change the value of the Parameter RT Class to IRT.

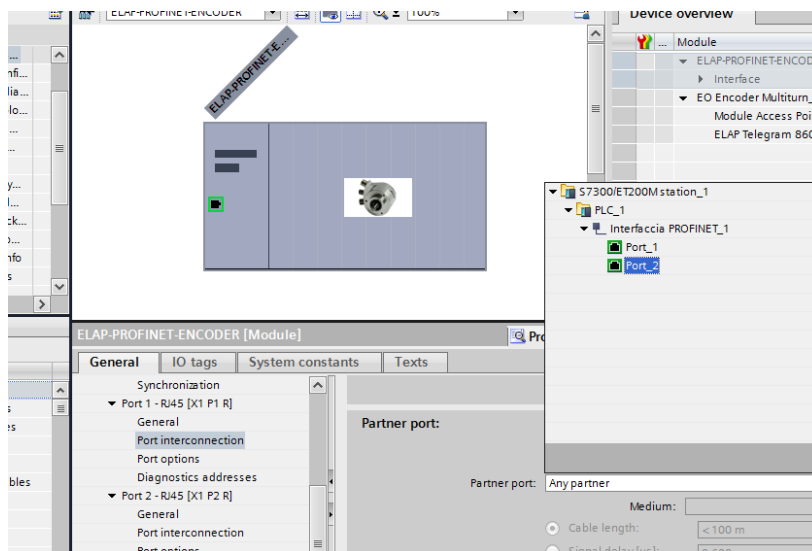
Before the encoder can operate in IRT mode, it is necessary to set from which port of the encoder the connection to the network has been done.

To set the topology, double click on the port from which the encoder is connected to the network.

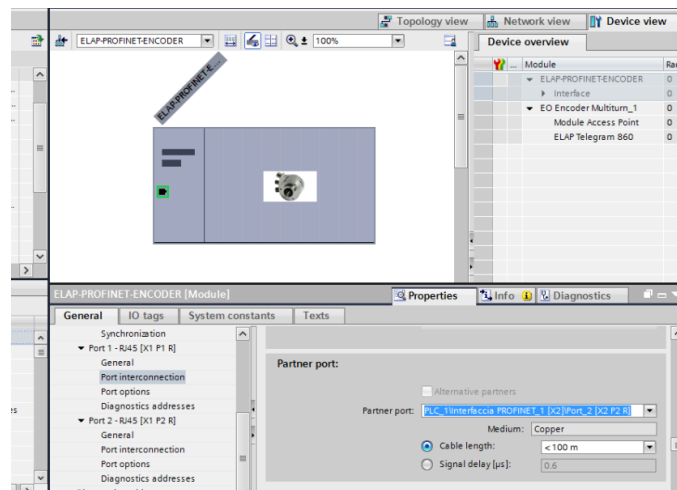
This is either slot 0.P1 (port 1, RJ45) or slot 0.P2 (port 2, RJ45). In the example in figure below Port 1 is used on the encoder.



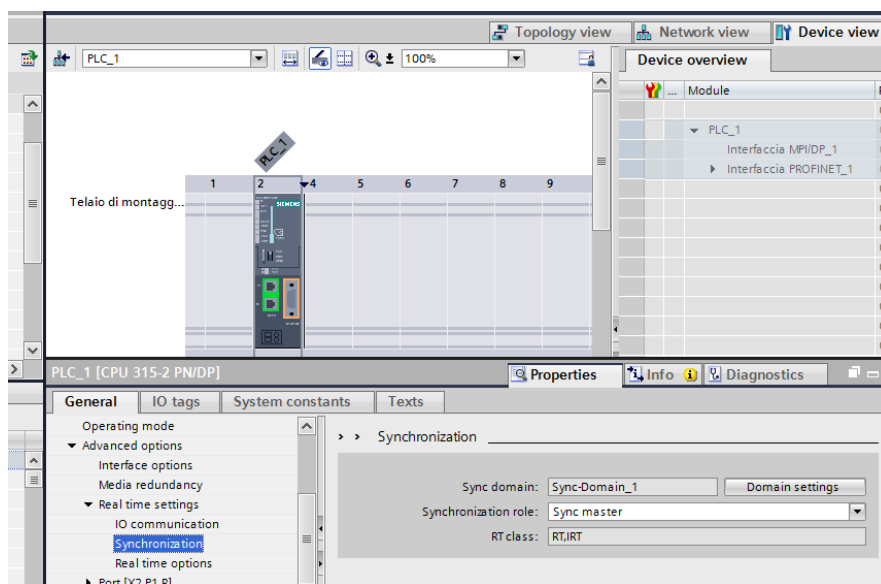
Select the “Port Interconnections” tab and set the “Partner Port”, that is the port from which the IO Controller is connected to the network.



Now the selected port appears in the “Partner Port” box.



Set the option *Sync Master* for the controller IO in *Advanced Option* → *Real time Settings* → *Synchronization*.



Now the encoder is ready to operate in IRT mode.

4 Configuration with STEP7

This chapter illustrates how to setup and configure a PROFINET encoder to work in RT and IRT modes, using STEP7 programming tool.

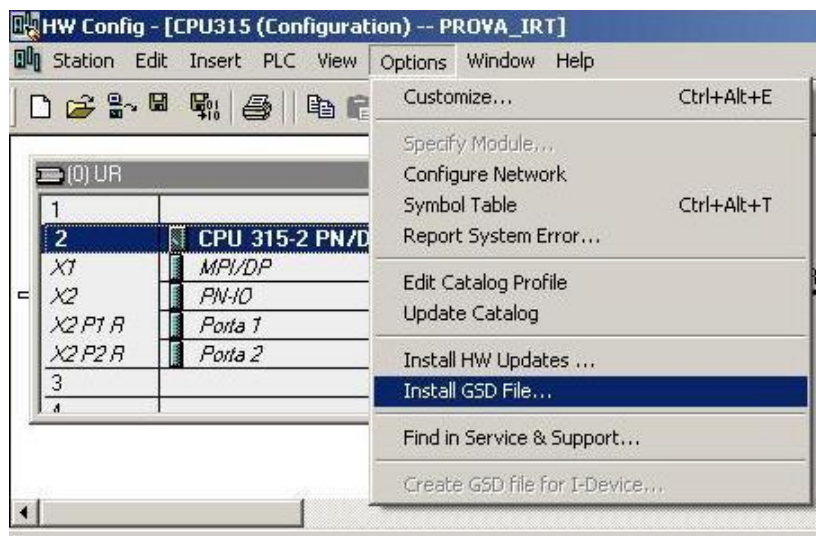
The following examples were generated with

- SIMATIC STEP7 V5.5 + SP3 programming tool
- Siemens CPU315-2PN/DP 315-2EH14-0AB0 V3.2
- PROFINET ELAP Encoder MEM540BPNTM10

These examples are about a project in which the main CPU was already inserted. More details about the programming tool can be found in the examples provided directly by Siemens, at link: <http://support.automation.siemens.com/WWW/view/it/40263542/0/it>

4.1 Device description file installation (GSDML)

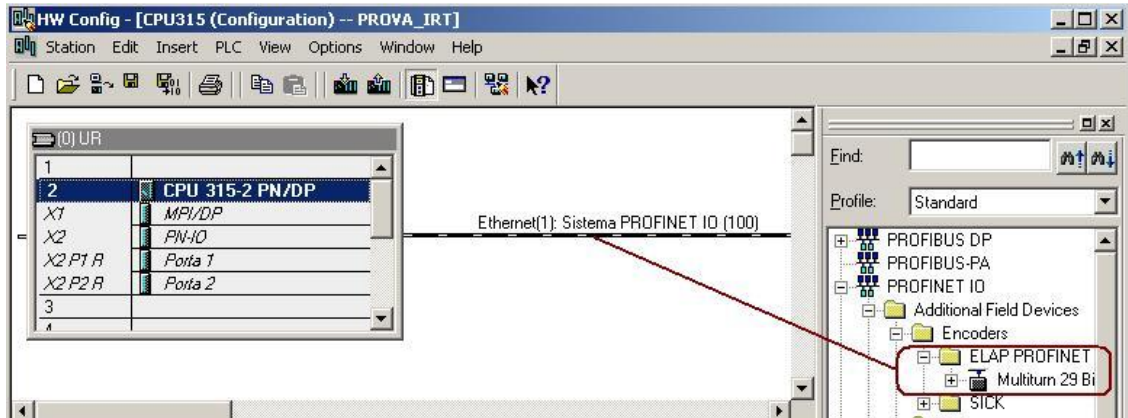
In order to start using an absolute encoder with PROFINET interface, a device description file needs to be imported into the configuration software. The device description file is a Generic Station Description Markup Language file, which contains the necessary implementation parameters needed for a PROFINET IO device.



1. Select *Options* → *Install GSD File* and click the *Browse button* to navigate to the location of the GSD file. If a bitmap picture representing the encoder is requested, make sure that the bitmap file is located in the same folder as the GSDML file. A bitmap file is provided with the GSDML file by ELAP.
2. Select the GSD file and click the *Install button* to start installing the file selected.

4.2 Setting the encoder configuration

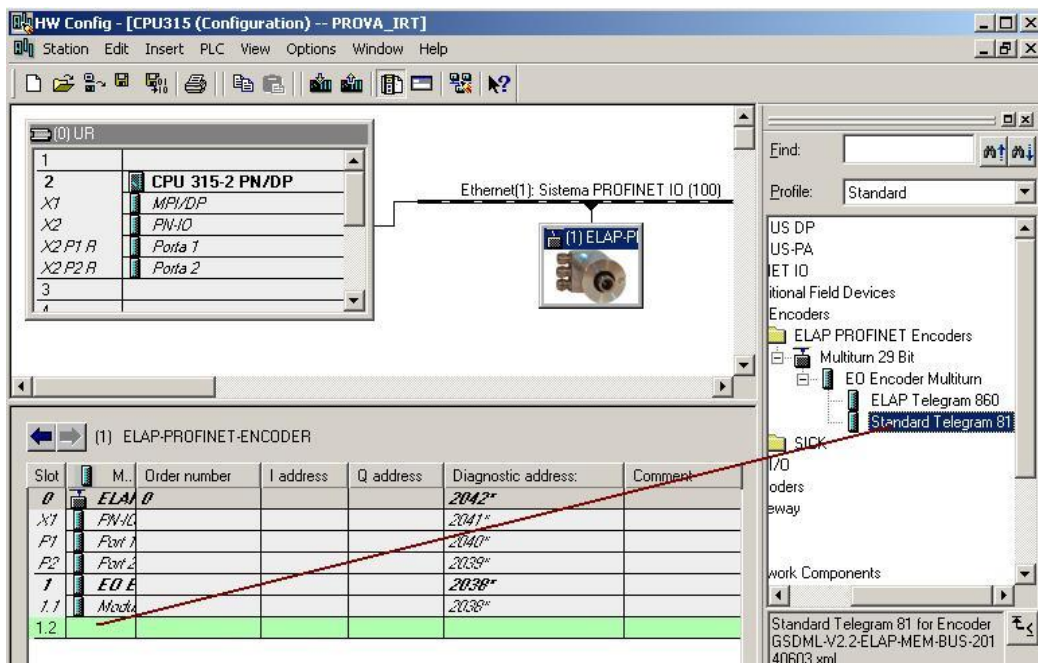
When the GSD file has been installed, the supported encoder types can be found in the HW Configuration under *PROFINET IO* → *Additional Field Devices* → *Encoders* → *ELAP PROFINET Encoder*. Select the encoder, and drag and drop it onto the PROFINET IO system as shown in the picture below. If more than one encoder shall be configured, then the following steps must be done once for each device.



When correctly done, the encoder will appear on the PROFINET IO system.

The next step will be to choose the type and length of data that should be sent to and from the IO controller. This is done by choosing one of different telegrams. The available telegrams for the Multi-turn 29 bit encoder can be found under *Multi-turn 29 bit* → *EO Encoder Multi-turn*.

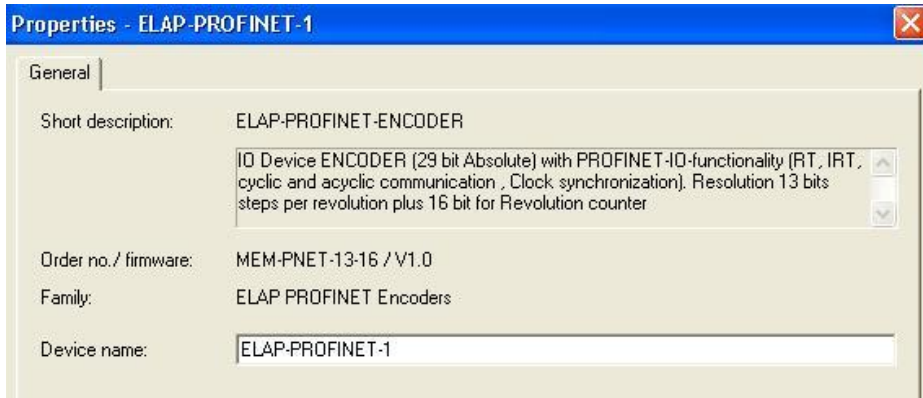
In the example below, standard telegram 81 is used. Drag and drop the telegram onto slot 1, sub slot 2, as shown in the figure below.



NOTE: The steps above are to be performed once for each device.

4.3 Setting the encoder device name

In a PROFINET network all IO devices have got a unique device name. To set the encoder device name, double click on the encoder icon to open the *Properties* window. The default name for the encoder appears in the “*Device name*” field. Enter an appropriate device name (for example “ELAP-PROFINET-1”) and then click the *OK* button.



Then select *PLC* → *Ethernet* → *Assign Device Name* to open the *Assign device name* window.

Choose the device on which the device name should be changed and then click on the *Assign name button* to adopt the changes, then click on the *Close* button. The MAC address of the encoder is written on the encoder label.

NOTE: All connected devices need to be assigned a unique device name.

After changing device name, it is recommended to check the name change by opening the *Verify Device Name* window under *PLC* → *Ethernet* → *Verify Device Name*.

4.4 Setting the encoder parameters

To set the encoder user parameters double click on the *Parameter Access Point* field, located under slot 1.1 as shown in the figure below, to open the *Properties* window.

Slot	Module	Order number	I address	Q address	Diagnostic a...	C...
0	ELAP-PROFINET-1	MEM-PNET-13-16			2042*	
X1	FN-IO				2041*	
F1	Part 1 - R145				2040*	
F2	Part 2 - R145				2039*	
1	EO Encoder Multit...				2038*	
1.1	Module Access Point				2038*	
1.2	Standard Telegram 81		0...11	0...3		

Choose the "Parameters" tab. To set the parameter data, change the value of the different parameters by clicking on the drop down list in the *Value field* for the respective parameter.

Parameter	Value
Code sequence	CW
Encoder Class 4 functionality	Enable
Preset affects XIST1	Disable
Scaling function control	Disable
Alarm channel control	Disable
Compatibility Mode	Profile version 4
Scaling: Measuring units per Rev...	8192
Scaling: Total measuring range	536870912
Tolerated sign of life faults	1

After configuration and parameterization of the device, the settings must be saved and compiled by clicking on the command *Station* → *Save and Compile*.

