

# Manual

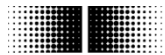
## **Absolute encoders EAx with EtherNet/IP interface**

Firmware version 1.003 and later

<b>Contents</b>	<b>Page</b>
<b>1. Introduction</b>	<b>6</b>
1.1. Scope of delivery	6
1.2. Product classification	6
<b>2. Safety and operating instructions</b>	<b>7</b>
<b>3. Commissioning</b>	<b>8</b>
3.1. Mechanical mounting	8
3.2. Electrical connection	8
3.2.1. Cabling	8
3.2.2. Connecting	9
3.2.3. Rotary Switches	11
<b>4. IP Address Configuration</b>	<b>12</b>
4.1. Via BOOTP/DHCP Server	13
4.2. Via Rotary Switches	14
4.2.1. Position of rotary switches equal to 0	15
4.2.2. Position of rotary switches unequal to 0	15
<b>5. Engineering (Allen Bradley Studio 5000)</b>	<b>16</b>
5.1. Start Studio 5000	16
5.2. Create project	16
5.3. Import EDS file	16
5.4. Module configuration	18
5.4.1. Configuring an encoder module	18
5.4.2. Set network path and download configuration	20
5.4.3. View encoder data using controller tags	20
5.5. Module Configuration, Connections	21
5.6. Input Data	21
5.7. Configuration data	22
5.8. Module Configuration with configuration data	23
5.9. Programming example (preset)	24
<b>6. EtherNet/IP cyclic operation</b>	<b>31</b>
6.1. PLC (EtherNet/IP scanner)	31
6.2. Basics	31
6.3. Diagnostic LEDs	32
6.3.1. Link/Activity (L/A) LEDs	32
6.3.2. Network Status Indicator	32
6.3.3. Module Status Indicator	33
6.4. Mapping of process data	33
6.5. Speed	34
6.5.1. Speed measuring unit	34
6.5.2. Speed update period	34
6.5.3. Speed filter depth	34
6.6. Preset function	35
6.6.1. Preset via Position Sensor Object	35
6.6.2. Preset using the push button	35
6.7. Cycle time and supported functionality	35
<b>7. EtherNet/IP acyclic operation</b>	<b>36</b>
7.1. Introduction	36
7.2. Device Type 0x22 (encoder profile)	36
7.3. CIP Object Model	36
7.4. Standard objects	36
7.5. Profile-specific objects	36
7.6. Parameterization	37
7.6.1. Measuring units per revolution	37
7.6.2. Total measuring range	37
7.6.3. Code sequence	37



7.6.4.	Scaling functionality	37
7.6.5.	Speed measuring unit	37
7.6.6.	Speed update period	37
7.6.7.	Speed filter depth	37
7.6.8.	Gear factor: activation	38
7.6.9.	Gear factor: numerator	39
7.6.10.	Gear factor: denominator	39
7.6.11.	Gear factor: parameterization	39
7.6.12.	Important note for multiturn encoder operation	40
7.6.13.	Preset	40
7.6.14.	Parameterization sequence	40
7.6.15.	Parameterization behaviour	41
<b>7.7.</b>	<b>Error Handling</b>	<b>42</b>
7.7.1.	Warning and alarm bits	42
7.7.2.	Mapping table	42
7.7.3.	Mapping of warning bits to device status	43
7.7.4.	Mapping of alarm bits to device status	43
7.7.5.	Device Status Prioritization	43
<b>7.8.</b>	<b>Firmware update via webserver</b>	<b>43</b>
<b>7.9.</b>	<b>Identity Object (0x01)</b>	<b>44</b>
7.9.1.	Class Services	44
7.9.2.	Class Attributes	45
7.9.3.	Instance Services	45
7.9.4.	Instance Attributes	46
<b>7.10.</b>	<b>Message Router Object (0x02)</b>	<b>48</b>
<b>7.11.</b>	<b>Assembly Object (0x04)</b>	<b>48</b>
7.11.1.	Class Services	48
7.11.2.	Class Attributes	48
7.11.3.	Instance Services	49
7.11.4.	Instance Attributes	49
7.11.5.	Instances	49
<b>7.12.</b>	<b>Connection Manager Object (0x06)</b>	<b>54</b>
7.12.1.	Class Services	54
7.12.2.	Class Attributes	54
7.12.3.	Instance Services	54
7.12.4.	Instance Attributes	54
<b>7.13.</b>	<b>Position Sensor Object (0x23)</b>	<b>55</b>
7.13.1.	Class Services	55
7.13.2.	Class Attributes	56
7.13.3.	Instance Services	56
7.13.4.	Instance Attributes	57
7.13.5.	Error codes (Position Sensor Object)	68
<b>7.14.</b>	<b>Device Level Ring Object (0x47)</b>	<b>70</b>
7.14.1.	Class Services	70
7.14.2.	Class Attributes	70
7.14.3.	Instance Services	70
7.14.4.	Instance Attributes	71
<b>7.15.</b>	<b>Quality of Service Object (0x48)</b>	<b>72</b>
7.15.1.	Class Services	72
7.15.2.	Class Attributes	72
7.15.3.	Instance Services	72
7.15.4.	Instance Attributes	73
<b>7.16.</b>	<b>TCP/IP Interface Object (0xF5)</b>	<b>74</b>
7.16.1.	Class Services	74
7.16.2.	Class Attributes	74
7.16.3.	Instance Services	74
7.16.4.	Instance Attributes	75
<b>7.17.</b>	<b>Ethernet Link Object (0xF6)</b>	<b>78</b>
7.17.1.	Class Services	78



7.17.2. Class Attributes	78
7.17.3. Instance Services	79
7.17.4. Instance Attributes	79
<b>8. Troubleshooting – Frequently Asked Questions – FAQ</b>	<b>80</b>
<b>8.1. FAQ: Project Work</b>	<b>80</b>
8.1.1. Where do I get an encoder manual?	80
8.1.2. Where do I get the applicable EDS file?	80
<b>8.2. FAQ: Operation</b>	<b>80</b>
8.2.1. What is the significance of the LEDs provided at the encoder?	80
8.2.2. How to adapt the resolution?	80
<b>9. Appendix A</b>	<b>81</b>
<b>9.1. Software Change Log (from firmware V1.003 to V1.004)</b>	<b>81</b>

### Disclaimer of liability

The present manual was compiled with utmost care, errors and omissions reserved. For this reason Baumer rejects any liability for the information compiled in the present manual. Baumer nor the author will accept any liability for direct or indirect damages resulting from the use of the present information.

At any time we should be pleased receiving your comments and proposals for further improvement of the present manual.

Created by:  
Baumer IVO GmbH & Co. KG  
Villingen-Schwenningen, Germany

### Registered trademarks

Studio5000® and LogixDesigner® are registered trademarks of Allen-Bradley®.

Names mentioned in the present manual and other names that may be registered trademarks are not marked correspondingly. Having omitted the respective marking does not necessarily imply that the names are not registered trademarks or that there are no existing patents and protected patented designs.

### Change history

This document is subject to changes. The latest version is available at [www.baumer.com](http://www.baumer.com).

Document index	Date	Firmware revision	Author	Changes
0001	09.18	1.003	div.	Initial version (replaces all draft documents)
0002	02.20	1.004	div.	Important changes: editorial changes and revision of following chapters: - 5.4.1 ("Configuring an encoder module") - 5.7 ("Configuration data") - 6.3 ("Diagnostic LEDs") - 6.5 ("Speed") - 7.6.8 to 7.6.11 ("gear factor") - 7.6.12 ("Important note for multiturn encoder operation") - 7.13 ("Position Sensor Object (0x23)") - 7.13.4 ("Instance Attributes") - 7.13.5 ("Error codes (Position Sensor Object)")  New chapters: - 7.6.15 ("Parameterization behaviour") - 9 ("Appendix A")

## 1. Introduction

### 1.1. Scope of delivery

Please check the delivery upon completeness prior to commissioning.

Depending on encoder configuration and part number the delivery may include:

- EtherNet/IP encoder
- EDS files and manual (available at [www.baumer.com](http://www.baumer.com))

### 1.2. Product classification

Product	Product family	Suitable input in hardware-catalog
EAL580-xxx.xxEN.13160.x	optical - multiturn	EAL580 MT encoder ST13 MT16, optical
EAL580-xxx.xxEN.18130.x	optical - multiturn	EAL580 MT encoder ST18 MT13, optical
EAM580-xxx.xxEN.14160.x	magnetic - multiturn	EAM580 MT encoder ST14 MT16, magnetic

#### Explanation

- Placeholders marked “x” in the product reference number will not influence the selection
- “MT” means “multiturn”
- “ST” means “singleturn”
- “ST13 MT16” means “13 bits physical singleturn resolution, 16 bits physical multiturn resolution”

#### EDS file

Several EDS files for different encoder types exist. The EDS files for an EAx580 EtherNet/IP encoder are available for download at [www.baumer.com](http://www.baumer.com).

EDS files for the following products are available:

- Baumer EAL 580 ST13MT16
- Baumer EAL 580 ST18MT13
- Baumer EAM 580 ST14MT16

For further information see chapter [Import EDS File](#).

#### Supported Standards and Protocols

Common Industrial Protocol (CIP™) Vol1, Ed 3.21

EtherNet/IP Adaptation of CIP Vol2, Ed. 1.22, Nov 2016

Encoder Device Profile (Volume 1, 6.21)

ODVA Certification Test CT14 passed with firmware revision 1.002.

Firmware revisions 1.003 and later support Apply\_Attributes, Restore and Save services for the Position Sensor Object to make the configuration of the device more comfortable.

## 2. Safety and operating instructions

### Intended use

- The encoder is a precision measuring device that is used to record positions and speeds. It provides measuring values as electronic output signals for the subsequently connected device. It must not be used for any other purpose. Unless this product is specially labeled, it may not be used for operation in potentially explosive environments.
- Make sure by appropriate safety measures, that in case of error or failure of the encoder, no danger to persons or damage to the system or operating facilities occurs.

### Personnel qualification

- Installation and assembly of this product may be performed only by a person qualified in electronics and precision mechanics.

### Maintenance

- The encoder is maintenance-free and must not be opened up nor mechanically or electronically modified. Opening up the encoder can lead to injury.

### Disposal

- The encoder contains electronic components. At its disposal, local environmental guidelines must be followed.

### Mounting

- Solid shaft: Do not connect encoder shaft and drive shaft rigidly. Connect drive and encoder shaft with a suitable coupling.
- Hollow shaft: Open clamping ring completely before mounting the encoder. Foreign objects must be kept at a sufficient distance from the stator coupling. The stator coupling is not allowed to have any contact to the encoder or the machine except at the mounting points.

### Electrical commissioning

- Do not proceed any electrical modifications at the encoder.
- Do not proceed any wiring work while encoder is live.
- Do not remove or plug on connector whilst under power supply.
- Ensure that the entire system is installed in line with EMC/EMI requirements. Operating environment and wiring have an impact on the electromagnetic compatibility of the encoder. Install encoder and supply cables separately or far away from sources with high emitted interference (frequency converters, contactors, etc.).
- When working with consumers with high emitted interference provide separate encoder supply voltage.
- Completely shield encoder housing and connecting cables.
- Connect encoder to protective earth (PE) using shielded cables. The braided shield must be connected to the cable gland or connector. Ideally, aim at dual connection to protective earth (PE), i.e. housing by mechanical assembly and cable shield by the downstream devices.

### Supplementary information

- The present manual is intended as a supplement to already existing documentation (e.g. catalogues, data sheets or mounting instructions).

### **3. Commissioning**

#### **3.1. Mechanical mounting**

##### **Shaft encoders**

- Mount the encoder using the mounting holes in the encoder flange and fitting screws. Observe thread diameter and depth.
- There is an alternative mounting option in any angular position by eccentric fixings available as an accessory.
- Connect drive shaft and encoder shaft by using an appropriate coupling. The shaft ends must not touch each other. The coupling must compensate temperature and mechanical tolerances. Observe the maximum permitted axial or radial shaft load. For appropriate couplings please refer to accessories.
- Tighten the mounting screws firmly.

##### **Hollow shaft encoders**

- Mounting by clamping ring  
Prior to mounting the encoder open the clamping ring completely. Push encoder onto the drive shaft and tighten the clamping ring firmly.
- Adjusting element with rubber buffer  
Push the encoder onto the drive shaft and insert the cylindrical pin into the adjusting element (customer-mounted) and the rubber buffer.
- Spring washer  
Fasten the spring washer at the mounting holes of the encoder housing using screws. Push the encoder onto the drive shaft and mount the spring washer to the contact surface.

#### **3.2. Electrical connection**

##### **3.2.1. Cabling**

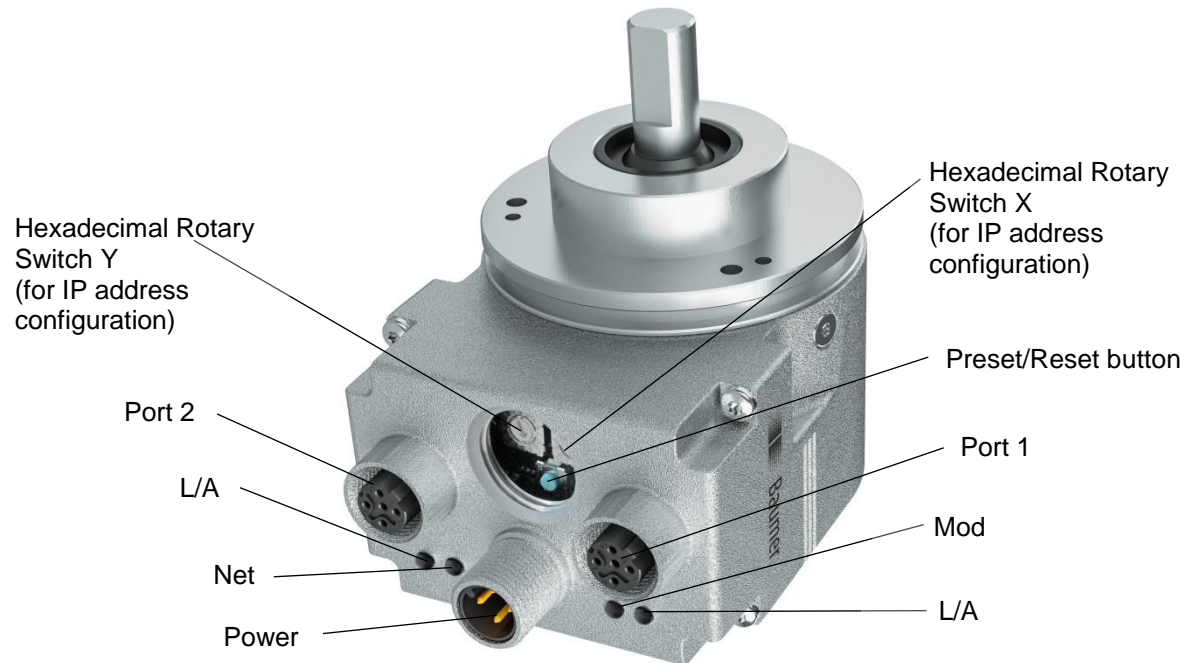
EtherNet/IP utilizes Fast Ethernet cables (100 Mbit/s, Cat. 5).



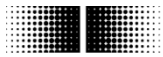
### 3.2.2. Connecting

The encoder provides three M12 flange connectors.

Two M12 flange connectors (D-coding, according IEC 61076-2-101) named "Port 1" and "Port 2" (see below) serve for the connection.

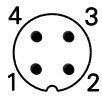


- For voltage supply use an A-coded M12 connector only.
- The D-coded M12 connector "Port 1" and "Port 2" may only be used to connect the encoder to the EtherNet/IP network.
- Seal up any unused M12 connector using a screw cap (included in the delivery).



## Pin assignment

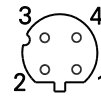
### Voltage supply



1 x M12 flange connector (male)  
A-coded

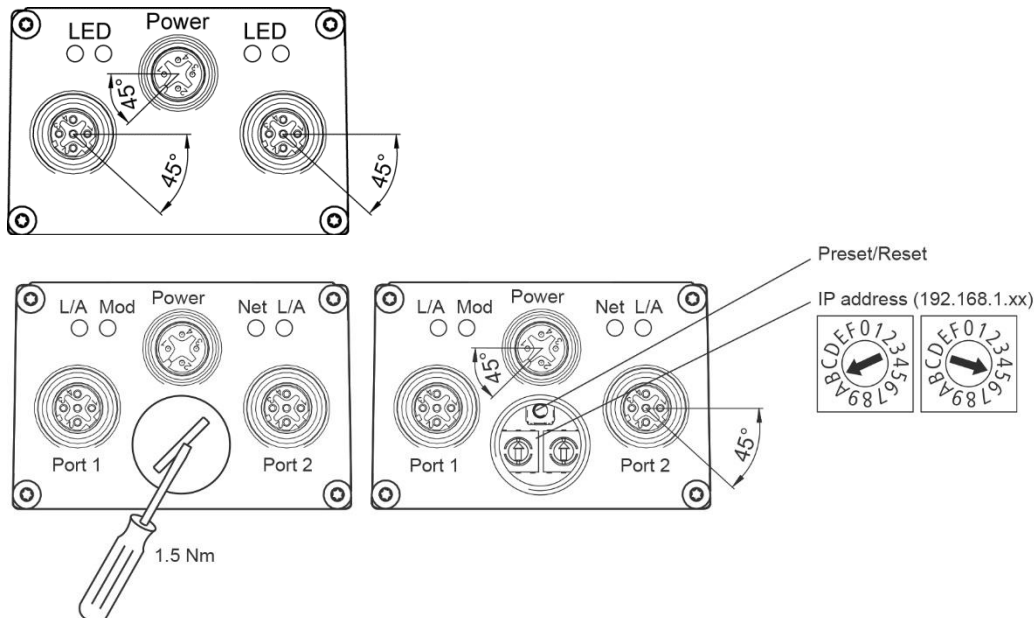
Pin	Assignment
1	UB / +Vs (10...30 VDC)
2	Do not connect
3	GND / 0 V
4	Do not connect

### EtherNet/IP (data line)



2 x M12 flange connector (female)  
D-coded

Pin	Assignment
1	TxD+
2	RxD+
3	TxD-
4	RxD-



### Preset/Reset button

The encoder may have a screw cap located where M12 connectors and LEDs are located as well. After removing the screw cap the preset/reset button is visible.

The preset/reset button works in two modes:

**Reset Mode:** effective in the first 60 seconds after power-on

Pressing the button results in a "Type 1 Reset". The encoder is reset to factory defaults. For details see chapter [Class Services](#) of the Identity Object.

Caution: Even communication parameters (e.g. IP address) are reset to factory defaults.

**Preset Mode:** effective after more than 60 seconds after power-on

Pressing the button activates a preset with a value defined in [Instance Attribute 107 - Preset Request Value](#) of the Position Sensor Object (value is 0 by default). The contained position value will be interpreted as absolute value.

In order to perform a preset/reset the push button must be pressed for a duration of at least three seconds (LEDs blinking red) and longest five seconds. The button has to be released when the LEDs change from red blinking to green blinking.



**Note:**

After having used the preset/reset button the screw cap must be screwed in again and tightened with a torque of **1.5 Nm**.

### **3.2.3. Rotary Switches**

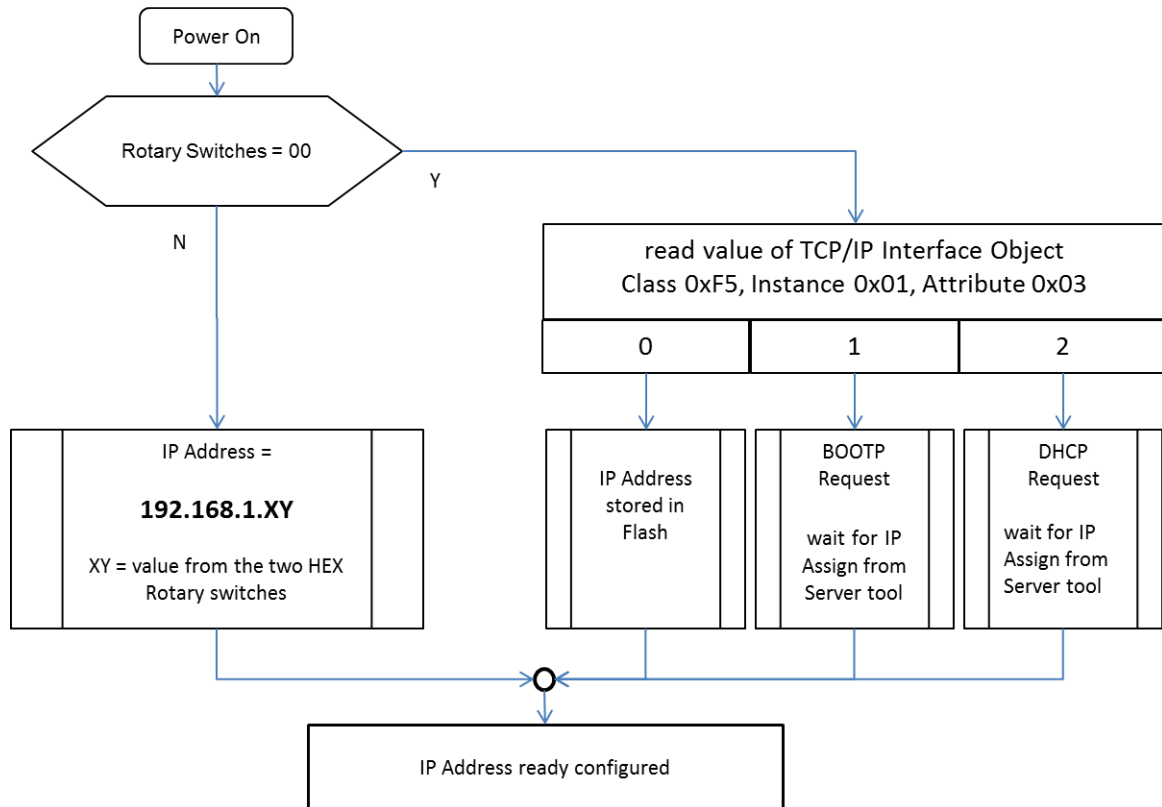
With the two rotary switches the address can be set to **192.168.1.xy** where x and y are defined by the hexadecimal values set by the rotary switches.