Then the settings are to be downloaded to the IO-controller by clicking on the command *PLC* → *Download*.

This data record is accessible run time, for read and write operations, at address 0xBF00.

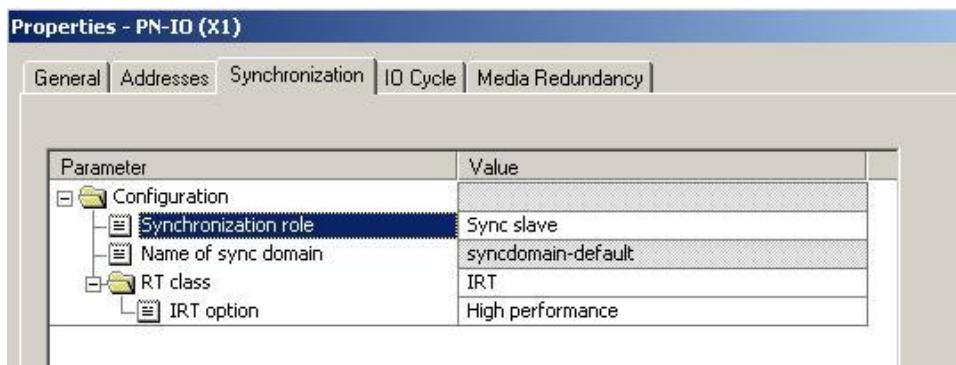
4.5 Isochronous Real Time Setting (IRT)

The basic procedures for encoder configuration and parameterization are the same as described above.

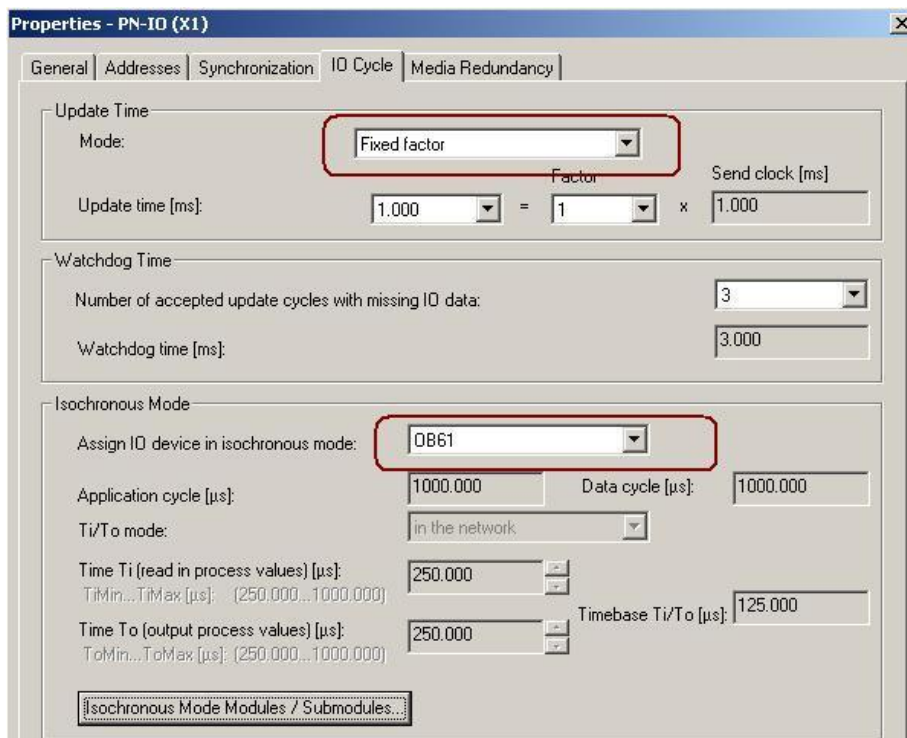
To enter the IRT settings of the encoder, double click on the “PN-IO” field (slot 0.X1) to open the *Properties* window.

Slot	Module	Order number	I address	Q address	Diagnostic a...	C...
0	ELAP-PROFINET-AMEM-PNET-13-16				2042*	
X1	PN-IO				2041*	
F1	Port 1 - RJ45				2040*	
F2	Port 2 - RJ45				2039*	
1	ED Encoder Multit...				2038*	
1.1	Module Access Point				2038*	
1.2	Standard Telegram 81		0...11	0...3		

Select “Synchronization” tab to change the value of the Parameter RT Class to IRT and then choose the “Sync slave” and IRT “High Performance” options, according to the picture below.



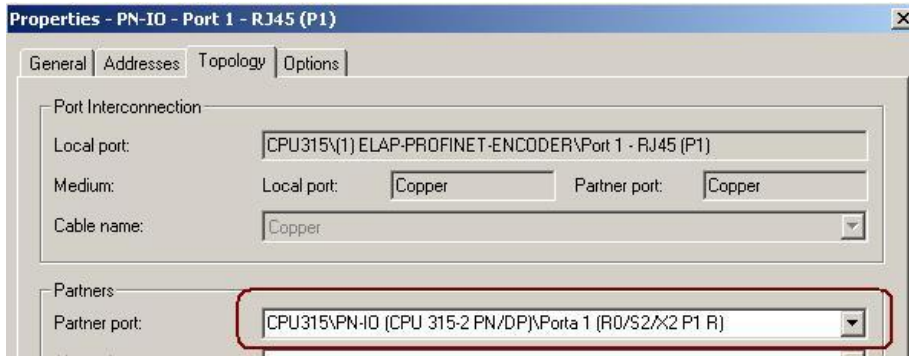
Select “IO Cycle” tab to choose the Update Time in “Fixed factor” mode, and assign the encoder device to the synchronization object OB61.



Before the encoder can operate in IRT mode it is necessary to set from which port of the encoder the connection to the network has been done.

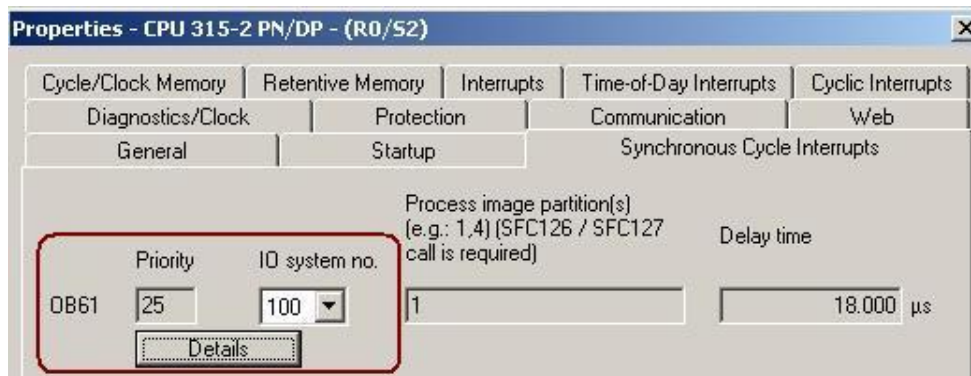
To set the topology, double click on the port from which the encoder is connected to the network.

This is either slot 0.P1 (port 1, RJ45) or slot 0.P2 (port 2, RJ45). In the example in figure below Port 1 is used on the encoder.



Select the “Topology” tab and set the “Partner Port”, that is the port from which the IO Controller is connected to the network.

After these steps, set the IRT “High Performance” mode for the IO Controller, and enable the synchronization signal of the OB61 object.



It is recommended to check all the settings for the encoder and the IO Controller by the command *Edit* → *PROFINET IO* → *Domain Management*.

Make sure that in the *Domain* window the encoder class is IRT with “High Performance” option.

Now, your encoder is ready for IRT operating mode.

5 PROFINET IO data description

According to the Encoder Profile V4.1 (PNO 3.162), encoders are divided into two classes, Class 3 and Class 4.

For further information regarding the encoder functionality refer to the device profile. The profile and PROFINET technical information can be ordered at PNO in Karlsruhe, Germany (www.profinet.com).

5.1 Application Class definition

PROFINET encoders can be configured as a Class 3 or Class 4 PROFINET IO device(see 3.4 and 4.4), according to the encoder profile V.4.1 (PNO 3.162). A Class 4 configured encoder fully supports all functionalities according to the encoder profile V4.1.

- **CLASS 3:** Encoder with base mode parameter access and limited parameterization of the encoder functionality. Isochronous mode is not supported.
- **CLASS 4:** Encoder with scaling, Preset and base mode parameter access. Isochronous mode is supported.

5.2 Standard signals

The table below describes the standard signals that are used to configure the IO data.

Meaning	Abbreviation	Data type	Data Flow
Velocity Value A	NIST_A	Signed 16	Enc. =>PLC
Velocity Value B	NIST_B	Signed 32	Enc. =>PLC
Control Word 1	G1_STW	Unsigned 16	PLC =>Enc.
Status Word 1	G1_ZSW	Unsigned 16	Enc. =>PLC
Position Value 1	G1_XIST1	Unsigned 32	Enc. =>PLC
Position Value 2	G1_XIST2	Unsigned 32	Enc. =>PLC
Position Value 3	G1_XIST3	Unsigned 64	Enc. =>PLC
Control Word 2	STW2_ENC	Unsigned 16	PLC =>Enc
Status Word 2	ZSW2_ENC	Unsigned 16	Enc. =>PLC

5.3 Telegrams

The configuration of PROFINET encoders is performed by choosing different telegram structures. The telegrams are used to specify the data length, and which type of data are sent to and from the IO controller.

The following telegrams are supported:

- **ELAP Telegram 860:** it uses 4 bytes for output data from the IO Controller to the encoder and 4 bytes of input data from the encoder to the IO Controller. It is the same used by the PROFIBUS protocol.

Output Data from the IO Controller:

Preset value (4 byte, Bit 31 used as command trigger).

Input Data to the IO Controller:

Position value (4 byte).

- **Standard Telegram 81:** it uses 4 bytes for output data from the IO Controller to the encoder and 12 bytes of input data from the encoder to the IO Controller.

Output Data from the IO Controller:

STW2_ENC (2 byte)
G1_STW (2 byte)

Input Data to the IO Controller:

ZSW2_ENC (2 byte)
G1_ZSW (2 byte)
G1_XIST1 (4 byte)
G1_XIST2 (4 byte)

- **Standard Telegram 82:** it uses 4 bytes for output data from the IO Controller to the encoder and 14 bytes of input data from the encoder to the IO Controller.

Output Data from the IO Controller:

STW2_ENC (2 byte)
G1_STW (2 byte)

Input Data to the IO Controller:

ZSW2_ENC (2 byte)
G1_ZSW (2 byte)
G1_XIST1 (4 byte)
G1_XIST2 (4 byte)
NIST_A (2 byte)

- **Standard Telegram 83:** it uses 4 bytes for output data from the IO Controller to the encoder and 16 bytes of input data from the encoder to the IO Controller.

Output Data from the IO Controller:

STW2_ENC (2 byte)
G1_STW (2 byte)

Input Data to the IO Controller:

ZSW2_ENC (2 byte)
G1_ZSW (2 byte)
G1_XIST1 (4 byte)
G1_XIST2 (4 byte)
NIST_B (4 byte)

- **Standard Telegram 84:** it uses 4 bytes for output data from the IO Controller to the encoder and 20 bytes of input data from the encoder to the IO Controller.

Output Data from the IO Controller:

STW2_ENC (2 byte)
G1_STW (2 byte)

Input Data to the IO Controller:

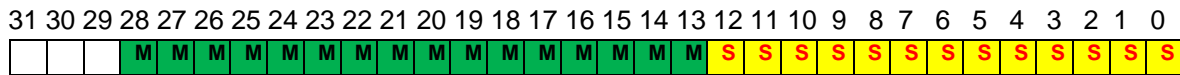
ZSW2_ENC (2 byte)
G1_ZSW (2 byte)
G1_XIST3 (8 byte)
G1_XIST2 (4 byte)
NIST_B (4 byte)

5.4 Format of Position Values in G1_XIST1 and G1_XIST2

The G1_XIST1 and G1_XIST2 signals consist of the absolute position value in binary format. By default the G1_XIST1 signal is equal to the G1_XIST2 signal. In particular:

- All values are presented in binary format.
- The shift factor is zero (right aligned value) for both G1_XIST1 and G1_XIST2.
- The setting in the encoder parameter data (measuring range) affects the position value in both G1_XIST1 and G1_XIST2. The preset value, transmitted via acyclic process data, has an effect on G1_XIST1 if the parameter "G1_XIST1 Preset Control" is active.
- G1_XIST2 displays the error code instead of the position value if any error occurs.

ELAP encoder is an absolute multi-turn encoder with 29-bit resolution (8192 steps per revolution, 65536 distinguishable revolutions).



MSB

LSB

M = Multi-turn value (number of revolutions).
 S = Position in single turn.

5.5 Format of the Position Value in G1_XIST3

In addition to the actual position value defined in the PROFIdrive Profile (Gx_XIST1 and Gx_XIST2) a 64-bit position value named G1_XIST3 is defined to support encoders with a measuring length exceeding 32 bits.

G1_XIST3 has the following format:

- Binary format
- The actual position value is always right aligned, no shifting factor is used
- The settings in the encoder parameter data affect the position value in G1_XIST3 if Class 4 is enabled.(See 3.4 and 4.4)

5.6 Control Word 2 (STW2_ENC)

The control word 2 (STW2_ENC) is referred to as the master Sign Of Life, and it includes the fault buffer handling and Control by PLC mechanism from PROFIdrive STW1 and the IO Controller Sign-Of-Life mechanism from PROFIdrive STW2.

Detailed assignment of control word 2 (STW2_ENC) is shown in the table below.

Bit	Function
0... 6	Reserved
7	Fault Acknowledge
8, 9	Reserved
10	Control by PLC
11	Reserved
12...15	IO Controller "Sign-of-Life" counter

Bit 7, Fault Acknowledge:

- 1 → The fault signal is acknowledged with a positive edge. The encoder reaction to a fault depends on the type of fault
- 0 → No significance

Bit 10, Control by PLC:

- 1 → Control via interface, EO IO Data is valid
- 0 → EO IO Data is not valid, except Sign-Of-Life

Bit 12... 15, IO Controller "Sign-of-Life" counter.

NOTE: In case of operation with Technology Objects, see Appendix E of the present manual.

5.7 Status Word 2 (SZW2_ENC)

The status word 2 (ZSW2_ENC) is referred to as the slave's Sign Of Life and it includes the fault buffer handling and Control by PLC mechanism from PROFIdrive ZSW1 and the Slave Sign-Of-Life mechanism from PROFIdrive ZSW2.

Detailed assignment of status word 2 (SZW2_ENC) is shown in the table below.

Bit	Function
0... 2	Reserved
3	Fault present / No fault
4... 8	Reserved
9	Control requested
10, 11	Reserved
12... 15	Encoder "Sign-of-Life" counter

Bit 3, Fault present:

- 1 → Unacknowledged faults or currently not acknowledged faults (fault messages) are present. The fault reaction is fault-specific and device-specific. The acknowledging of a fault may only be successful, if the fault cause has disappeared or has been removed before. If the fault has been removed, the encoder returns to operation.
- 0 → No fault.