For detailed information see chapter [Via Rotary Switches](#).



## 4. IP Address Configuration

The IP address of the device has to be allocated before a communication with the device is possible. By default the device expects to receive an IP address via DHCP. The methods to allocate an IP address to the device are described below.

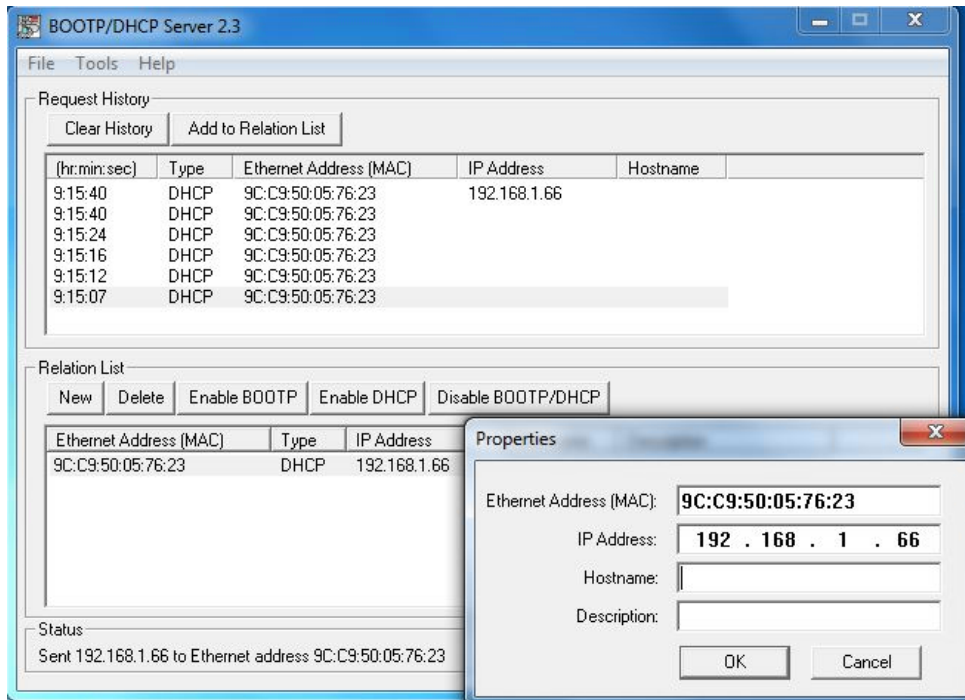




## 4.1. Via BOOTP/DHCP Server

Open the BOOTP/DHCP Server tool from Rockwell (tested with version 2.3). At power-on the device sends a request with its MAC address.

Double-click on the MAC address of the device and enter the desired IP address.



If a static IP address is desired click on "Disable BOOTP/DHCP" once the relation list entry is done.

## 4.2. Via Rotary Switches

If the value read from the rotary switches is unequal to 00 the IP address is "192.168.1.XY". The value of XY is calculated as follows.

X	Y
Value of rotary switch 1	Value of rotary switch 2

Open the screw cap to get access to the rotary switches. Set the rotary switches to the desired IP address. An example is shown below.



Generally the formula for the value of the 4th octet (byte) of the IP address is:

$$X * 16 + Y$$

In this example the rotary switches are set to X=0x04 and Y=0x02 which results to:

$$4 * 16 + 2 = 66 \rightarrow \text{The IP address is set to } 192.168.1.66.$$

In the following chapters the details about the functionality of the device depending on the value of the rotary switches are described.

#### 4.2.1. Position of rotary switches equal to 0

If the value of the rotary switches is equal to 0 the IP address currently used was either obtained from BOOTP or DHCP or previously assigned and read from remanent storage. The table below shows the behaviour of attributes 1, 3 and 5 of the TCP/IP Interface Object (0xF5) when the value of the rotary switches is set to 0.

Instance Attribute of TCP/IP Interface Object (0xF5)	Get	Set	Comment
1	1	Rejected (read-only)	Interface Configuration Status  1 = obtained from BOOTP, DHCP or non-volatile storage
3	0, 1, 2	Accepted	Configuration method  0 = statically 1 = BOOTP 2 = DHCP (default)
5	Current IP configuration	Accepted	Current IP configuration is the stored IP configuration.

#### 4.2.2. Position of rotary switches unequal to 0

If the value read from the rotary switch is unequal to 0 the IP address is "192.168.1." plus the value of the rotary switches which is used for the 4th octet (byte). The value of the 4th octet (byte) of the IP address is defined by the values of the two rotary switches. See also Via Rotary Switches.

When the value of the rotary switches is unequal to 0 (zero) the IP address cannot be changed via the TCP/IP Interface Object (0xF5). The table below shows the behaviour of attributes 1, 3 and 5 of the TCP/IP Interface Object.

Instance Attribute of TCP/IP Interface Object (0xF5)	Get	Set	Comment
1	2	Rejected (read-only)	Valid configuration obtained by hardware settings
3	0 = static	Rejected (0x0C = Object State Conflict)	Configuration method  0 = statically 1 = BOOTP 2 = DHCP (default)
5	Current IP configuration	Rejected (0x0C = Object State Conflict)	Current IP configuration is the stored IP configuration.

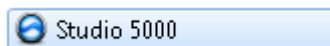
## 5. Engineering (Allen Bradley Studio 5000)

The following examples relate to Allen Bradley PLCs using the engineering tool Studio 5000.

Of course the encoder will also accept PLCs and engineering software of other manufacturers. In this case please proceed in an analog way.

### 5.1. Start Studio 5000

Start Studio 5000 for example from the Windows start menu.



### 5.2. Create project

- Create a new project
- Select the PLC type
- Insert the project name

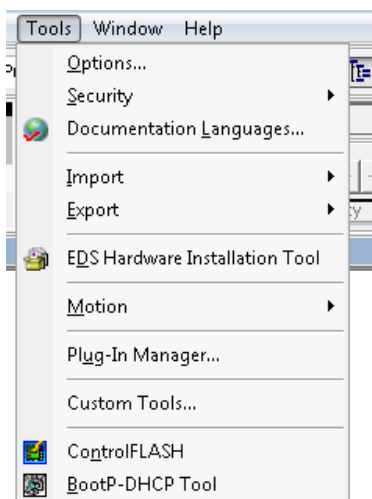
### 5.3. Import EDS file

An EtherNet/IP adapter is described by a so-called "Electronic Data Sheet" (EDS) file.

The EDS file for an EAx580 EtherNet/IP encoder is available for download at [www.baumer.com](http://www.baumer.com).

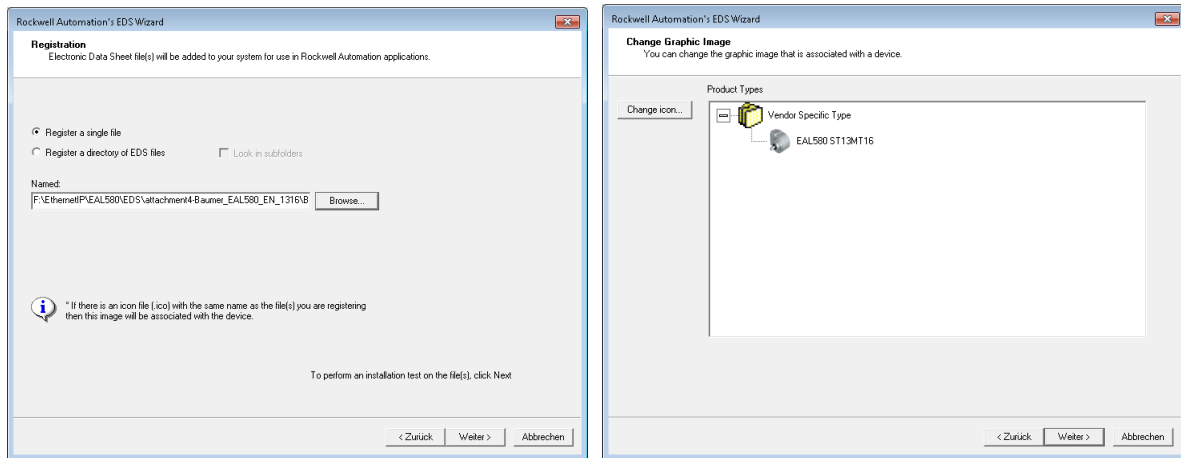
To provide the EDS file to Studio 5000 the following steps have to be done:

- Select „Tools“ -> „EDS Hardware Installation Tool“





Register an EDS file:

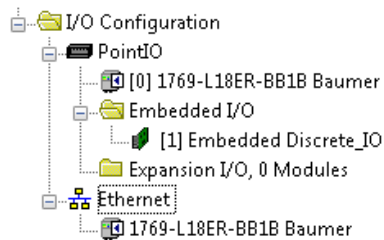


**Note:**

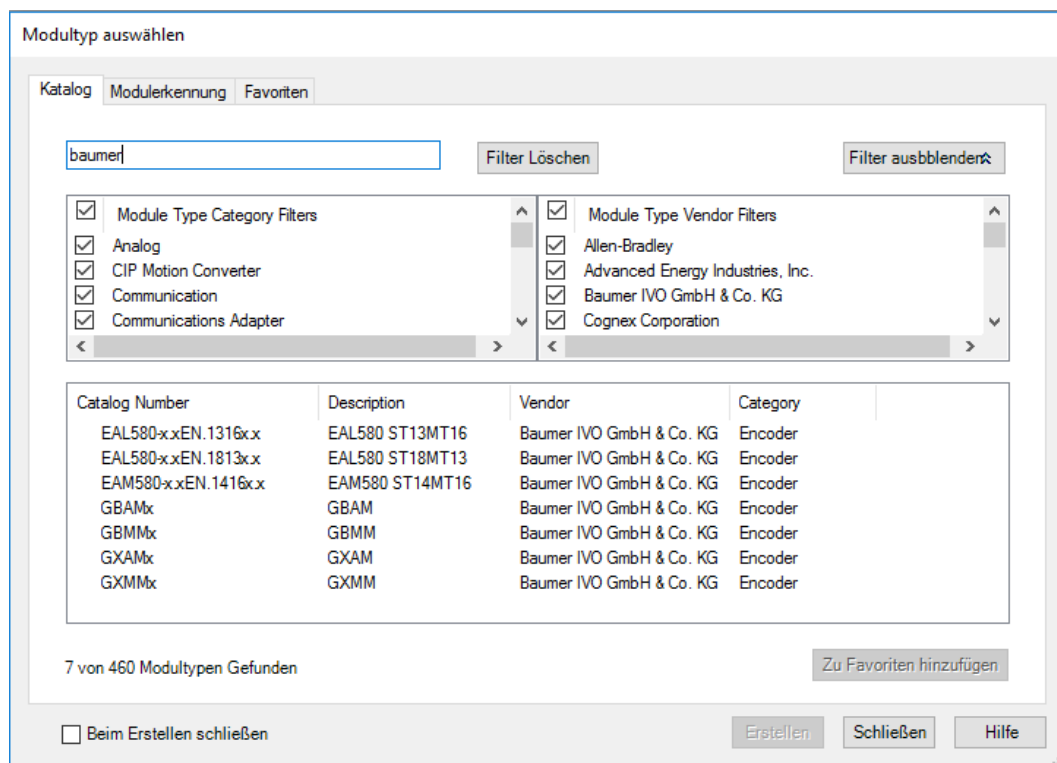
The icon for “EAL580 ST13MT16” in the screenshot above is shown when an EAL580 EtherNet/IP encoder is configured. If an EAM580 EtherNet/IP encoder is configured a different icon is displayed. Additionally EAM580 EtherNet/IP encoders have different product codes.

## 5.4. Module configuration

### 5.4.1. Configuring an encoder module



Right-click on Ethernet symbol in Controller Organizer window and click on “New Module”. Search for “baumer” device and select “EAX580-x.xEN.XXXXx.x”:





Click “Create”, insert name and set the IP address of the encoder. Click “Change ...” to change the configuration.

General | Connection | Module Info | Internet Protocol | Port Configuration | Network

Type: EAL580-x.xEN.1316x.x EAL580 ST13MT16  
Vendor: Baumer IVD GmbH Co. KG  
Parent: Local  
Name: encoder  
Description:   
Ethernet Address:   
☒ Private Network: 192.168.1. 66   
☐ IP Address:   
☐ Host Name:   
Module Definition   
Revision: 1.001   
Electronic Keying: Compatible Module   
Connections: Input Only (1): Position value   
Change ...   
Status: Creating   
OK Cancel Help

Select data type (column “Size”, not changeable for all connections).

Revision: 1 001  
Electronic Keying: Compatible Module  
Connections: 

Name	Remote Data	Size	DINT	Tag Suffix
Input Only (1): Position value	Input:	1	DINT	1 encoder:l1
	Output:	0		<none>
Select a connection				


  
OK Cancel Help

By default connection 1 (Input Only) is selected. The position value is proposed and can be configured to the desired connection.

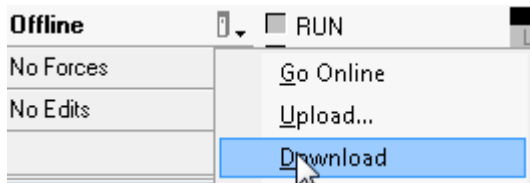
The layout of the cyclic process data image and even the configuration data of the encoder is defined by the type of the connection and can be changed at column “Name”. See chapter [Module Configuration, Connections](#) for more details.

The requested packet interval (RPI) in milliseconds can be configured in tab “Connection”.

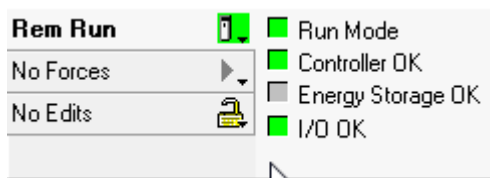
### 5.4.2. Set network path and download configuration

Click on symbol  to select the network path to the PLC.

Click “Download” to download the project to the PLC:

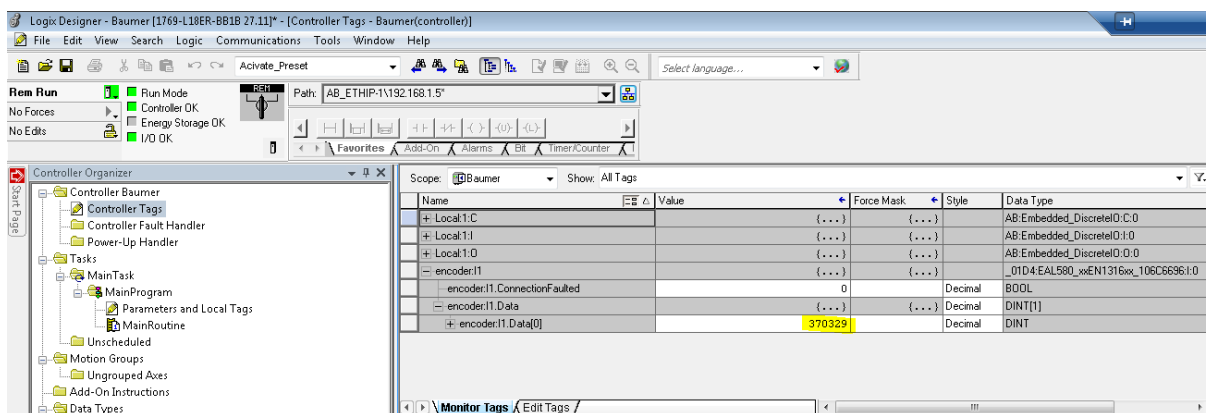


If the encoder is configured and running the following is shown.



### 5.4.3. View encoder data using controller tags

Double-click on “Controller Tags” and watch values at tab “Monitor Tags”:



## 5.5. Module Configuration, Connections

The layout of the cyclic process data image and even the configuration data of the encoder is defined by the type of the connection inside the module configuration. Several connection types are available.

## 5.6. Input Data

The layout of the cyclic process data image is defined by the following four assembly instances.

Assembly Instance 1, Input Data, Length: 4 bytes		
Byte Number	Position Sensor Object, Instance Attribute	Description
Byte 0	3	Position Value
Byte 1		
Byte 2		
Byte 3		

Assembly Instance 2, Input Data, Length: 5 bytes		
Byte Number	Position Sensor Object, Instance Attribute	Description
Byte 0	3	Position Value
Byte 1		
Byte 2		
Byte 3		
Byte 4	46 + 49	Attribute 46 (Alarm Flag) is represented in bit 0. Attribute 49 (Warning Flag) is represented in bit 1.

Assembly Instance 3, Input Data, Length: 8 bytes		
Byte Number	Position Sensor Object, Instance Attribute	Description
Byte 0	3	Position Value
Byte 1		
Byte 2		
Byte 3		
Byte 4	24	Velocity
Byte 5		
Byte 6		
Byte 7		

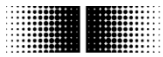
Assembly Instance 110, Input Data, Length: 9 bytes		
Byte Number	Position Sensor Object, Instance Attribute	Description
Byte 0	3	Position Value
Byte 1		
Byte 2		
Byte 3		
Byte 4	24	Velocity
Byte 5		
Byte 6		
Byte 7		
Byte 8	46 + 49	Attribute 46 (Alarm Flag) is represented in bit 0. Attribute 49 (Warning Flag) is represented in bit 1.

## 5.7. Configuration data

The layout of the configuration data is described in the following table.

Assembly Instance 105, Configuration Data, Length: 32 bytes					
Byte Number	Position Sensor Object, Instance Attribute	Description	Default values of encoder variant "ST13MT16"	Default values of encoder variant "ST18MT13"	Default values of encoder variant "ST14MT16"
Byte 0	12	Direction Counting Toggle	0	0	0
Byte 1	14	Scaling Function Control	1	1	1
Byte 2	16	Measuring Units per Span	0x2000	0x40000	0x4000
Byte 3					
Byte 4					
Byte 5					
Byte 6	17	Total Measuring Range	0x20000000	0x80000000	0x40000000
Byte 7					
Byte 8					
Byte 9					
Byte 10	25	Velocity Format	0x1F0F	0x1F0F	0x1F0F
Byte 11					
Byte 12	26	Velocity Resolution	1	1	1
Byte 13					
Byte 14					
Byte 15					
Byte 16	100	Velocity Sample Rate	16	16	16
Byte 17	101	Velocity Filter	5	5	5
Byte 18	102	Gear Factor Activation	0	0	0
Byte 19					
Byte 20	103	Gear Factor Numerator	0x2000	0x1000	0x4000
Byte 21					
Byte 22					
Byte 23					
Byte 24	104	Gear Factor Denominator	1	1	1
Byte 25					
Byte 26					
Byte 27					
Byte 28	107	Preset Request Value	0	0	0
Byte 29					
Byte 30					
Byte 31					

For changing values of any attribute via explicit messages see [Parametrierung per Acyclic Services](#).

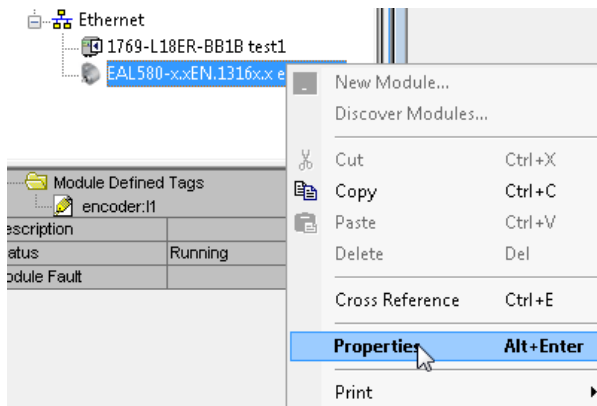


## 5.8. Module Configuration with configuration data

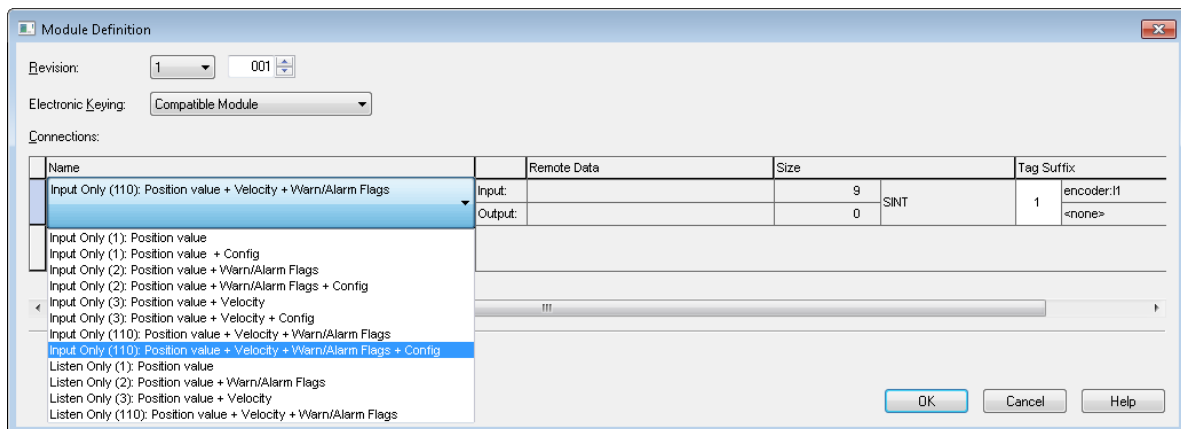
### Note:

For every connection which includes configuration data (see entries with “+ Config”) the configured data is downloaded at connection establishment (“Forward Open” event, e.g. at power-on).

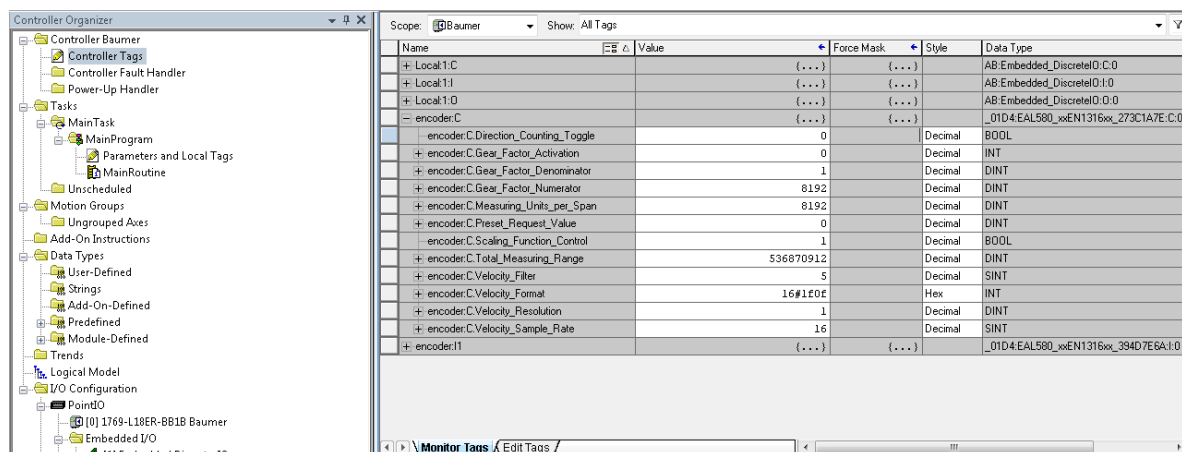
Right-click on symbol of EAx580 EtherNet/IP encoder in Controller Organizer window:

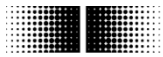


Click on “Properties”. Select “Change” on Tab “General”. Select appropriate connection:



View and edit configuration data in “Controller Tags”:





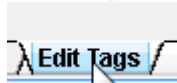
## 5.9. Programming example (preset)

The following steps have to be done to execute a preset by using explicit messaging (acyclic messages).

Click on „Controller Tags“:



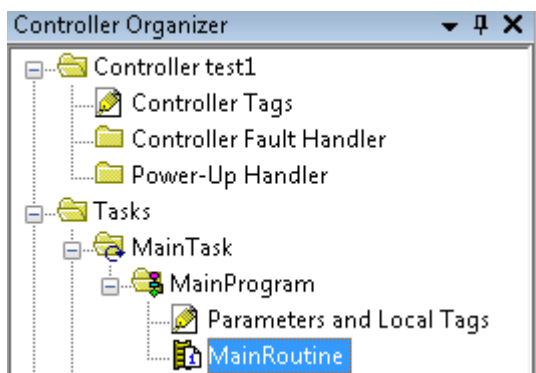
Click on „Edit Tags“:



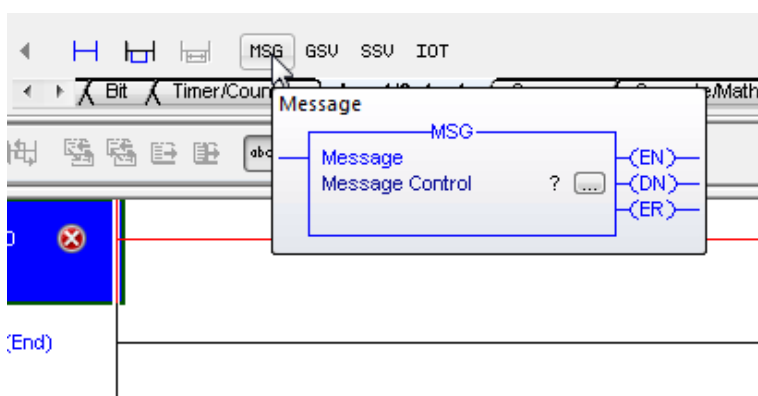
Insert the following 3 new tags:

Tag	Data Type	Description
Msg_SET_23_13_Preset_Value	MESSAGE	Set Preset (message tag holds information about class, instance, attribute and service)
Preset_Value	DINT	Value which is set by preset function
Activate_Preset	BOOL	Forces preset function

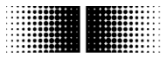
Double-click on “MainRoutine” in “Controller Organizer”:



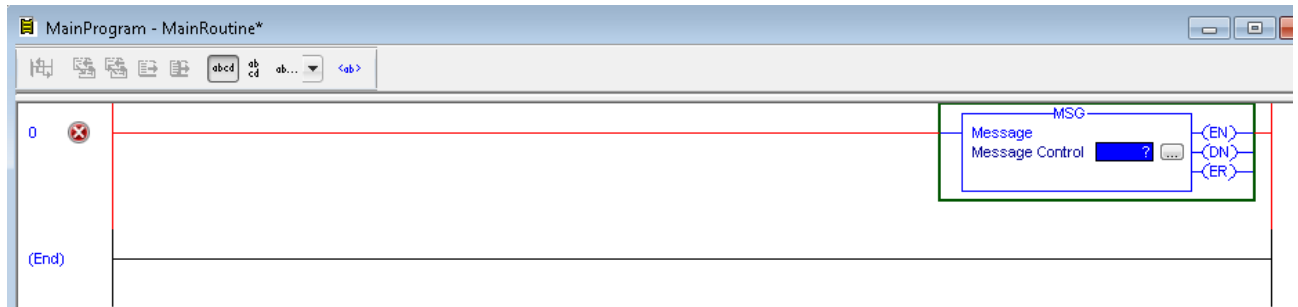
Click on “MSG” in tab “Input/Output”:



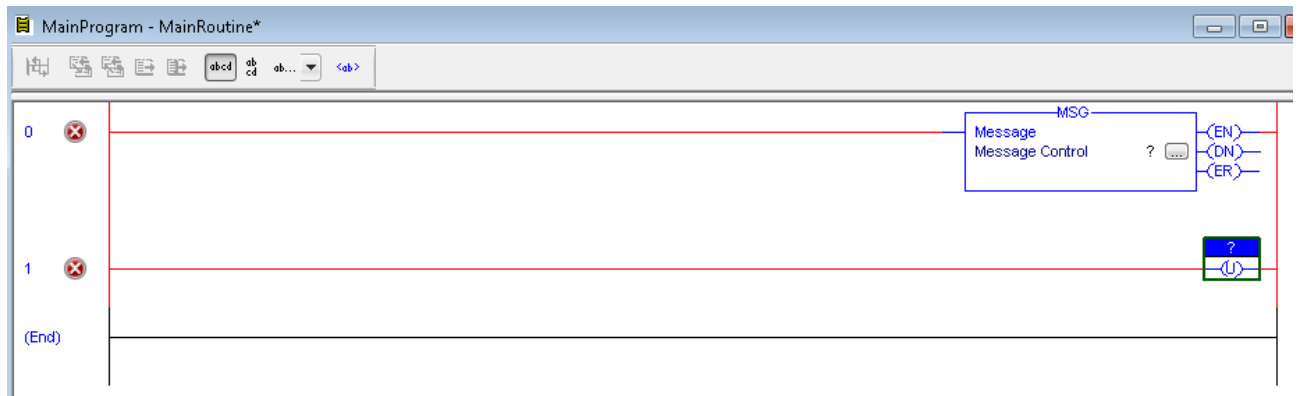




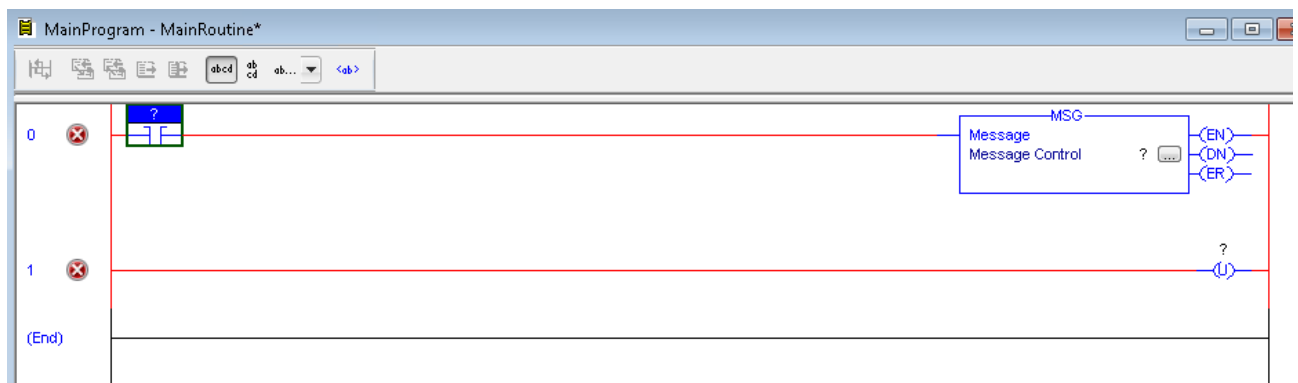
The following box appears.



Right-click in area below “0” and click on “Add Rung”. A new red line appears. Click near “1” to select the lower red line. Click on “Output Unlatch” in tab “Favorites”. A new box appears.