Bit 9, Control requested:

- 1 → The automation system is requested to assume control.
- 0 → No control requested.

Bit 12... 15, Encoder “Sign-of-Life” counter.

5.8 Control Word G1_STW

This word controls the functionality of major encoder functions.

Detailed assignment of control word G1_STW is shown in the table below.

Bit	Function
0... 7	Function requests: Reference mark search, etc.
8... 10	Reserved
11	Preset mode (absolute or relative)
12	Preset request
13	Request absolute value cyclically
14	Activate parking sensor
15	Acknowledging a sensor error

Note: If the sensor parking is activated (bit 14 = 1) the encoder is still on the bus with the slave Sign Of Life active and the encoder error and diagnostics switched off.

Note: In case of operation with Technology Objects, see Appendix E of the present manual

5.9 Status Word G1_SZW

This word defines encoder states, acknowledgements, error messages of major encoder functions.

Detailed assignment of status word G1_SZW is shown in the table below.

Bit	Function
0... 7	Function status: Reference mark search, etc.
8	Probe 1 deflected
9	Probe 2 deflected
10	Reserved (set to 0)
11	Requirements of error acknowledgment detected
12	Preset operation executed
13	Transmit absolute value cyclically
14	Parking sensor active
15	Sensor error

Note 1: If bit 13 (Transmit absolute value cyclically) or bit 15 (Sensor error) is not set, there is no valid value or error code transferred in G1_XIST2.

Note 2: Bit 13 (Transmit absolute value cyclically) cannot be set at the same time as bit 15 (Sensor error) as these bits are used to indicate either a valid position value transmission (bit 13) or the error code transmission (bit 15) in G1_XIST2.

5.10 Preset function

The preset function is controlled by bit 11 and bit 12 in the control word G1_STW and acknowledged by bit 12 in the status word G1_SZW. The preset value is 0 by default and may be set by an acyclic data exchange parameter defined in the parameter section.

The preset function has an absolute and a relative operating mode, selectable by bit 11 in the control word G1_STW. In particular:

- Normal operating mode: **Bit 12 = 0**
The encoder will make no change in the output value.
- Preset mode absolute: **Bit 11 = 0, Bit 12 = 1**
The encoder reads the current position value and calculates an internal offset value from the preset value and the real position value. The position value is then shifted with the calculated offset value to get a position actual value equal to the preset value. Negative values are not accepted in this case for the preset position.
- Preset mode relative: **Bit 11 = 1, Bit 12 = 1**
The encoder uses the preset value as a relative offset value. The position actual value is shifted by the value taken from the preset value. Positive and negative values are both accepted for the preset position.

If **ELAP telegram 860** is used, the preset function is executed just like in PROFIBUS-DP.

The preset value is transferred to the encoder in the 4 byte output value from the IO Controller twice at least, the first time with the highest bit active (MSB = 1) and then with the highest bit low (MSB = 0). The MSB bit is used as command trigger, so the preset value is limited in a 31 bit interval of values. Only the absolute preset mode is possible in this case.

Example: Resetting the encoder position value (preset = 0).

1. The IO Controller transmits 0x80000000
2. The IO Controller transmits 0x00000000

The encoder reads the current position value and calculates an internal offset value from the preset value and the real position value. The position value is then shifted with the calculated offset value to get a position actual value equal to the preset value. Negative values are not accepted in this case for the preset position.

5.11 Real-Time communication

PROFINET IO uses three different communication channels to exchange data with programmable controllers and other devices.

The non real time channel based on TCP (UDP)/IP, for example, is used for parameterization, configuration and acyclic read/write operations.

The RT or Real Time channel is used for process data transfer and alarms. Real-time data are treated with a higher priority than data sent over the open channel. RT communications override the open channel to handle the data exchange with programmable Controllers.

The third channel, Isochronous Real Time (IRT) is the high performance, high speed channel used for demanding motion control applications. IRT data are treated with a higher priority than RT data sent over the RT channel.

PROFINET distinguishes among three real time classes for transmission of time critical process data.

Real-Time, RT Class 1

The typical cycle time for data exchange is about 100 ms.

- Unsynchronized Real-Time communication.
- Industrial standard switches can be used.
- Typical application area: factory automation.

Real-Time, RT Class 2

The typical cycle time for data exchange is about 10 ms.

- Synchronized and unsynchronized data transmission.
- Special switches supporting IRT are needed.
- Typical application area: factory automation.

Isochronous Real-Time, class 3

The isochronous operation mode is used when real-time positioning with high performance is required. The basic principle is that all PROFINET devices on the net are clock-synchronized with the controller, that using a global control broadcast, enables the simultaneous data accusation from all devices with microsecond accuracy. The data exchange cycles for IRT are usually in the range of a few hundred microseconds up to a few milliseconds. The difference from real-time communication is essentially the high degree of determinism, so that the start of a bus cycle is maintained with high precision. The synchronization is monitored by sign-of life messages in Control word 2 (STW2_ENC) and Status word 2 (ZSW2_ENC).

- Clock-synchronized data transmission.
- Special switches supporting IRT are needed.
- IRT is required, for example, in motion control applications.

6 IRT Communication and synchronization

For both transmission directions (Controller \leftrightarrow DO), user data reliability is achieved using a Sign-Of-Life 4-bit counter. The value range of the Sign-Of-Life is only 1 to 15 respectively (0 = invalid).

6.1 Controller Sign-of-Life (C-LS)

Transmission C-LS

A 4-bit counter is used in Control Word 2 (STW2_ENC) as the Sign-Of-Life for the controller. This counter is incremented by the controller in each controller application cycle, and thus also identifies the computation of the position controller (first DP cycle in the TMAPC). The DO receives the new Sign-Of-Life of the controller together with the new set-point at the time TO in the following DP-cycle.

Synchronization C-LS

The Controller application starts the Controller-LS with an arbitrary value between 1 and 15, at the earliest when changing from Preparation \rightarrow Synchronization.

Monitoring C-LS

If in a Controller application cycle, the DO application does not recognize a correct count (i.e. a positive or a negative deviation is recognized), it initially processes with the old telegram data from the last valid controller telegram. For set-point generation, a device-specific failure strategy may be used.

If DO application does not recognize the expected numerical value after a parameterized number of controller application cycles ($TMLS = n \times TMAPC$), the affected Drive Axis messages a fault. After fault acknowledgement, the DO application then attempts to automatically re-synchronize itself to the Sign-Of-Life of the controller application. Depending on the particular application, a new start may be required.

If the Sign-Of-Life fails, it may be for the following reasons:

- Failure of the controller application level (with DP transmission still operational)
- PLL failure
- The DP cycle TDP has been exceeded (through telegram repetition)

Example 1: Permanent LS failure, $TMLS = 5 \times TMAPC$ (see 5.3 for counting strategy)

T_{MAPC} :										
Controller LS (reference):	1	2	3	4	5	6	7	8	9	10
Controller LS (actual):	1	2	2	2	2	2	2	2	2	2
Failure counter:	0	0	10	20	30	40	50	50	50	50
Response:			-> Failure				-> Switch-off			

Example 2: Temporary LS failure, $TMLS = 5 \times TMAPC$ (see 5.3 for counting strategy)

T_{MAPC} :										
Controller LS (reference):	1	2	3	4	5	6	7	8	9	10
Controller LS (actual):	1	2	2	2	5	6	7	8	9	10
Failure counter:	0	0	10	20	19	18	17	16	15	14
Response:			-> Failure							

Temporary failure with negative deviation.

T _{MAPC} :										
Controller LS (reference):	1	2	3	4	5	6	7	8	9	10
Controller LS (actual):	1	2	4	5	5	6	7	8	9	10
Failure counter:	0	0	10	20	19	18	17	16	15	14
Response:			-> Failure							

Temporary failure with positive deviation.

6.2 Device Sign-of-Life (DO-LS)

Transmission DO-LS

A 4-bit counter in status word 2 (SZW2_ENC) is used as a Sign-Of-Life for the DO. The DO increments this counter at each DP cycle.

Synchronization DO-LS

The DO application starts the DO's Sign-Of-Life with an arbitrary value between 1 and 15, after successful PLL synchronization and at the change (n -> n + 1) of the controller Sign-Of-Life.

Monitoring DO-LS

If the controller application does not recognize a correct count in a controller application cycle (i.e. a positive or negative deviation has been recognized), it initially uses the old telegram data from the last valid DO telegram. To generate the actual value, a device-specific failure strategy may be implemented.

If the controller application does not recognize the expected numerical value after a parameterized time (TSLs = n x TDP), the affected Drive Axis is shut down by the controller application (possibly also involved drives), and an appropriate fault is signaled to the user. The controller application then attempts to automatically re-synchronize itself to the Sign-Of-Life of the DO application. Depending on the particular application, a re-start may be required or it may be sufficient to acknowledge the fault.

Example reasons for the Sign-Of-Life to fail may be:

- Failure of the DO application level (while DP transmission is still functioning)
- PLL failure
- DO failure in the sense of DP (DO does not respond although telegram was repeated)

Example 1: Permanent LS failure TSLs = 5 x TDP (see 5.3 for counting strategy)

DP cycle:										
DO LS (reference):	1	2	3	4	5	6	7	8	9	10
DO LS (actual):	1	2	2	2	2	2	2	2	2	2
Failure counter:	0	0	10	20	30	40	50	50	50	50
Response:			-> Failure				-> Switch-off			

Example 2: Temporary LS failure TSLs = 5 x TDP (see 5.3 for counting strategy)

DP cycle:										
DO LS (reference):	1	2	3	4	5	6	7	8	9	10
DO LS (actual):	1	2	2	2	5	6	7	8	9	10
Failure counter:	0	0	10	20	19	18	17	16	15	14
Response:			-> failure							

Temporary failure with negative deviation.

DP cycle:										
DO LS (reference):	1	2	3	4	5	6	7	8	9	10
DO LS (actual):	1	2	4	5	5	6	7	8	9	10
Failure counter:	0	0	10	20	19	18	17	16	15	14
Response:			-> failure							

Temporary failure with positive deviation.

6.3 Counting strategy for the Sign-of-Life failure counter

The strategy which is applied in order to prevent fast shutdown for a sporadically faulted controller or DO application is described in the following text. This strategy guarantees that at least a specific percentage of the telegrams shall be valid before a Drive Axis is powered down.

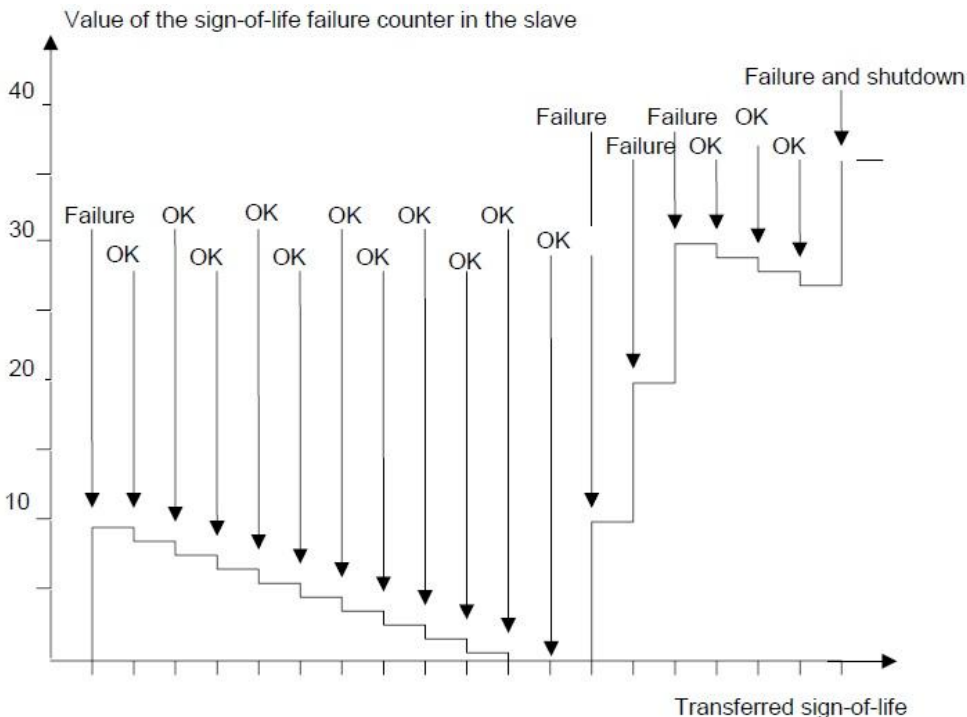
A counter is defined on the DO side, in which for each deviation (regardless whether it is a positive or negative deviation) between the expected and actually transferred value for the controller Sign-Of-Life, it is incremented by ten. For each additional deviation, the counter is incremented by ten again. If a deviation between the expected and received controller Sign-Of-Life is not recognized, the counter is decreased by one. The minimum value which may then be counted down to is zero. This is simultaneously the value from which counting is started.

This method ensures that more than 90% of the telegrams transferred in continuous operation originate from an undisturbed controller application.

Depending on the previous history, it is possible that even a few controller Sign-Of-Life failures are enough to cause a failure of a Drive Axis. If the Drive Axis is powered-down, the Sign-Of-Life failure counter maintains its value up to the start of the re-synchronization operation.

In the following example, the Sign-Of-Life failure counter in the Drive Axis is viewed over time with respect to the transferred controller Sign-Of-Life. The maximum number of controller Sign-Of-Life failures which may be tolerated is set to three.

The same strategy is recommended when monitoring the DO Sign-Of-Life in the controller. However no parameter relating to the maximum number of tolerable DO Sign-Of-Life character failures has been defined.



7 Alarms and warnings

7.1 Diagnostics and Alarms

Diagnostic data are always transferred using Record Data communications over the non real time channel. An IO Supervisor must specifically request the diagnostic or status data from the IO device using RDO (Record Data Object) services.

Alarm data are transmitted from the IO device to the IO controller via the RT channel. Alarm is generated by the encoder in case of failure the position value. Alarms can be reset (deleted) when all encoder parameters are within the specified value ranges and the position value is correct.

7.2 Channel diagnostics

The encoder outputs a diagnostic interrupt to the CPU when it detects one of the supported channel diagnostics:

- Position error: code 0x900A (36874)
The encoder fails to read the correct position value as the data stored into RAM (preset and offset values) are corrupted. It is advisable to contact ELAP service.
- Battery failure: code 0x9000 (36864)
The back up battery voltage has got to a critical level. It is advisable to contact ELAP service.

In a SIMATIC STEP 7 system the operation system responds by calling a diagnostic OB. The OB number and start information provides the cause and location of the error. The error information can be read by calling a system Function block (SFB54 RALRM for STEP 7). Then the user can decide how the system must handle the error.

Note: If the called OB is not included in the PLC program, the CPU will go to stop.

7.3 Sensor status word

Diagnosis information can be obtained by monitoring the Error bit in the Sensor Status word G1_ZSW (Bit 15) and evaluating the error code transmitted in G1_XIST2.

Supported diagnostic	Error code in G1_XIST2	Description
Memory error	0x1001	The data stored into RAM are corrupted.
Battery at critical level	0x1002	The back up battery voltage has got to a critical level.
Master Sign-of-Life fault	0x0F02	The number of failures of the controller's life sign was exceeded.

8 Acyclic Parameter Data

8.1 Acyclic data exchange

In addition to the cyclic data exchange, the PROFINET encoder also supports acyclic data exchange. The acyclic data exchange is transferred over the non-real time channel and is used to read out and write status information from and to the IO device. The acyclic data exchange is performed in parallel to the cyclic data communication.

Example of acyclic data:

- Reading of diagnostic
- Reading of I&M functions (record 0xAFF0)
- Reading / Writings of PROFIdrive parameters (record 0xB02E)
- Reading / Writing configuration data (record 0xBF00)

8.2 Identification and Maintenance (I&M functions)

ELAP encoder, according to the encoder profile 3.162 and guidelines 5.502, also supports I&M functionality.

The main purpose of the I&M functions is to support the end user if the device is running faulty or missing some of its functionality. The I&M functions could be seen as an electronic nameplate containing common information regarding the device and its manufacturer.

According to the PROFINET specification, all IO devices must support at least the following I&M functions:

- Order ID
- Hardware Version
- Software Version
- Product type
- Manufacturer ID

8.3 Base mode parameter access

Acyclic parameters can be transmitted by 1 (single) or in blocks - up to 39 (multi) - in one access. A parameter access can be up to 240 bytes long.

The request / response message is structured as follows:

Request ID	Drive-Object ID	Number of parameters	Parameter address	Parameter values
------------	-----------------	----------------------	-------------------	------------------

- Parameter address: one address for each parameter, if several parameters are accessed.
- Parameter values: if the Request ID is 0x02 (change value) the value is set in the request message, if the Request ID is 0x01 (request value) the value appears in the reply.

8.4 Changing the preset value, parameter 65000

Request message:

Request reference	0x01	
Request ID	0x02	0x01 → read value, 0x02 → change value
DO-ID	0x00	Drive Object identifier
Number of parameters	0x01	
Attribute	0x10	0x10 → value
Number of elements	0x01	
Parameter address	0xFDE8	Parameter 65000
Sub-index	0x0000	
Format	0x43	Data type: 0x41 → Byte, 0x42 → Word, 0x43 → Long
Number of values	0x01	Number of values = Number of elements
Value	0x00000064	Preset value = 100

Reply message:

Request reference	0x01	The same value as in the request message
Response ID	0x02	
DO-ID	0x00	The same value as in the request message
Number of parameters	0x01	
Format	0x43	
Number of values	0x01	

8.5 Reading the Preset value, parameter 65000

Request message:

Request reference	0x02	
Request ID	0x01	0x01 → read value, 0x02 → change value
DO-ID	0x00	Drive Object identifier
Number of parameters	0x01	
Attribute	0x10	0x10 → value
Number of elements	0x01	
Parameter address	0xFDE8	Parameter 65000
Sub-index	0x0000	
Format	0x43	Data type: 0x41 → Byte, 0x42 → Word, 0x43 → Long
Number of values	0x01	Number of values = Number of elements

Reply message:

Request reference	0x02	The same value as in the request message
Response ID	0x01	
DO-ID	0x00	The same value as in the request message
Number of parameters	0x01	
Format	0x43	
Number of values	0x01	
Value	0x00000064	Read value

8.6 Supported parameters

ELAP MEM-BUS encoder supports the PROFIdrive parameters of the Encoder Profile 3.162, V4.1 (see also section [9.13 Acyclic Data](#)).

General Parameters

P922 → Telegram selection

Unsigned16, read only, it presents which telegram is used. Telegrams 81, 82, 83, 84 or 860 are possible.

P964 → Device identification

Array[5] of Unsigned16, read only.

P964[0] = 0x02AB	Vendor ID (ELAP)
P964[1] = 0	Drive Unit type
P964[2] = xx.xx	Firmware version
P964[3] = yyyy	Firmware date (year)
P964[4] = ddmm	Firmware date (day.month)

P965 → Profile identification number

Octet String 2, read only.

P965[0] = 0x3D	Encoder profile number
P965[1] = 31 or 41	Profile version (depending on parameter Compatibility mode)

P971 → Data transfer to non-volatile memory

Unsigned16, write only. Writing this parameter commands local data (as preset value P65000) to be saved.

P975 → Encoder object identification

Array[8] of Unsigned16, read only.

P975[0] = 0x02AB	Vendor ID (ELAP)
P975[1] = 0	Drive Object type
P975[2] = xx.xx	Firmware version
P975[3] = yyyy	Firmware date (year)
P975[4] = ddmm	Firmware date (day.month)
P975[5] = 0x0005	PROFIdrive DO type classification (5 → Encoder interface)
P975[6] = 0x8000	PROFIdrive DO sub-classification 1 (Bit 15 = 1 → Encoder class 4)
P975[7] = 0x0001	Drive Object identifier (DO ID)

P979 → Sensor format

Array[6] of Unsigned32, read only.

P979[0] = 0x00005111	5 → max index, 1 → number of encoders, 11 → structure version
P979[1] = 0x80000002	Bit0 = 0 → Rotary encoder Bit1 = 1 → Absolute position in G1_XIST1 Bit2 = 0 → Position value 32 Bit . Bit31 = 1 → Configuration and parameterization OK
P979[2] = 8192	Single turn resolution (13 Bit)
P979[3] = 0	Shift factor for G1_XIST1
P979[4] = 0	Shift factor for G1_XIST2
P979[5] = 65536	Number of distinguishable revolutions

P980 → List of supported parameters

Array[n] of Unsigned16, read only.

P980[0] = 922	
P980[1] = 964	
P980[2] = 965	
P980[3] = 971	
P980[4] = 975	
P980[5] = 979	
P980[6] = 65000	
P980[7] = 65001	
P980[8] = 0	End of list

Encoder specific parameters

P65000 → Preset value

Integer32, Read / Write access parameter, it is used with standard telegrams 81, 82, 83, 84.

P650001 → Operating status

Array[12] of Unsigned32.

P65001[0] = 0x000B0101	Structure version (1.01), max index (11)
P65001[1] = Operating status	
P65001[2] = Faults	
P65001[3] = 0x00000021	Supported faults
P65001[4] = Warnings	
P65001[5] = 0x00000020	Supported warnings
P65001[6] = 0x00000401	Version of the Encoder Profile
P65001[7] = 0xFFFFFFFF	Operating time (non implemented)
P65001[8] = Offset value	
P65001[9] = Single turn resolution (max 8192)	
P65001[10] = Total measuring range (max 536870912)	
P65001[11] = Velocity measuring units (0, 1, 2, 3)	

Manufacturer specific parameters

P9 → G1_STW signal management (bit13 cyclical position request)

Unsigned16, Read / Write access parameter. Writing this parameter commands bit13 of the G1_STW word in telegrams 81, 82, 83 and 84.

P9 = 0 bit13 in G1_STW is controlled by PLC program
P9 = 0x2000 encoder works like bit13 in G1_STW is always 1

P80 → STW2_ENC signal management (bit10 control by PLC)

Unsigned16, Read / Write access parameter. Writing this parameter commands bit10 of the STW2_ENC word in telegrams 81, 82, 83 and 84.

P80 = 0 bit10 in STW2_ENC is controlled by PLC program
P80 = 0x0400 encoder works like bit10 in STW2_ENC is always 1

9 Functional description of the encoder

This chapter describes the functions implemented into ELAP PROFINET encoders:

- Code sequence
- Class 4 functionality
- G1_XIST1 Preset control
- Scaling function control
- Alarm channel control
- Compatibility mode
- Preset value
- Single turn resolution
- Total measuring range
- Maximum Master Sign-of-Life failures
- Velocity measuring units
- Encoder Profile Version
- Offset value

See also sections [8.6 Supported Parameters](#), [9.13 Acyclic Data](#) and [3.4 / 4.4](#) Setting the encoder parameters.

9.1 Code sequence

The code sequence defines the rotation direction, seen on the shaft, in which the position value increases. The default value is 0.

- 0 (CW) → Increasing position value with clockwise rotation (seen from the shaft side).
- 1 (CCW) → Increasing position value with counter clockwise rotation.

This parameter is only used if Class 4 functionality is active.

Note: The position value will be affected when the code sequence is changed during operation. It is recommended to carry out a preset after changing the code sequence.

9.2 Class 4 functionality

This parameter enables or disables the measuring task functions: Scaling, Preset and Code sequence.

If the function is enabled, Scaling and Code sequence control affects the position value in G1_XIST2 and G1_XIST3. A Preset operation in this case always affects G1_XIST2 and G1_XIST3; if the parameter G1_XIST1 Preset control is disabled, the preset will not affect the position value in G1_XIST1. The default value is 1.

- 0 → Scaling, Preset and Code sequence controls are disabled
- 1 → Scaling, Preset and Code sequence controls are enabled (default)

9.3 G1_XIST1 Preset control

This parameter controls the effect of a preset operation on the G1_XIST1 actual value.

If Class 4 functionality is activated and G1_XIST1 Preset control is disabled, the position value in G1_XIST1 will not be affected by a Preset operation. The default value is 1.

- 0 → G1_XIST1 is affected by a preset command
- 1 → Preset does not affect G1_XIST1 (default)

Note 1: The preset control is disabled by setting this parameter to 1.

Note 2: There is no functionality of this parameter if class 4 is disabled.

9.4 Scaling function control

This parameter enables or disables the Scaling function of the encoder. The default value is 0.

- 0 → Scaling function is disabled (default)
- 1 → Scaling function is enabled

Note: The parameter “Class 4 functionality” must be enabled to use this parameter.

9.5 Alarm channel control

This parameter enables or disables the encoder specific Alarm channel transferred as Channel Related Diagnosis. This functionality is used to limit the amount of data sent in isochronous mode. If the value is zero (default value) only the communication related alarms are sent via the alarm channel. If the value is one (1) also the encoder profile specific faults and warnings are sent via the alarm channel.

- 0 → No profile specific diagnosis is transmitted (default)
- 1 → The profile specific diagnosis is switched on

Note: This parameter is only supported in compatibility mode.

9.6 Compatibility mode

This parameter defines if the encoder operating mode has to be compatible with the Encoder Profile Version 3.1. The default value is 1.

- 0 → Compatibility with the Profile Encoder 3.1 is enabled
- 1 → Profile Encoder 4.1 is used; compatibility with previous versions is not enabled (default)

The table below shows all the functions affected by compatibility mode.

Function	Compatibility mode enabled (0)	Compatibility mode disabled (1)
Control by PLC (STW2_ENC, Bit 10)	Ignored, the control word (G1_STW) and the set point values are always valid. Control requested (ZSW2_ENC) is not supported and it is set to 0.	Supported.
Maximum Master Sign-of-Life failures	Supported.	Not supported. Only one failure is tolerated.
Alarm channel control	Supported.	Not supported. The application alarm channel is active and controlled by a PROFIdrive parameter.
P965[1] Profile identification number	31 (V3.1)	41 (v4.1)

9.7 Preset value

The preset value function enables to match the encoder position value to a known mechanical reference point of the system. The preset function sets the actual position of the encoder to zero (default value) or to the selected preset value. The preset function is controlled by bits in the control word (G1_STW) and acknowledged by a bit in the status word (G1_ZSW). A preset value can be set more than once in parameter P65000 and it can be stored into the non volatile memory using PROFIdrive parameter P971.

The preset function has an absolute and a relative operating mode selectable by bit 11 in the Control word (G1_STW). Bit 11 and bit 12 in the Control word handle the preset function in the following way:

- Normal operating mode: **Bit 12 = 0**
The encoder makes no change in the output value.
- Preset absolute mode: **Bit 11 = 0, Bit 12 = 1**
The encoder reads the current position value and calculates an internal offset between the preset and the position values. The position value is then shifted by the calculated offset to get a position value equal to the preset value. Negative values are not accepted in this case.

- Preset relative mode: **Bit 11 = 1, Bit 12 = 1**
The current position is shifted by the preset value, which could be either a negative or a positive value set by the encoder parameter P65000.

The steps below should be followed by the IO Controller when modifying the Preset value parameter:

1. Read the preset parameter P65000 and check if this value meets the application requirements. If not, proceed with the following steps.
2. Write the preset value into the individual parameter P65000.
3. Store the parameter into the non volatile memory by writing the parameter P971, in case the value must be used after the next power on sequence.

Note 1: The preset function should only be performed with standstill encoder.

Note 2: The number of possible preset cycles is unlimited.

9.8 Scaling function parameters

The scaling function converts the encoder physical absolute position value by software in order to change the resolution of the encoder. The scaling parameters will only be activated if the parameter Class 4 functionality and Scaling function control are enabled. The permissible value range for the scaling is limited by the resolution of the encoder. The scaling parameters are securely stored in the IO Controller and are reloaded into the encoder at each power-up.

The **Single turn resolution** parameter sets the number of different measuring steps during one revolution of the encoder. The physical single turn resolution value of ELAP encoder is 13 Bit; therefore the permissible value range is between 1 ($2^{\text{exp}0}$) and 8192 ($2^{\text{exp}13}$).

The **Total measuring range** of the encoder is calculated by multiplying the single turn resolution by the number of distinguishable revolutions. ELAP encoder features a global physical resolution of 29 Bit; therefore the permissible value range is between 1 and 536870912 ($2^{\text{exp}29}$).

That is:

single turn resolution x number of revolutions = $8192 (2^{\text{exp}13}) \times 65536 (2^{\text{exp}16}) = 536870912 (2^{\text{exp}29})$.

Note: After downloading new scaling parameters, the preset function must be used to set the encoder starting point to absolute position 0 or to any required starting position within the scaled operating range.

9.9 Maximum Master Sign-of-Life failures

This parameter defines the number of allowed failures of the master's sign of life.

All the values between 1 and 255 are allowed, the default value is 1.

Note: This parameter is only supported in compatibility mode. Otherwise, one failure only is tolerated.

9.10 Velocity measuring units

This parameter defines the coding of the velocity measuring units used to configure the signals NIST_A and NIST_B. Standard telegram 81 has no velocity information included and the encoder does not use the velocity unit information in this case. Telegrams 82, 83 and 84 include velocity output and need a declaration of the velocity measuring units. The following values are available:

- 0 → Steps / s
- 1 → Steps / 100 ms
- 2 → Steps / 10 ms
- 3 → RPM (default)

9.11 Encoder profile version

It is the version of the profile document implemented in the encoder. This parameter (P65001[6]) is not affected by the Compatibility mode settings.

Bit	Meaning
0... 7	Least significant number of version (from 0 to 99)
8... 15	Most significant number of version (from 0 to 99)
16... 31	Reserved

9.12 Offset value

The offset value is calculated in the preset function and shifts the position value by the calculated value. The offset value is stored into a non volatile memory and can be read by the encoder at any time (P65001[8]). The data type for the offset value is a 32-bit value with sign, whereby the offset value range is equal to the measuring range of the device. The preset function is used after the scaling function. This means that the offset value is indicated according to the scaled resolution of the device.

Note: The offset value is read only, and cannot be modified by a parameter writing access.

9.13 Acyclic Data

The PROFINET encoder supports the following acyclic data exchange functions, for PROFIdrive parameters defined in the Profile Encoder 3.162, V4.1.