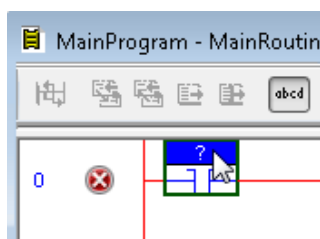


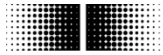
Click near “0” to select the upper red line. Click on “Examine On” in tab “Favorites”. A new box appears.



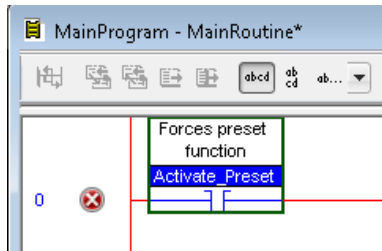
The three new boxes contain question marks. They have to be linked to tags.

Double-click on the question mark of the “Examine On” box:

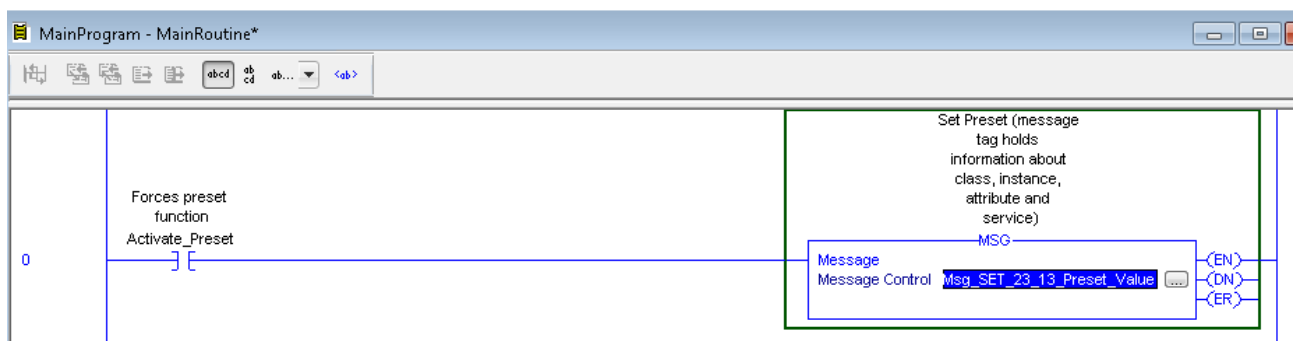




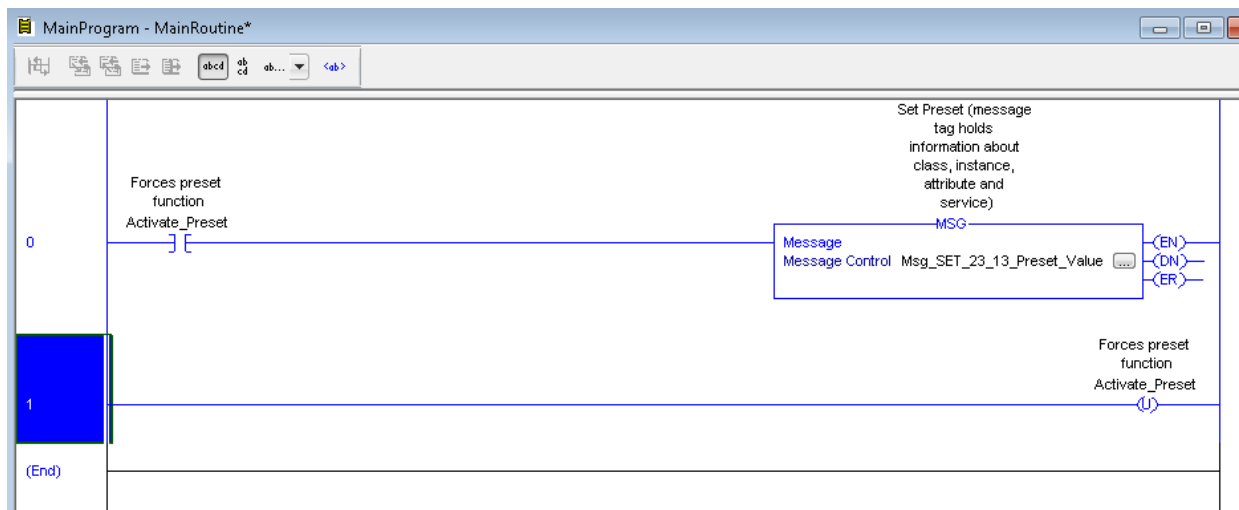
Select “Activate\_Preset” in the appearing drop-down menu. The result is shown below.




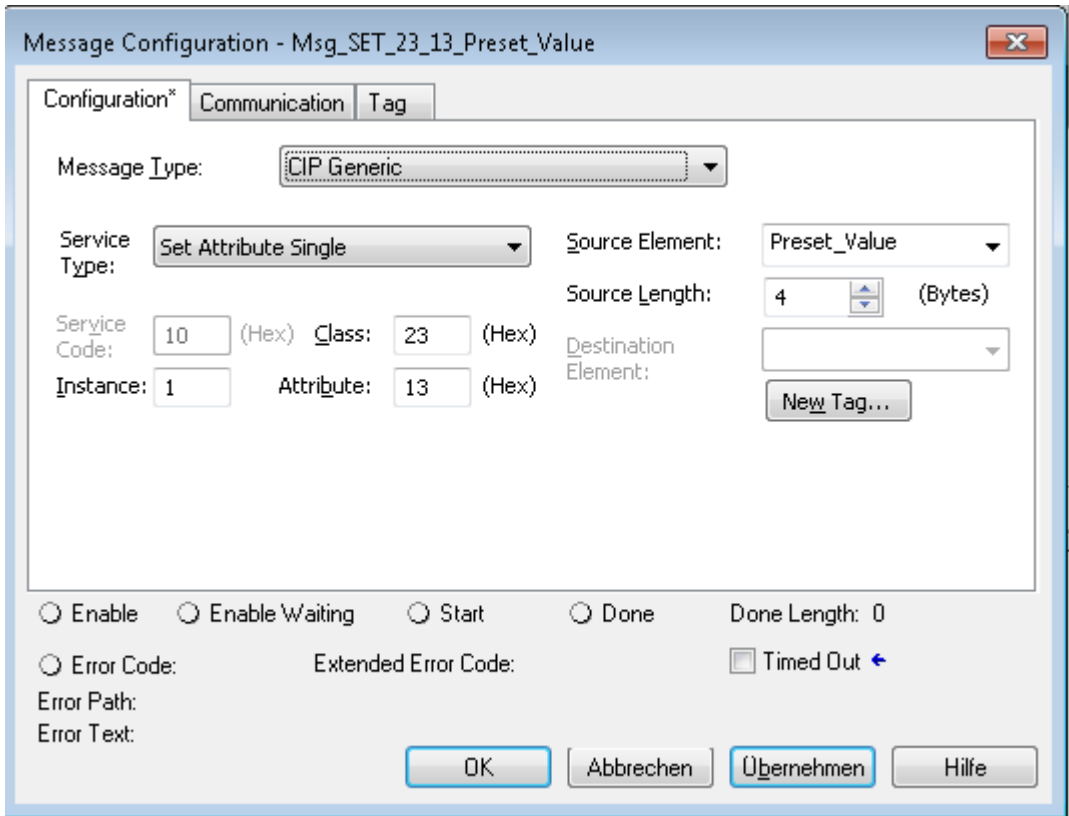
Double-click on question mark of “MSG” box. Select “Msg\_SET\_23\_13\_Preset\_Value” in the appearing drop-down menu. The result is shown below.



Double-click on question mark of “Output Unlatch” box. Select “Activate\_Preset” in the appearing drop-down menu. The result is shown below.



Click on symbol  of “MSG” box (see above). A configuration window appears. To write the preset value (instance attribute 0x13 of Position Sensor Object) insert the following values in the configuration window. Be careful to set all fields as shown below.



The dialog box is titled "Message Configuration - Msg\_SET\_23\_13\_Preset\_Value". It has three tabs: "Configuration\*", "Communication", and "Tag". The "Configuration\*" tab is active.

Fields in the "Configuration\*" tab:

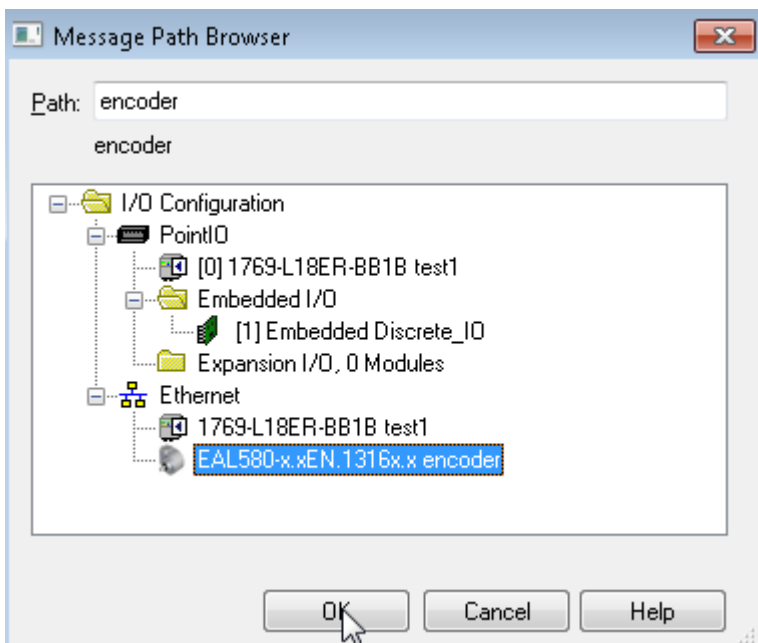
- Message Type: CIP Generic (dropdown)
- Service Type: Set Attribute Single (dropdown)
- Source Element: Preset\_Value (dropdown)
- Source Length: 4 (spin box) (Bytes)
- Service Code: 10 (Hex)
- Class: 23 (Hex)
- Destination Element: (empty dropdown)
- Instance: 1
- Attribute: 13 (Hex)
- New Tag... (button)

Buttons at the bottom:

- Enable (radio button)
- Enable Waiting (radio button)
- Start (radio button)
- Done (radio button)
- Done Length: 0
- Error Code: (radio button)
- Extended Error Code: (radio button)
- Timed Out (checkbox)
- OK (button)
- Abbrechen (button)
- Übernehmen (button)
- Hilfe (button)

Before you can click on “Übernehmen” (apply) in tab “Configuration” you have to open tab “Communication” to set the path to the encoder.

Click on “Browse...”, select encoder and click OK.