The following parameters are included in the Record Data Object 0xB02E.

General parameters:

Parameter	Meaning	Data type	Access (R → read, W → write)
P922	Telegram selection	Unsigned16	R
P964	Device identification	Array[5] Unsigned16	R
P965	Encoder profile number	Octet String 2	R
P971	Transfer to non volatile memory	Unsigned16	W
P975	Encoder object identification	Array[8] Unsigned16	R
P979	Sensor format	Array[6] Unsigned32	R
P980	List of supported parameters	Array[n] Unsigned16	R

Encoder specific parameters:

Parameter	Meaning	Data type	Access (R → read, W → write)
P65000	Preset value	Integer32	R / W
P65001	Operating status	Array[12] Unsigned32	R

The parameter P65000 sets the value for the preset function. This value can be stored into non volatile memory by writing 1 in parameter P971, and then it is reload at power up.

The parameter P65001 is a read-only structure containing information on the Encoder operating status. It is a complement to the PROFIdrive parameter P979 described in the Profile for Drive Technology, PROFIdrive V4.1, Order nr 3.172 available from PROFIBUS and PROFINET International.

Sub index	Meaning
0	Header
1	Operating status
2	Faults
3	Supported faults
4	Warnings
5	Supported warnings
6	Encoder Profile version
7	Operating time
8	Offset value
9	Single turn resolution
10	Total measuring range
11	Velocity measuring units

Sub index 1: Operating status

The status of different encoder functions can be read out. The mapping of the respective functions is shown in the table below:

Bit	Definition
0	Code sequence
1	Class 4 functionality
2	G1_XIST1 Preset control
3	Scaling function control
4	Alarm channel control
5	Compatibility mode
6... 7	Reserved for the encoder manufacturer
8... 31	Reserved for future use

Sub index 2/3: Faults / Supported faults

Bit	Definition	Supported
0	Position error	Supported
1	Under voltage	
2	Over voltage	
3	Short circuit	
4	Commissioning diagnostics	
5	Memory access error	Supported
6... 31	Currently not assigned	

Sub index 4/5: Warnings / Supported warnings

Bit	Definition	Supported
0	Frequency exceeded	
1	Over temperature	
2	Light control reserve	
3	CPU Watchdog status	
4	Operating time limit warning	
5	Battery voltage low	Supported
6... 31	Currently not assigned	

Manufacturer specific parameters:

Parameter	Meaning	Data type	Access (R → read, W → write)
P9	Management of bit13 in G1_STW	Unsigned16	R / W
P80	Management of bit10 in STW2_ENC	Unsigned16	R / W

9.14 Identification and Maintenance (I&M Function)

In addition to the PROFIdrive parameter P964 - Device Identification, I&M functions are supported by the encoder. The I&M functions can be accessed at record index 0xAFF0. The following I&M functions are supported.

I&M parameter	Byte	Data type	Comment
MANUFACTURER_ID	2	Unsigned16	Manufacturer identification code (0x02AB for ELAP), assigned by PROFIBUS & PROFINET International (PI)
ORDER_ID	20	ASCII String	Contains the most relevant part of the order code
SERIAL_NUMBER	16	ASCII String	Contains the serial number of the encoder
HARDWARE_REVISION	2	Unsigned16	Hardware edition
SOFTWARE_REVISION	4	1 Char + 3 Unsigned8	Software edition ("V", 0x01, 0x00, 0x00 → V1.0.0)
REVISION_COUNTER	2	Unsigned16	Revision counter
PROFILE_ID	2	Unsigned16	Profile identifier (0x3D00 for encoders)
PROFILE_SPECIFIC_TYPE	2	Unsigned16	Contains the encoder type (0x0001 → Absolute Multi-turn Encoder)
IM_VERSION	2	2 Unsigned8	I&M 3.502 Document version (0x0200 → 2.0)
IM_SUPPORTED	2	Unsigned16 (Array of Bit)	Value = 0 → Only I&M0 block is supported

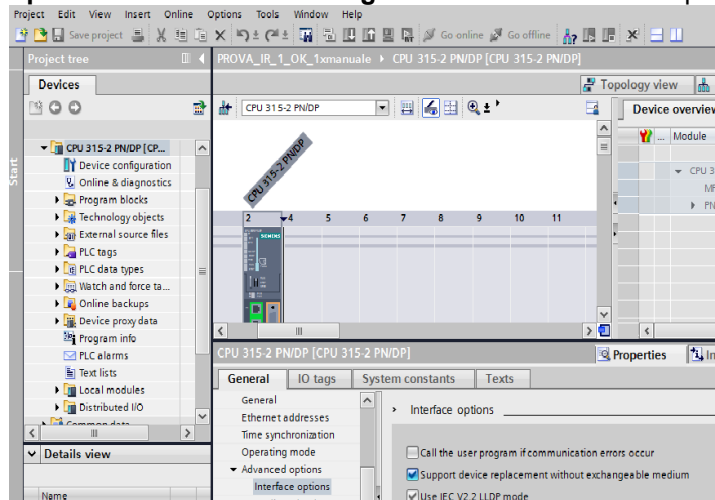
10 Encoder replacement using LLDP

ELAP encoder supports LLDP (Link Layer Discovery Protocol), that is Layer 2 Protocol, used to detect the closest neighbors in the network. It enables a device to send information about itself and to save information received from neighboring devices, i.e. it provides the option of communicating data between neighboring devices (e.g. device name, port, MAC address). This information allows a network management system to determine the network topology. The protocol is formally referred to by the IEEE as Station and Media Access Control Connectivity Discovery specified in standards document IEEE 802.1AB.

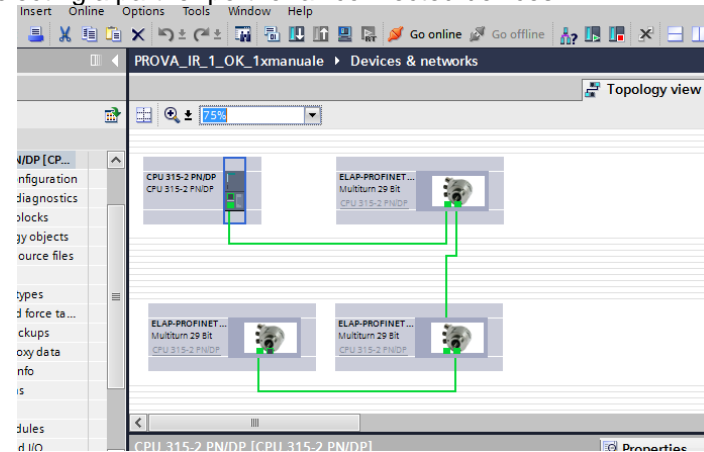
Among the main uses, LLDP allows to replace a device of the PROFINET network. The partner ports before and behind the replaced device save the relevant information so that no additional configuration is necessary.

10.1 Encoder replacement with TIA PORTAL V13

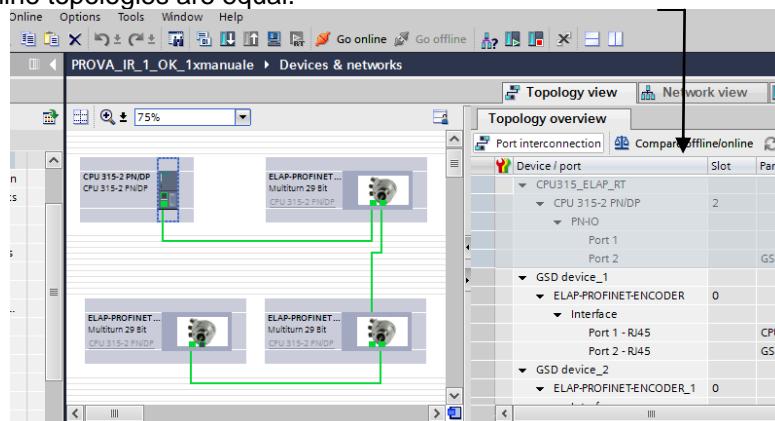
Enable “Support device replacement without exchangeable medium” in PN-IO properties.



Configure the topology, selecting a partner port for all connected devices.



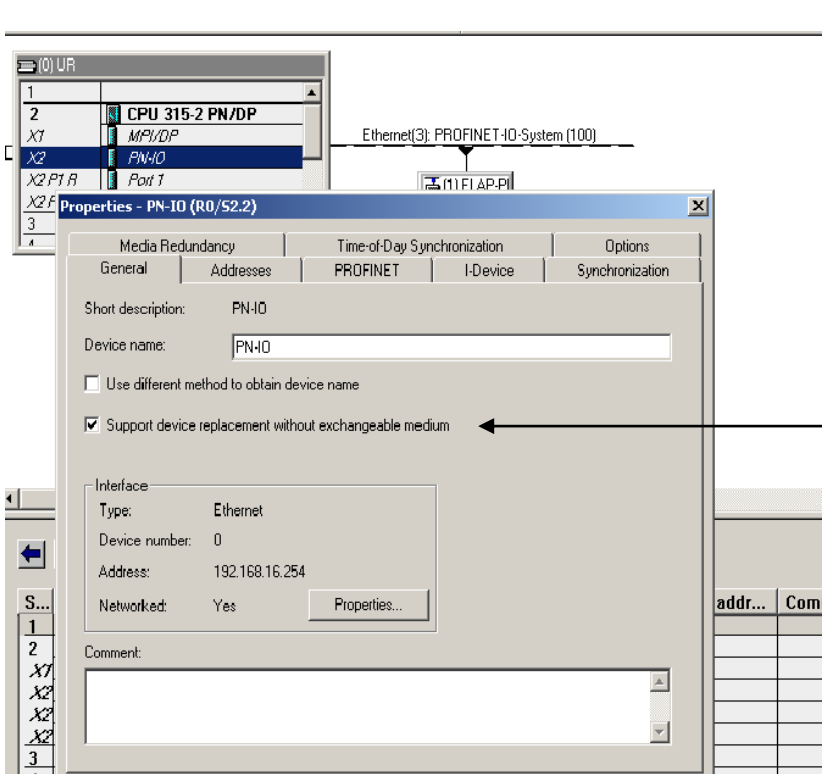
Check that Online / Offline topologies are equal.



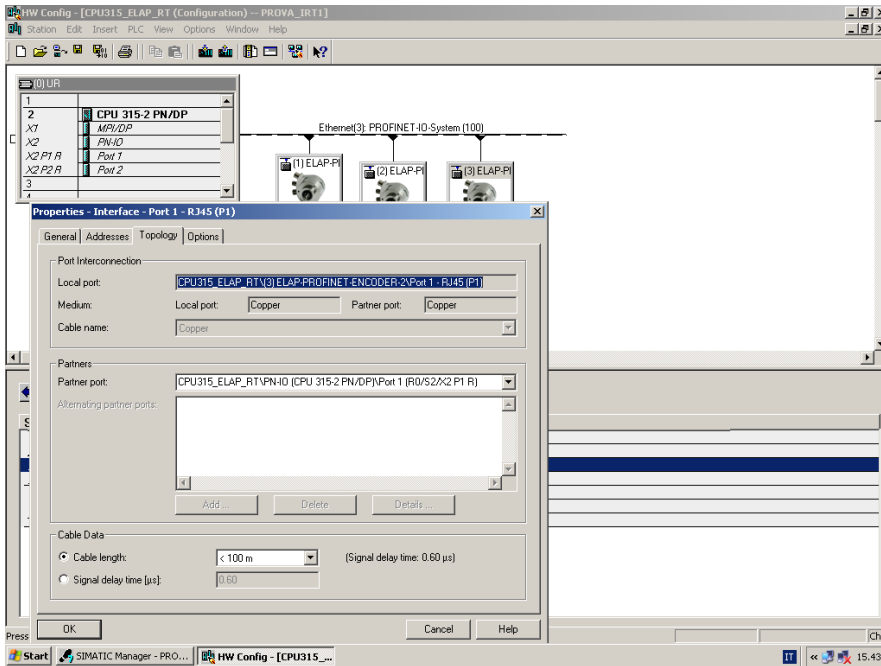
Note: When you replace an IO device (set to factory reset), make sure that the PROFINET cable is then inserted into the correct port as it is configured in the topology. Otherwise, the system will not run.

10.2 Encoder replacement with STEP7

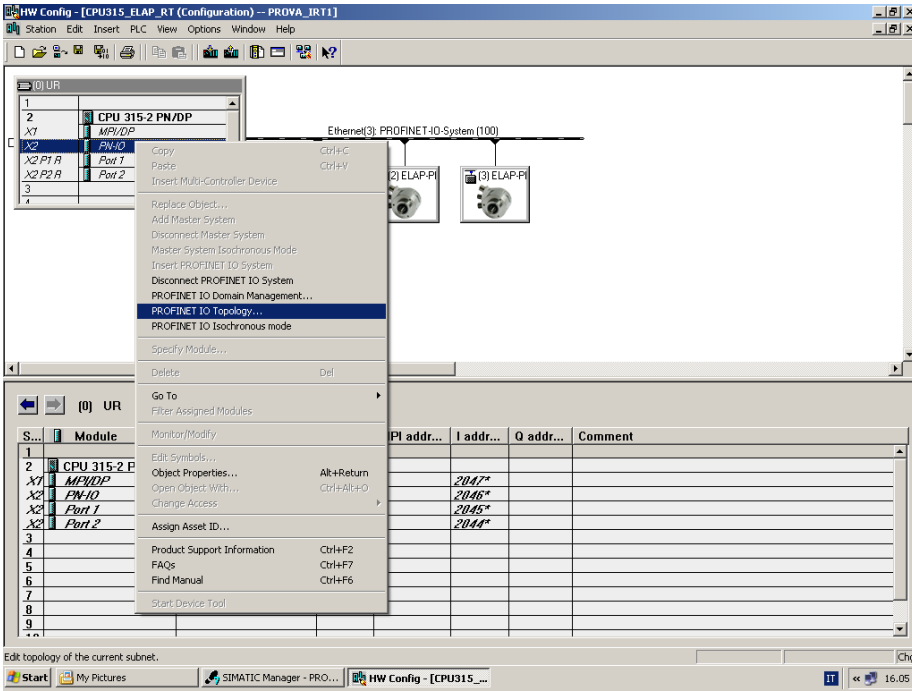
Enable **“Support device replacement without exchangeable medium”** in PN-IO properties.



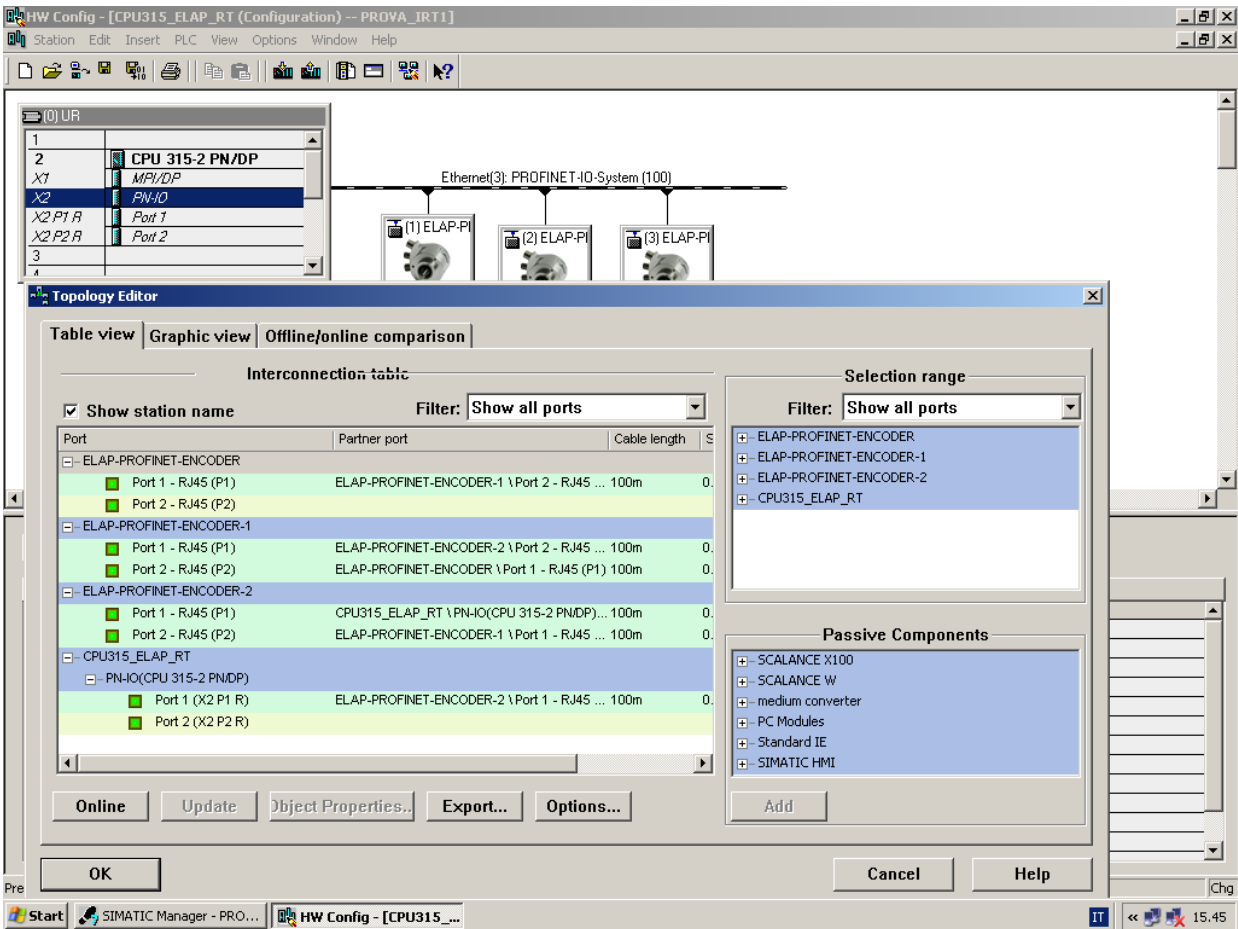
Configure the topology, selecting a partner port for all connected devices.



Now open the topology editor.



Check that Online / Offline topologies are equal.



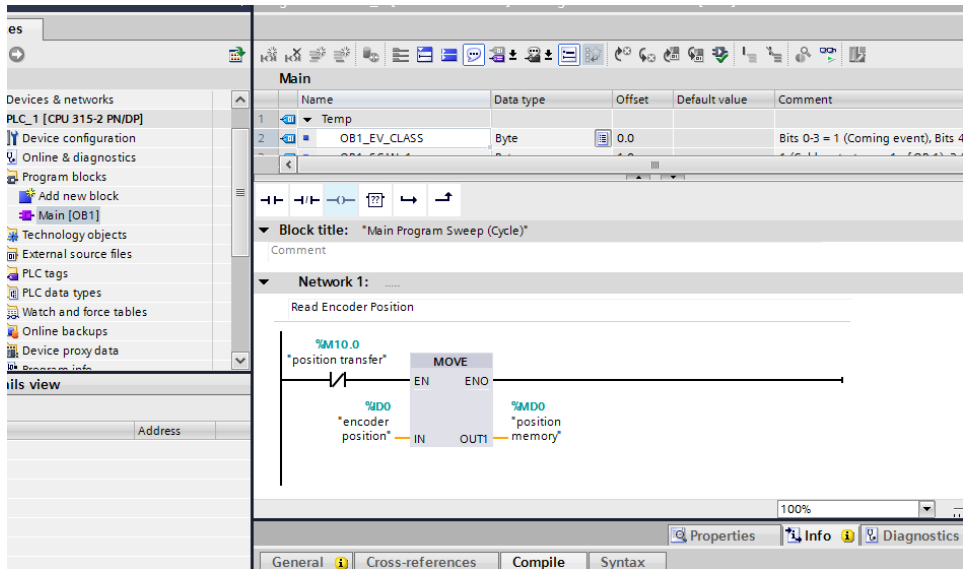
Note: When you replace an IO device (set to factory reset), make sure that the PROFINET cable is then inserted into the correct port as it is configured in the topology. Otherwise, the system will not run.

APPENDIX A: Reading position with TIA PORTAL V13

This example shows how to read the encoder position and store it in memory, with telegram 860. Experience with TIA PORTAL V13 programming tool and KOP programming language is required.

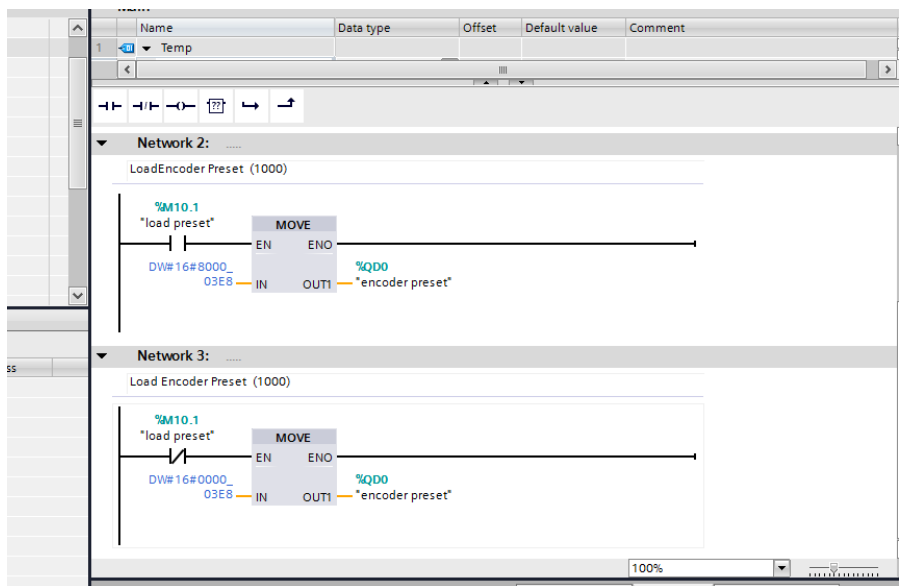
- Hardware components**
 IO Controller → SIEMENS CPU315-2PN/DP 315-2EH14-0AB0 V3.2
 IO Device → ELAP PROFINET Encoder
- Software components**
 SIEMENS TIA PORTAL V13
 GSDML file for ELAP PROFINET Encoder GSDML-V2.2-ELAP-MEM-BUS-xxxxxxx.XML

Instructions in MAIN Block :



In this section the encoder position is read and transferred into the Double-Word variable MD0 by the MOVE instruction. This operation is conditioned by bit M10.0 state.

The next section shows how to use the encoder PRESET function:



Setting the state of bit M10.0 to one, the immediate value 1000 will be transferred to the Encoder PRESET value (%QD0). For correct operation, reset the state of bit M10.0 to zero after the execution of two PLC cycles at least.

APPENDIX B: Reading / Writing parameters in record data block 0xB02E

The following examples require experience with TIA PORTAL or STEP7 programming tool and STL (Statement List) programming language.

- Hardware components**
 IO Controller → SIEMENS CPU315-2PN/DP, CPU1512SP-1 PN
 IO Device → ELAP PROFINET Encoder
- Software components**
 SIMATIC STEP7 V5.5 + SP3 / TIA PORTAL V13 (or (V15)
 GSDML file for ELAP PROFINET Encoder GSDML-V2.2-ELAP-MEM-BUS-xxxxxxx.XML
- Used Blocks**
 SFB53 WRREC → Special function block for writing a data record
 SFB52 RDREC → Special function block for reading a data record
 DB53, DB52 → Instance data blocks, assigned to SFB52 and SFB53 respectively
 DB1 → Data block for the reply message structure
 DB2 → Data block for the request message structure
 OB1, OB82 e OB86 → Organization blocks
 FC1, FC2 → User functions

1) Create the DB2 block (reply message structure)

with TIA-PORTAL V13

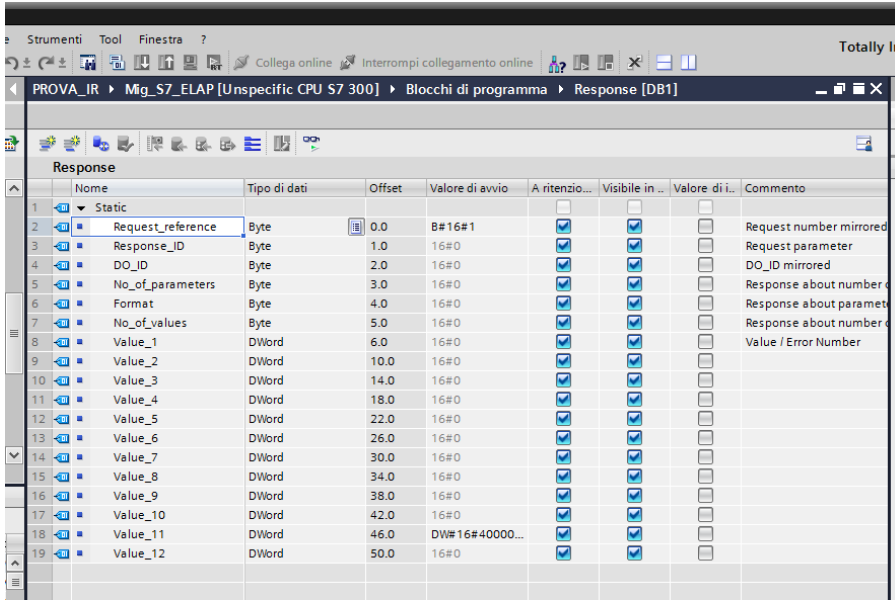
Nome	Tipo di...	Offset	Valore di av..	A ritenzio...	Visibile in HMI	Valore...	Commento
Static							
Request_reference	Byte	0.0	B#16#1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Request number
Request_ID	Byte	1.0	B#16#2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Req.param.->1; change param.->2
DO_ID	Byte	2.0	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Axis addressing for multiaxing drives
No_of_parameters	Byte	3.0	B#16#1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read/Write 1 parameter
Attribute	Byte	4.0	B#16#10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Attribute / Format
No_of_elements	Byte	5.0	B#16#1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Number of elements
Parameter_index	Word	6.0	W#16#FDE8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Parameter number
Subindex	Word	8.0	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sub-index
Format	Byte	10.0	B#16#4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data type: 41->byte, 42->word, 43->dword
No_of_values	Byte	11.0	B#16#1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Number of values = Number of elements
Value	DWord	12.0	DW#16#64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value / Error number

with STEP7

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Request_reference	BYTE	B#16#1	Request number
+1.0	Request_ID	BYTE	B#16#1	Req.param.->1; change param.->2
+2.0	DO_ID	BYTE	B#16#0	Axis addressing for multiaxing drives
+3.0	No_of_parameters	BYTE	B#16#1	Read/Write 1 parameter
+4.0	Attribute	BYTE	B#16#10	Attribute / Format
+5.0	No_of_elements	BYTE	B#16#1	Number of elements
+6.0	Parameter_index	WORD	W#16#FDE8	Parameter number
+8.0	Subindex	WORD	W#16#0	Sub-index
+10.0	Format	BYTE	B#16#43	Data type: 41->byte, 42->word, 43->dword
+11.0	No_of_values	BYTE	B#16#1	Number of values = Number of elements
+12.0	Value	DWORD	DW#16#0	Value / Error number
=16.0		END_STRUCT		

2) Create DB1 block (request message structure)

with TIA-PORTAL V13



with STEP7

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Request_reference	BYTE	B#16#0	Request number mirrored
+1.0	Response_ID	BYTE	B#16#0	Request parameter
+2.0	DO_ID	BYTE	B#16#0	DO_ID mirrored
+3.0	No_of_parameters	BYTE	B#16#0	Response about number of pa
+4.0	Format	BYTE	B#16#0	Response about parameter fo
+5.0	No_of_values	BYTE	B#16#0	Response about number of va.
+6.0	Value_1	DWORD	DW#16#0	Value / Error Number
+10.0	Value_2	DWORD	DW#16#0	
+14.0	Value_3	DWORD	DW#16#0	
+18.0	Value_4	DWORD	DW#16#0	
+22.0	Value_5	DWORD	DW#16#0	
+26.0	Value_6	DWORD	DW#16#0	

B.1 Writing Preset value (P65000)

We want to reset our preset value.

Let's write this FC1 function:

Network 1: Compile request to write parameter P65000 = 0

```

A      M 7.0                ; Write preset command
JNB   _001

L      1
T      DB2.Request_reference
L      2
T      DB2.Request_ID      ; 1 = read, 2 = write
L      0
T      DB2.DO_ID
L      1
T      DB2.No_of_parameters
L      B#16#10
T      DB2.Attribute
L      1
T      DB2.No_of_elements
L      W#16#FDE8          ; 0xFDE8 = 65000
T      DB2.Parameter_index
L      W#16#0
T      DB2.Subindex
L      B#16#43           ; 0x41 = byte, 0x42 = word, 0x43 = dword
T      DB2.Format
L      1
T      DB2.No_of_values
L      0
T      DB2.Value

_001: NOP 0

```

Network 2: Send write command

```

A      M 7.0
AN     M 7.3
AN     M 7.4
AN     M 7.5
AN     M 7.6
S      M 7.3

CALL  "WRREC" , DB53
REQ   :=M7.3
ID    :=W#16#7F6    ; from H/W configuration, "Module Access Point" (note 1)
INDEX :=W#16#B02E
LEN   :=16
DONE  :=M8.0
BUSY  :=M7.4
ERROR :=M9.1
STATUS:=MD34
RECORD:=P#DB2.DBX0.0 BYTE 16

A      M 7.4
R      M 7.3

```

Network 3: Read reply message

```
A      M 7.0
AN     M 7.3
AN     M 7.4
AN     M 7.5
AN     M 7.6
S      M 7.5
```

```
CALL  "RDREC" , DB52
REQ   :=M7.5
ID    :=W#16#7F6 ; from H/W configuration, "Module Access Point" (note 1)
INDEX :=W#16#B02E
MLEN  :=50
VALID :=M8.3
BUSY  :=M7.6
ERROR :=M9.0
STATUS:=MD30
LEN   :=MW28
RECORD:=P#DB1.DBX0.0 BYTE 50
```

```
A      M 7.6
R      M 7.5
```

NOTES

1. ID value in function blocks "WRRWC" and "RDREC" is CPU dependent.
For CPU315-2PN/DP, the *Module Access Point* of the first I/O device is 0x07F6, while for CPU1512SP-1 PN is 264. It is recommended to use symbolic address provided by TIA PORTAL. In this case, it is like "ELAP-1~EO_Encoder_Multiturn_1~Module_Access_Point".
2. RECORD value in function blocks "WRREC" and "RDREC" depends on CPU and programming tool.
TIA PORTAL allows programmers to use their data block symbolic names (RECORD:=DB2).