The dialog box is titled "Message Path Browser". It has a "Path:" field containing "encoder". Below the field is a tree view showing the following structure:

- I/O Configuration
  - PointIO
    - [0] 1769-L18ER-BB1B test1
  - Embedded I/O
    - [1] Embedded Discrete\_IO
  - Expansion I/O, 0 Modules
- Ethernet
  - 1769-L18ER-BB1B test1
  - EAL580-x.xEN.1316x.x encoder (selected)

Buttons at the bottom:

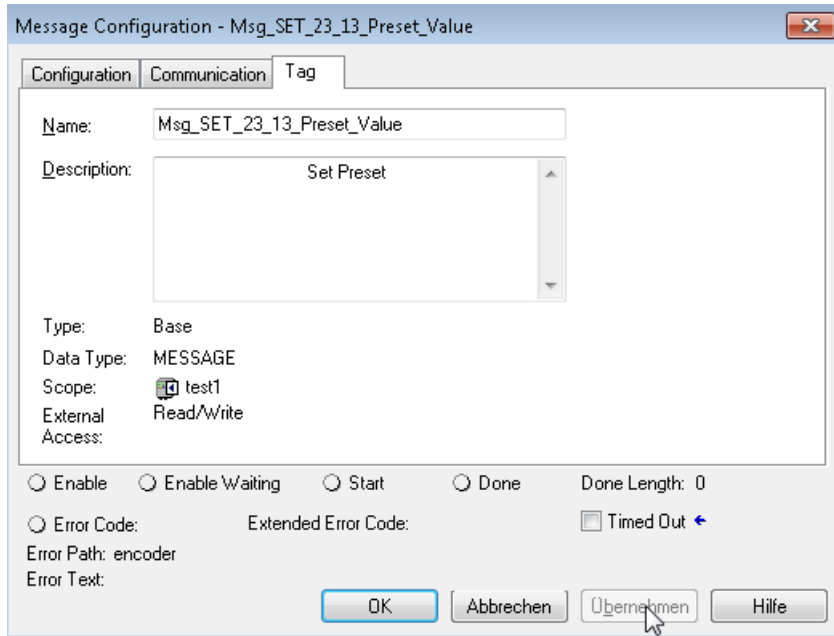
- OK (button)
- Cancel (button)
- Help (button)



Click on “Übernehmen” (apply) in tab “Communication”.

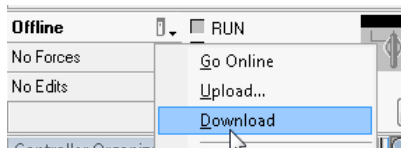
Click on “Übernehmen” (apply) in tab “Configuration” (if not already done).

Open tab “Tag” to enter a description and click on “Übernehmen” (apply) in tab “Tag”:

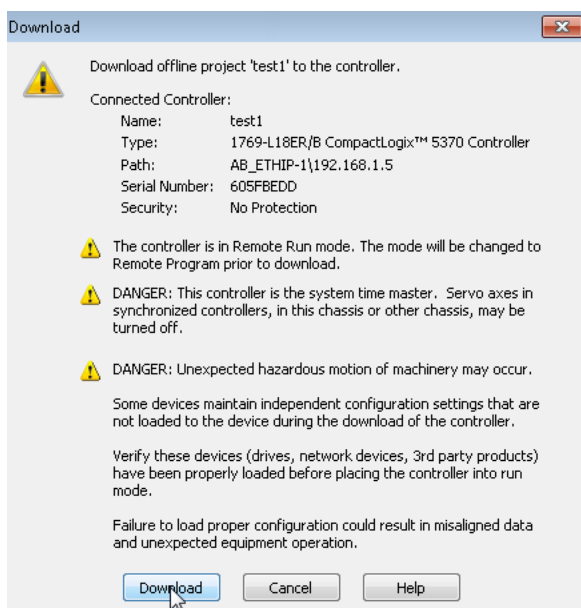


Click on OK to close the window.

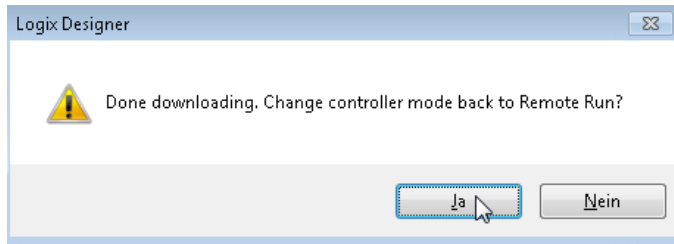
Now the project has to be compiled and downloaded to the PLC. To do this click on “Download”:



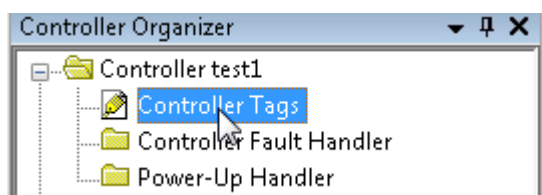
In appearing window click on “Download”:



Click “Ja” (yes) in the following window:



In the online view you can trigger a preset now. Before doing this please check the current position value by opening “Controller Tags” in Controller Organizer:

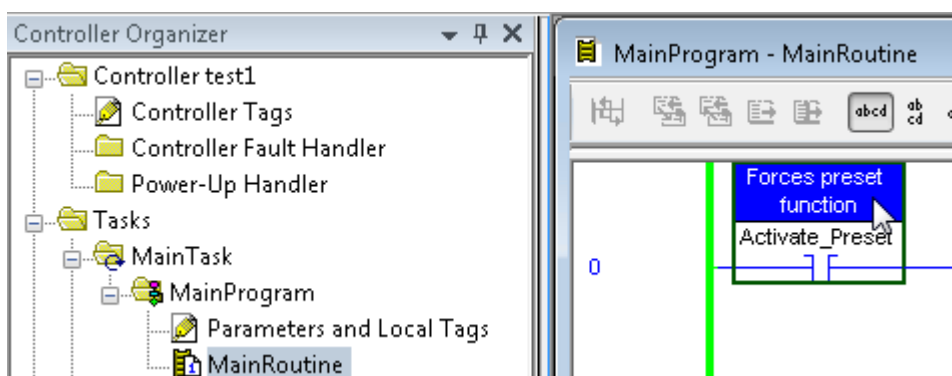


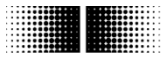
Open “Monitor Tags” and open data of encoder as follows:

Controller Tags - test1(controller)									
Scope: test1		Show: All Tags							
Name	Value	Force Mask	Style	Data Type	Description	Constant			
Activate_Preset	0		Decimal	BOOL	Forces preset fun...	<input type="checkbox"/>			
encoder:11	{...}	{...}		_01D4:EAL580_x...		<input type="checkbox"/>			
encoder:11.ConnectionFaulted	0		Decimal	BOOL					
encoder:11.Data	{...}	{...}	Decimal	SINT[4]					
encoder:11.Data[0]	-120		Decimal	SINT					
encoder:11.Data[1]	3		Decimal	SINT					
encoder:11.Data[2]	0		Decimal	SINT					
encoder:11.Data[3]	0		Decimal	SINT					
Local1:C	{...}	{...}		AB:Embedded_Di...		<input type="checkbox"/>			
Local1:I	{...}	{...}		AB:Embedded_Di...		<input type="checkbox"/>			
Local1:O	{...}	{...}		AB:Embedded_Di...		<input type="checkbox"/>			
Msg_SET_23_13_Preset_Value	{...}	{...}		MESSAGE	Set Preset	<input type="checkbox"/>			
Preset_Value	0		Decimal	DINT	Value which is set...	<input type="checkbox"/>			

Position value is unequal 0 in this example.

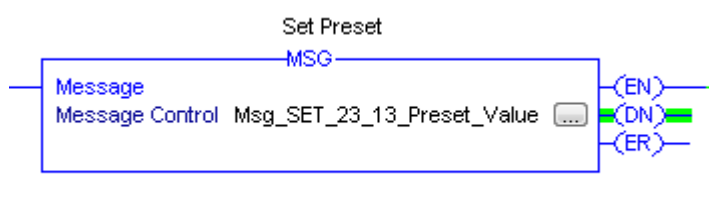
Open “MainRoutine” and click on following box.





Type Ctrl+T to trigger a preset.

Line “DN” of “MSG” box gets green to signal that the preset has been done successfully:



Open “Monitor Tags” and open data of encoder as follows:

Name	Value	Force Mask	Style	Data Type	Description	Constant
Activate_Preset	0		Decimal	BOOL	Forces preset fun...	<input type="checkbox"/>
encoder.I1	{...}	{...}				<input type="checkbox"/>
encoder.I1.ConnectionFaulted	0		Decimal	BOOL		
encoder.I1.Data	{...}	{...}	Decimal	SINT[4]		
+ encoder.I1.Data[0]	0		Decimal	SINT		
+ encoder.I1.Data[1]	0		Decimal	SINT		
+ encoder.I1.Data[2]	0		Decimal	SINT		
+ encoder.I1.Data[3]	0		Decimal	SINT		
+ Local1:C	{...}	{...}		AB: Embedded_Di...		<input type="checkbox"/>
+ Local1:I	{...}	{...}		AB: Embedded_Di...		<input type="checkbox"/>
+ Local1:O	{...}	{...}		AB: Embedded_Di...		<input type="checkbox"/>
+ Msg_SET_23_13_Preset_Value	{...}	{...}		MESSAGE	Set Preset	<input type="checkbox"/>
+ Preset_Value	0		Decimal	DINT	Value which is set...	<input type="checkbox"/>

Position value is 0 now. The preset has been executed successfully.