B.2 Writing parameters P9 and P80

We want to enable reading of position value in standard telegrams 81, 82, 83 and 84.

Let's write this FC2 function:

Network 1: Compile request to write parameter P9 = 0x2000

```

A      M 7.1                ; P9 write command
JNB   _001

L      2
T      DB2.Request_reference
L      2
T      DB2.Request_ID      ; 1 = read, 2 = write
L      0
T      DB2.DO_ID
L      1
T      DB2.No_of_parameters
L      B#16#10
T      DB2.Attribute
L      1
T      DB2.No_of_elements
L      9
T      DB2.Parameter_index
L      0
T      DB2.Subindex
L      B#16#42              ; 0x41 = byte, 0x42 = word, 0x43 = dword
T      DB2.Format
L      1
T      DB2.No_of_values
L      DW#16#2000_0000     ; High word of DB2.value
T      DB2.Value

_001: NOP 0

```

Network 1: Compile request to write parameter P80 = 0x0400

```

A      M 7.2                ; P80 write command
JNB   _002

L      3
T      DB2.Request_reference
L      2
T      DB2.Request_ID      ; 1 = read, 2 = write
L      0
T      DB2.DO_ID
L      1
T      DB2.No_of_parameters
L      B#16#10
T      DB2.Attribute
L      1
T      DB2.No_of_elements
L      80
T      DB2.Parameter_index
L      0
T      DB2.Subindex
L      B#16#42              ; 0x41 = byte, 0x42 = word, 0x43 = dword
T      DB2.Format
L      1
T      DB2.No_of_values
L      W#16#400_0000       ; parte alta della dword DB2.value
T      DB2.Value

_002: NOP 0

```

Network 3: Send write command

```

A (
A      M 7.1
O      M 7.2
)
AN     M 7.3
AN     M 7.4
AN     M 7.5
AN     M 7.6
S      M 7.3

CALL  "WRREC" , DB53
REQ   :=M7.3
ID    :=16#7F6      ; from H/W configuration, "Module Access Point" (note 1)
INDEX :=16#B02E
LEN   :=16
DONE  :=M8.0
BUSY  :=M7.4
ERROR :=M9.1
STATUS:=MD34
RECORD:=P#DB2.DBX0.0 BYTE 16

A      M 7.4
R      M 7.3

```

Network 4: Read reply message

```

A (
A      M 7.1
O      M 7.2
)
AN     M 7.3
AN     M 7.4
AN     M 7.5
AN     M 7.6
S      M 7.5

CALL  "RDREC" , DB52
REQ   :=M7.5
ID    :=W#16#7F6   ; from H/W configuration, "Module Access Point" (note 1)
INDEX :=W#16#B02E
MLEN  :=50
VALID :=M8.3
BUSY  :=M7.6
ERROR :=M9.0
STATUS:=MD30
LEN   :=MW28
RECORD:=P#DB1.DBX0.0 BYTE 50

A      M 7.6
R      M 7.5

```

NOTES

1. ID value in function blocks "WRRWC" and "RDREC" is CPU dependent.
For CPU315-2PN/DP, the *Module Access Point* of the first I/O device is 0x07F6, while for CPU1512SP-1 PN is 264. It is recommended to use symbolic address provided by TIA PORTAL. In this case, it is like "ELAP-1~EO_Encoder_Multiturn_1~Module_Access_Point".
2. RECORD value in function blocks "WRREC" and "RDREC" depends on CPU and programming tool.
TIA PORTAL allows programmers to use their data block symbolic names (RECORD:=DB2).

APPENDIX C: Using Telegram 81 in IRT mode

This example shows how to handle data with Telegram 81, in IRT mode, with object OB61. Experience with TIA PORTAL V13 or STEP7 programming tool and STL (Statement List) programming language is required.

- **Hardware components**
IO Controller → SIEMENS CPU315-2PN/DP 315-2EH14-0AB0 V3.2
IO Device → ELAP PROFINET Encoder
- **Software components**
TIA PORTAL V13 / SIMATIC STEP7 V5.5 + SP3
GSDML file for ELAP PROFINET Encoder GSDML-V2.2-ELAP-MEM-BUS-xxxxxxx.XML
- **Used Blocks**
SFC 126 → Special function, for reading peripheral inputs
SFC 127 → Special function, for writing peripheral outputs
OB61 → DP synchronous interrupt

Network 1: Reading peripheral inputs

```
CALL SFC 126
PART    :=B#16#1
RET_VAL:=MW300
FLADDR :=MW302

L      IW 0
T      "SZW2_ENC"
L      IW 2
T      "G1_SZW"

L      ID 4
T      "G1_XIST1"
L      ID 8
T      "G1_XIST2"
```

Network 2: Increasing controller Sign-of-Life counter (C-LS)

```
L      "SignOfLife"
L      1
+I
T      "SignOfLife"

L      "SignOfLife"
L      15
>I
=      M 20.0

A      M 20.0
JNB   _005

L      1
T      "SignOfLife"

_005: NOP    0
```

Network 3: Setting the Encoder Control Word

```
; C-LS count in bit 12 to 15.  
; bit10 → Control requested by PLC
```

```
A      "PLC_Control"  
=      M 200.2  
  
A      M 311.0  
=      M 200.4  
A      M 311.1  
=      M 200.5  
A      M 311.2  
=      M 200.6  
A      M 311.3  
=      M 200.7
```

Network 4: Setting the Sensor Control Word

```
; bit15 → Error acknowledgement  
; bit14 → Activate "park mode"  
; bit12 → Activate preset function  
; bit11 → Preset mode
```

```
A      "ACK_ERR"  
=      M 202.7  
  
A      "CMD_PARK"  
=      M 202.6  
  
A      "REQ_POS"  
=      M 202.5  
  
A      "CMD_PRESET"  
=      M 202.4  
=      M 120.7  
  
A      "PRESET_MODE"  
=      M 202.3
```

Network 5: Writing peripheral outputs

```
L      "STW2_ENC"  
T      QW 0  
  
L      "G1_STW1"  
T      QW 2  
  
CALL  SFC 127  
PART   :=B#16#1  
RET_VAL:=MW304  
FLADDR :=MW306
```

APPENDIX D: Reading / Writing configuration record 0xBF00

This example shows how to read and write the encoder configuration data (see paragraphs [3.4 / 4.4](#) and [chapter 9](#)) in PLC program. Experience with TIA PORTAL V13 or STEP7 programming tool and STL (Statement List) programming language is required.

- **Hardware components**
IO Controller → SIEMENS CPU315-2PN/DP 315-2EH14-0AB0 V3.2
IO Device → ELAP PROFINET Encoder
- **Software components**
TIA PORTAL V13 / SIMATIC STEP7 V5.5 + SP3
File GSDML per ELAP PROFINET Encoder GSDML-V2.2-ELAP-MEM-BUS-xxxxxxx.XML
- **Used blocks**
SFB53 WRREC → Special function block for writing a data record
SFB52 RDREC → Special function block for reading a data record
DB53, DB52 → Instance data blocks, assigned to SFB52 and SFB53 respectively
OB1, OB82 and OB86 → Organization blocks
FC1 → User function for read / write record 0xBF00 (configuration data)

Configuration data consist of 31 byte, as shown in the table below:

Byte offset	Name	Data type	Default value
0	Operation settings	Unsigned 8	0x26
1... 4	Reserved		
5... 8	Measuring units per revolution	Unsigned 32	0x00002000
9... 12	Reserved		
13... 16	Total measuring range	Unsigned 32	0x20000000
17	Tolerated sign of life faults	Unsigned 8	0x01
18	Velocity measuring units	Unsigned 8	0x03
19... 30	Reserved		

The operation settings consist of 8 bit, as shown in the table below:

Bit	Definition	Default value
0	Code sequence	0 (CW → clockwise)
1	Encoder class 4 functionality	1 (Enabled)
2	Preset affects XST1	1 (Disabled)
3	Scaling function control	0 (Disabled)
4	Alarm channel control	0 (Disabled)
5	Compatibility mode	1 (Profile version 4)
6... 7	Reserved	

In this example, we want to set the encoder scaling parameters at run-time:

- Measuring units per Revolution = 3600
- Total measuring range = 36000 (10 revolutions of 3600 pulses)
- We need to enable the scaling function control (bit 3), Operation settings = 0x2E

1) Reserve an area of 31 byte into the PLC memory:

```

CFG_Op_param           MB    319  BYTE
CFG_SigleTurnRes_0    MD    320  DWORD
CFG_SingleTurnRes_1   MD    324  DWORD
CFG_TotalRange_0      MD    328  DWORD
CFG_TotalRange_1      MD    332  DWORD
CFG_maxCLS_failures   MB    336  BYTE
CFG_velocityMu        MB    337  BYTE
    
```


2) Create FC1 function (handling the configuration record) :

Network 1: Request to read record 0xBF00

```

A      M 7.0
AN     M 7.1
AN     M 7.2
AN     M 8.1
AN     M 8.2
S      M 7.1

CALL  SFB 52 , DB52
  REQ   :=M7.1
  ID    :=DW#16#7F6
  INDEX :=W#16#BF00
  MLEN  :=20
  VALID :=M7.3
  BUSY  :=M7.2
  ERROR :=M7.4
  STATUS:=MD28
  LEN   :=MW32
  RECORD:=P#M 319.0 BYTE 20

A      M 7.2
R      M 7.1
S      M 8.0

```

Network 2: Setting new configuration data

```

A      M 8.0
JNB   _001

L      B#16#2E
T      "CFG_Op_param"           ;MB 319
L      3600
T      "CFG_SingleTurnRes_1"   ; MD 324
L      36000
T      "CFG_TotalRange_1"     ;MD 332

_001: NOP 0

```

Network 3: Request to write record 0xBF00

```

A      M 8.0
AN     M 8.1
AN     M 8.2
AN     M 7.1
AN     M 7.2
S      M 8.1

CALL  SFB 53 , DB53
  REQ   :=M8.1
  ID    :=DW#16#7F6
  INDEX :=W#16#BF00
  LEN   :=20
  DONE  :=M8.3
  BUSY  :=M8.2
  ERROR :=M8.4
  STATUS:=MD36
  RECORD:=P#M 319.0 BYTE 20

A      M 8.2
R      M 8.1
R      M 8.0

```

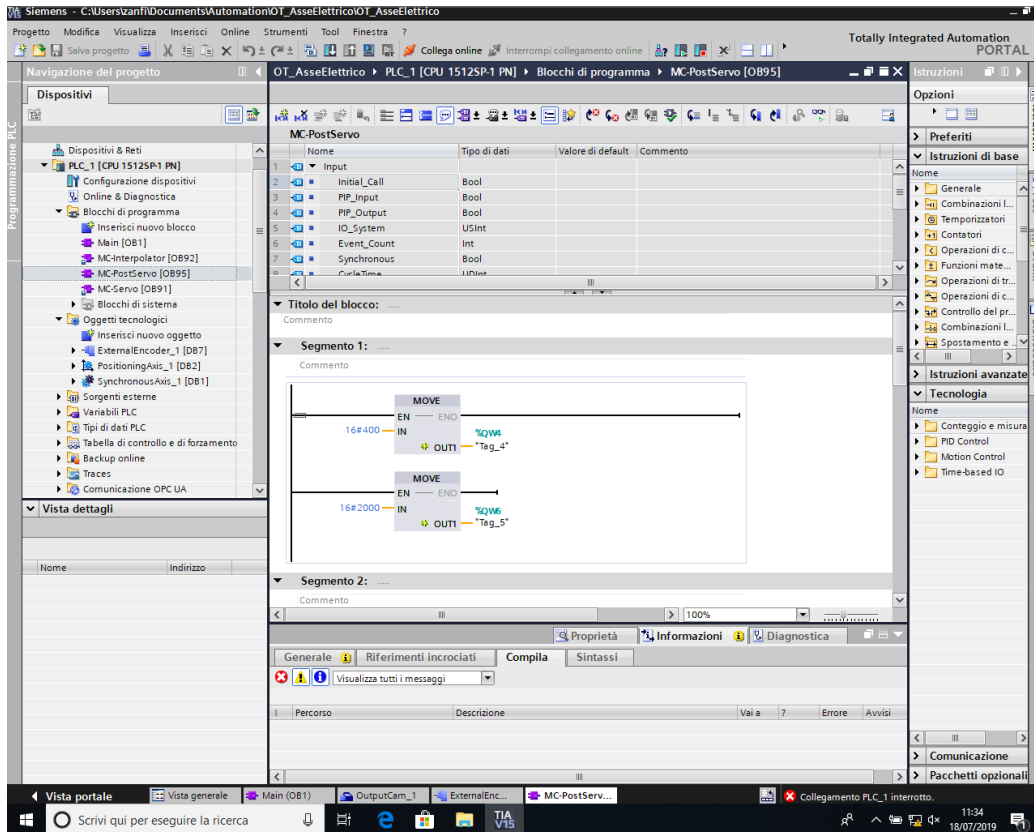
APPENDIX E: Operation with Technology Objects

The encoder can be used as Encoder Technology Object, or be employed among different Technology Objects such as positioning axis, synchronization axis and so on.

When one Siemens Technology Object is entered into the project, some of the bits of the encoder control words may be set to zero, thus preventing a correct operation.

In this case it is necessary to set bit 10 (PLC control) and bit 13 (cyclic position research) respectively in words STW2_ENC and G1_STW.

Two Move instructions must be entered into a previously configured MC-PostServo organization block in order to set the bit described above.



Check the correct addresses for the encoders to be managed according to the telegrams employed.

The example above employs MOVE instructions. Different instructions may also be used, provided they are aimed to set the mentioned bits in the respective control words.

Remark: Please, notice that the following parameters must be set to ZERO:

Technology Object setting page -> Hardware Interface -> Encoder Transmission Data

Bit in Gx_XIST1

Bit in Gx_XIST2

TECHNICAL SPECIFICATIONS

MECHANICAL & ENVIRONMENTAL SPECIFICATIONS			
		MEM-BUS 620 / 520 / 540	MEM-BUS 440 / 450
Materials:	case shaft	Aluminium Stainless steel	
Weight		500 g ca.	
Shaft/joint hole Ø		6, 8 ,10 mm	8, 10, 12, 14, 15 mm
Revolutions/minute		6000	
Starting torque		≤0.8 Ncm	
Inertia		≤25 g cm ²	
Max load		80 N axial/100 N radial	
Vibrations resistance	(10÷2000 Hz)	100 m/sec ²	
Shock	(11 ms)	50 G	
Protection degree		IP67 – IP65 shaft side	
Operating temperature		-30 ÷ 70°C	
Stocking temperature		-30 ÷ 85°C	

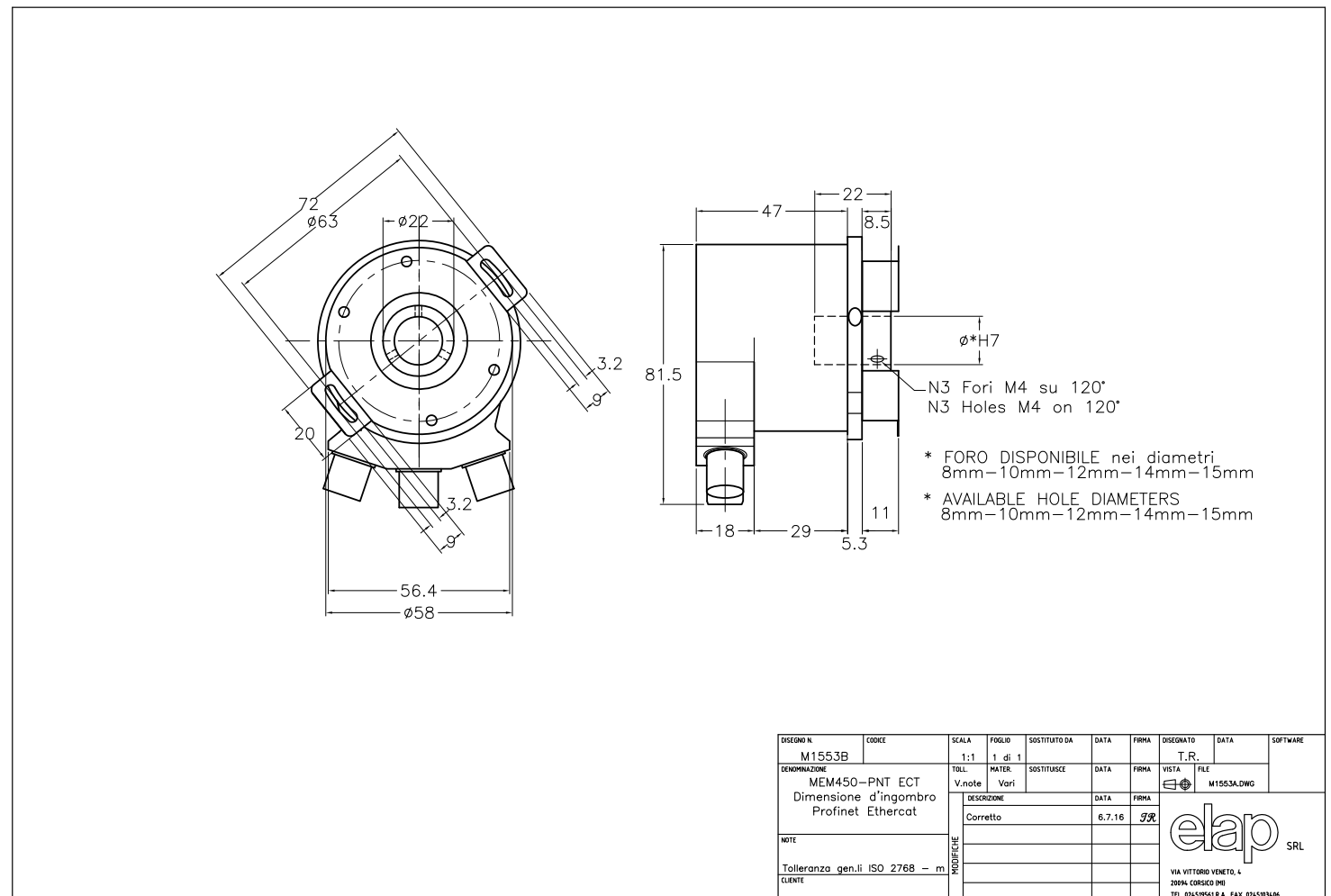
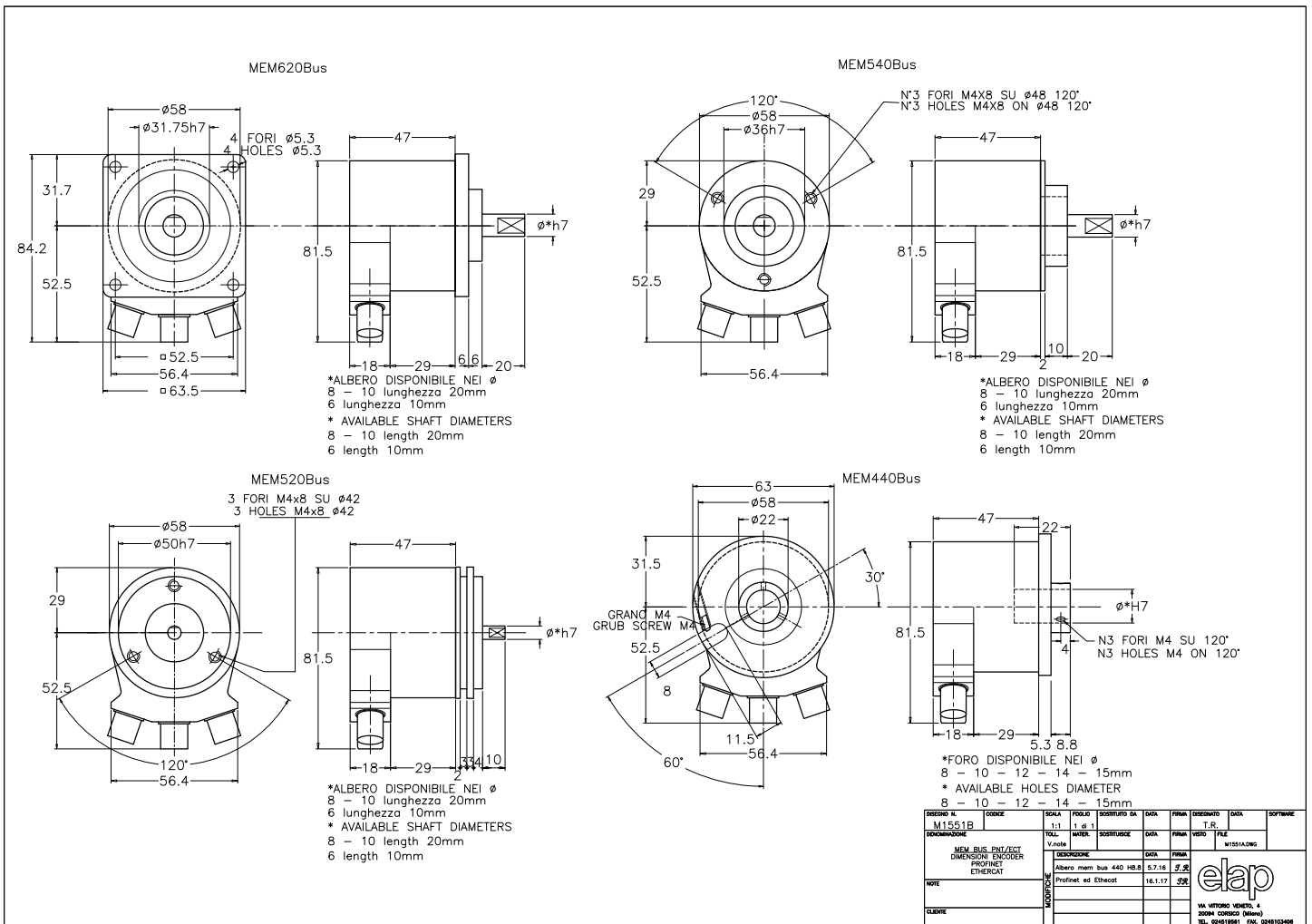
ELECTRICAL & OPERATING SPECIFICATIONS	
Operating principle	Magnetic
Resolution/revolution	8192 steps/rev – 13 bit
Revolutions no.	65536 - 16 bit
Initializing time	< 1 s
Data memory	>20 years No motion – Power off
Fieldbus	PROFINET
Supply	10 ÷ 30 Vdc Protection against polarity reversal
Power consumption	2 W
Accuracy	± ½ LSB
Connection	2 M12 female connectors +1 M12 male connector
Interference immunity	EN 61000-6-2
Emitted interference	EN61000-6-4

ORDERING INFORMATION

MEM520B	PNT	M	10	A
TYPE	INTERFACE	No. of TURNS M = Multiturn	SHAFT/HOLLOW SHAFT Ø 6 – 8 – 10- 12 – 14 – 15 mm	CONNECTORS POSITION - = Radial outlet A = Axial outlet

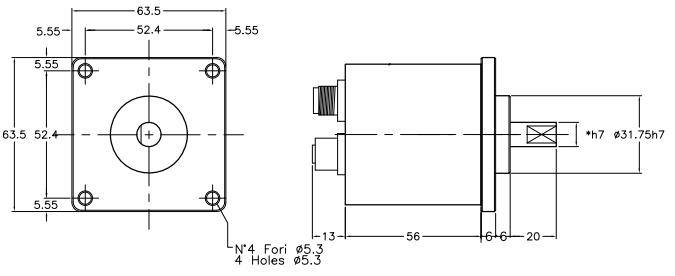
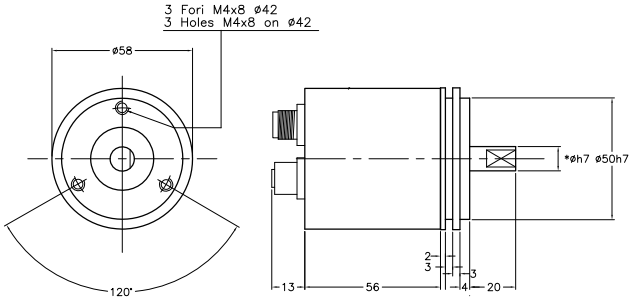
PNT = PROFINET
MEM520-Bus = Round flange Ø 58 mm
MEM540-Bus = Round flange Ø 58 mm
MEM620-Bus = Square flange 63.5x63.5 mm
MEM440-Bus = Blind hollow shaft for motor coupling
MEM450-Bus = Blind hollow shaft, fixing by elastic support

Elap reserves the right to upgrade the product without notice



MEM520B PNT/ETC M12 assiale

MEM620B PNT/ETC M12 assiale



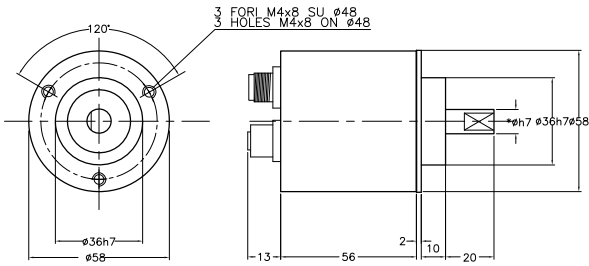
* ALBERO DISPONIBILE
Nei diametri 8mm-10mm
diametro 6mm lunghezza 10mm

* AVAILABLE SHAFT DIAMETERS
8mm-10mm
shaft diameter 6mm length 10mm

* ALBERO DISPONIBILE
Nei diametri 8mm-10mm
diametro 6mm lunghezza 10mm

* AVAILABLE SHAFT DIAMETERS
8mm-10mm
shaft diameter 6mm length 10mm

MEM540B PNT/ETC M12 assiale



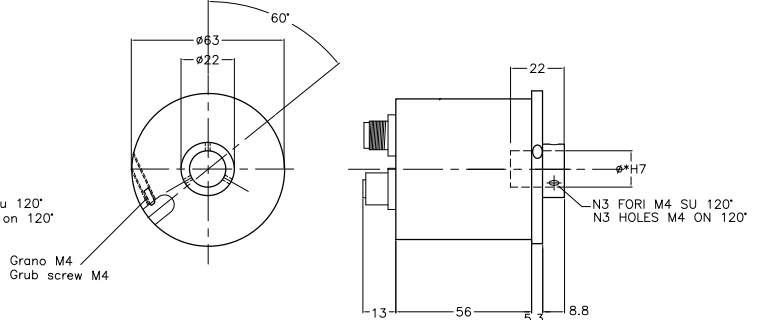
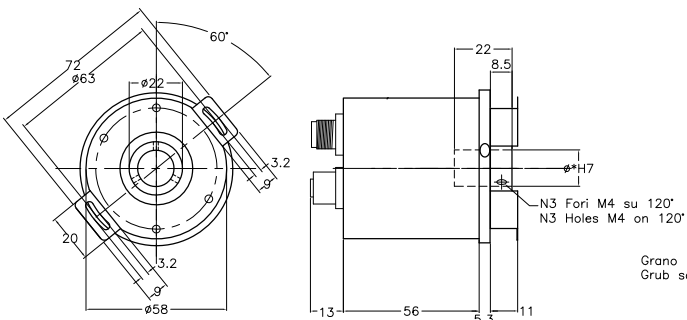
* ALBERO DISPONIBILE
Nei diametri 8mm-10mm
diametro 6mm lunghezza 10mm

* AVAILABLE SHAFT DIAMETERS
8mm-10mm
shaft diameter 6mm length 10mm

DISGNO N	CODICE	SCALA	FUSO	SOSTITUITO DA	DATA	FORMA	DISGNO	DATA	SOFTWARE
M2103		1:1	A-D				T.R.	24.4.19	
REVISIONE		DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
MEM-B PNT/ECT Dimensioni d'ingombro per uso catalogo uscita connettore M12 Assiale							<p>VIA VITTORIO VENETO, 4 30094 CORCHIO (TREVISO) TEL. 042319361 FAX 042319366</p>		
NOTE									
Tolleranze gen. II ISO 2768 - mS									
CLIENTE									

MEM450B uscita connettore M12 ASSIALE
MEM450B axial M12 connector outlet

MEM440B uscita connettore M12 ASSIALE
MEM440B axial M12 connector outlet



* FORO DISPONIBILE nei diametri
8mm-10mm-12mm-14mm-15mm

* AVAILABLE HOLE DIAMETERS
8mm-10mm-12mm-14mm-15mm

* FORO DISPONIBILE nei diametri
8mm-10mm-12mm-14mm-15mm

* AVAILABLE HOLE DIAMETERS
8mm-10mm-12mm-14mm-15mm

DISGNO N	CODICE	SCALA	FUSO	SOSTITUITO DA	DATA	FORMA	DISGNO	DATA	SOFTWARE
M2108		1:1	F e I				T.R.	14.5.19	
REVISIONE		DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
MEM 440B/450B dimensioni d'ingombro enc. serie MEM440B e 450B con usc. conn M12 assiale							<p>VIA VITTORIO VENETO, 4 30094 CORCHIO (TREVISO) TEL. 042319361 FAX 042319366</p>		
NOTE									
CLIENTE									

REFERENCES

MANUALS, SOFTWARE and DIMENSIONAL DRAWINGS can be downloaded at:

<https://www.elap.it/absolute-encoders/encoder-mem-bus-profinet/>



ELAP VIA VITTORIO VENETO, 4 • I-20094 CORSICO (MI) • TEL. +39.02.4519561
FAX +39.02.45103406 • E-MAIL INFO@ELAP.IT • SITE WWW.ELAP.IT