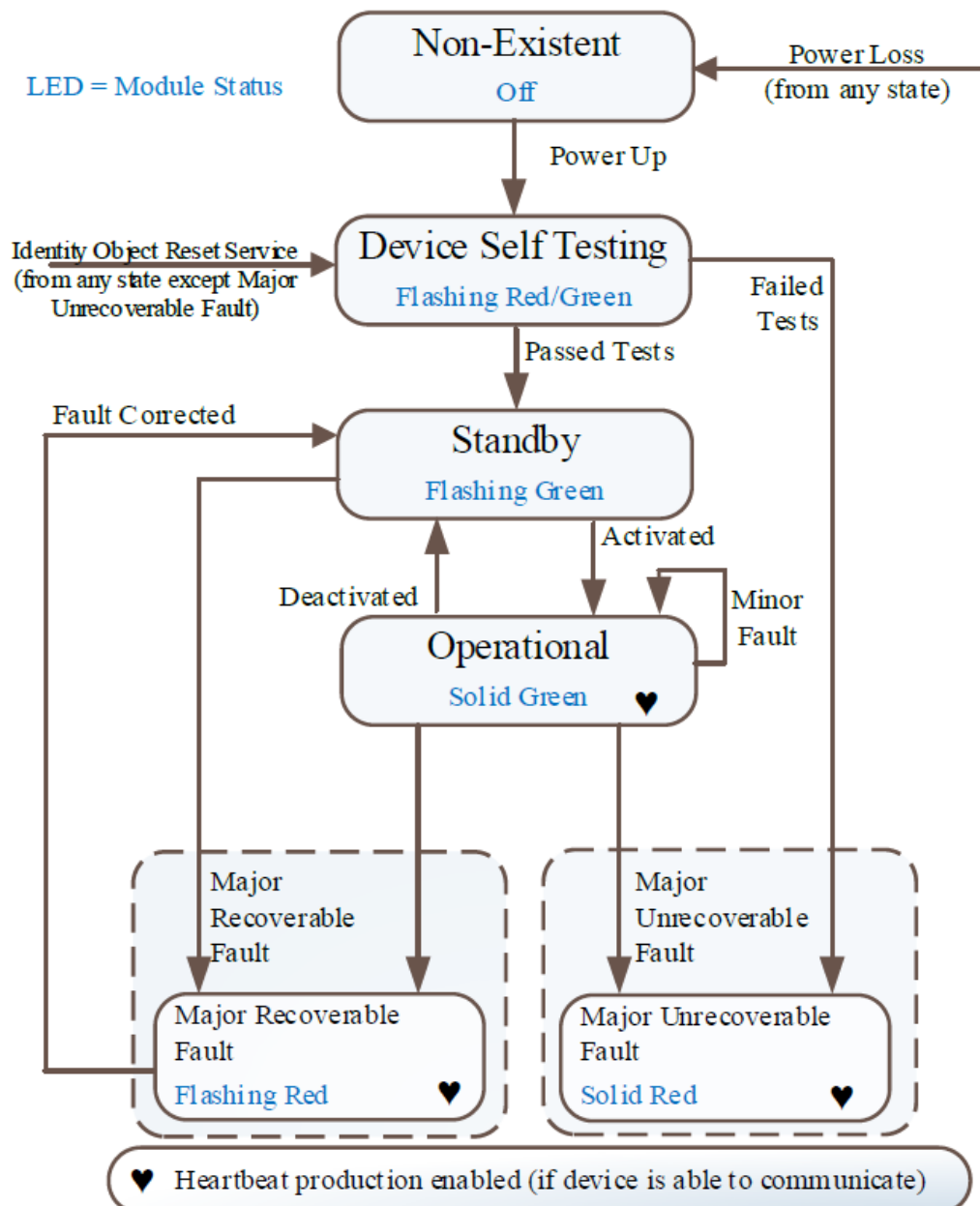
## 6. EtherNet/IP cyclic operation

### 6.1. PLC (EtherNet/IP scanner)

In an EtherNet/IP network the encoder acts as an EtherNet/IP adapter. The programmable logic controller (PLC) acts as an EtherNet/IP scanner. There are several EtherNet/IP scanners from different vendors available. The PLC exchanges cyclic and acyclic data with the encoder.

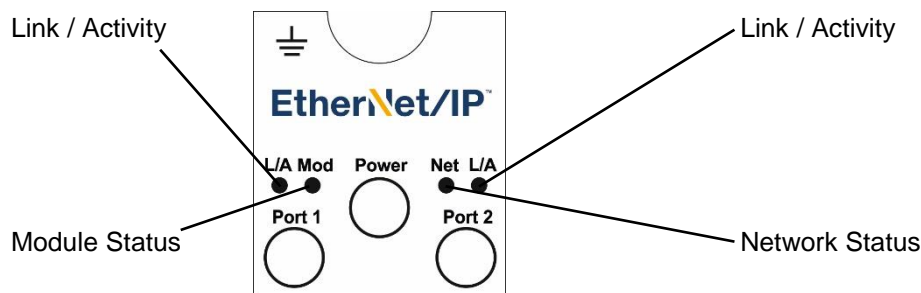
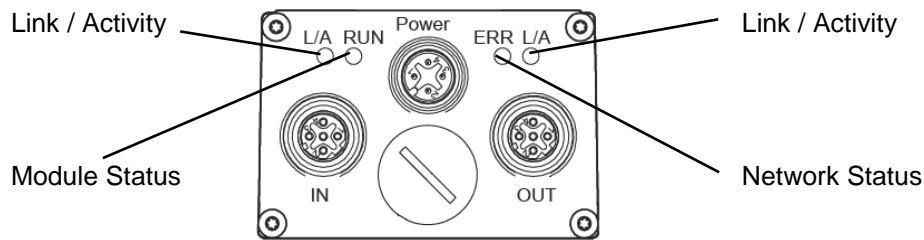
### 6.2. Basics

For cyclic operation a so-called cyclic process data image is used. This image contains input and output data. The EAx580 EtherNet/IP encoder only sends cyclic process data to the master. From the master to the EtherNet/IP encoder no cyclic process data is transmitted. EtherNet/IP adapters implement a state machine which consists of several device states. The device states are documented in the following picture (according to specification "The CIP Networks Library Volume 1", Edition 3.24, figure 5A-2.2).



### 6.3. Diagnostic LEDs

The LEDs of the encoder have the following meaning.



#### 6.3.1. Link/Activity (L/A) LEDs

The encoder provides two Duo LEDs to indicate the status of the Ethernet ports. The following table shows the different possible states of the link/activity LEDs.

LED	Color	Status	Description
LINK (Port 1 and 2)	Green	On	The device is linked to Ethernet.
	Green	Off	The device has no link to Ethernet.
ACT (Port 1 and 2)	Red	Flickering (load dependent)	The device sends/receives Ethernet frames.
	Red	Off	The device does not send/receive Ethernet frames.

If the device is linked to Ethernet and sends/receives Ethernet frames the related L/A LED will give a visual impression of a mix of the colors green and red.

#### 6.3.2. Network Status Indicator

The encoder provides a Duo LED to indicate the network status of the device.

Color	Status	Description
Green	On	<b>Connected:</b> The device has at least one established connection.
Green	Flashing (1 Hz)	<b>No Connections:</b> The device has not established connections but has obtained an IP address.
Green/Red	Flashing (1 Hz)	<b>Self-test:</b> The device is performing its power-up testing.
Red	On	<b>Duplicate IP:</b> The device has detected that its IP address is already in use.
Off	Off	<b>Not powered, no IP address:</b> The device does not have an IP address or the power supply of the device is switched off.



### 6.3.3. Module Status Indicator

The encoder provides a Duo LED to indicate the module status.

Color	Status	Description
Green	On	<b>Device operational:</b> The device is operating correctly.
Green	Flashing (1 Hz)	<b>Standby:</b> The device has not been configured.
Green/Red	Flashing (1 Hz)	<b>Self-test:</b> The device is performing its power up testing.
Red	Flashing (1 Hz)	<b>Minor fault:</b> The device has detected a recoverable minor fault. E.g. an incorrect or inconsistent configuration can be considered as a minor fault.
Red	On	<b>Major fault:</b> The device has detected a non-recoverable major fault.
Off	Off	<b>No power:</b> The power supply to the device is missing.

### 6.4. Mapping of process data

For the EAx580 EtherNet/IP encoder it is possible to change the content and layout of the cyclic process data which is sent to the EtherNet/IP scanner. The process data of the device is organized by the assembly object. See chapter [Assembly Object \(0x04\)](#) for further details.

## 6.5. Speed

The speed value is specified in Instance Attribute 24 - Velocity Value.

Assembly instances 3 and 110 provide the speed value and the absolute position value.

It is possible to use a filter (calculation of average value) for the calculation of the speed value. For the configuration of the filter both parameters "Speed Update Period" and "Speed Filter Depth" are used. In chapters Speed update period and Speed filter depth details can be found.

When a gear factor is used the following has to be considered for the calculation of the speed value:

- The gear factor is not taken into account in the speed value.
- The speed value refers to the drive side.

### 6.5.1. Speed measuring unit

The measuring unit for the speed value is defined by Instance Attribute 25 - Velocity Format and Instance Attribute 26 - Velocity Resolution.

The following scaling options are possible:

- rpm                      revolutions per minute
- steps/10ms          number of steps (in the configured singleturn resolution) per 10 ms
- steps/100ms        number of steps (in the configured singleturn resolution) per 100 ms
- steps/1000ms      number of steps (in the configured singleturn resolution) per second

For each scaling option the measured value is provided as a „signed integer“. Positive values indicate the direction of rotation with rising position values. Which rotational direction is assigned "positive" depends on the CW/CCW parameter setting.

### 6.5.2. Speed update period

The time window for the update of the speed value is defined by Instance Attribute 100 - Velocity Sample Rate.

A short speed update period entails a more dynamic speed output, whereas an extended update period ensures more stable values. The optimum speed update period depends on the requirements of the application.

The update period can be configured within a range from 1 ms to 255 ms.

### 6.5.3. Speed filter depth

The filter depth is defined by Instance Attribute 101 - Velocity Filter.

A shallow filter depth entails a more dynamic speed output, whereas a higher filter depth ensures more stable values. The optimum speed filter depth in interaction with the configured speed update time depends on the requirements of the application.

The filter depth is configurable within the range from 1 to 255.

## 6.6. Preset function

The preset function in the encoder is triggered by writing Instance Attribute 19 - Preset Value or by pushing the preset button. When the preset is triggered the encoder position value is set to the given preset value.

**For the optimum alignment of the mechanical position we recommend to set the preset value during encoder standstill.**

It is mandatory to parameterize the required resolution or code sequence (cw/ccw) prior to performing the preset operation.

Upon performing a preset operation an internal offset is calculated and stored in the non-volatile memory to make sure that the encoder will be at the same position after power cycling. Although the non-volatile memory provides more than 100000 write cycles frequent software-triggered or event-triggered preset operations may consume even this capacity. This has to be considered in the PLC software configuration.

**Note:**

Any alteration of the total measuring range, measuring units per revolution or a gear factor parameter during encoder re-parameterization will trigger clearing the internal preset offset value. This however has no effect in practice since in these cases the position reference is lost anyway.

### 6.6.1. Preset via Position Sensor Object

To execute a preset the preset value has to be written into Instance Attribute 19 - Preset Value by an explicit message with service "Set Attribute Single". See Position Sensor Object (0x23) for details.

### 6.6.2. Preset using the push button

Depending on the encoder type the encoder may have a screw cap located where connectors and LEDs are located as well. After removing the screw cap the preset push button is visible.

While the encoder is connected to an EtherNet/IP scanner (master) a preset can be performed using the push button. When the preset is executed the encoder position is set to the value held in Instance Attribute 107 - Preset Request Value of the Position Sensor Object. The contained position value will be interpreted absolute. A relative preset is not possible using the push button.

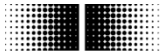
**Note:**

Prior to using the push button it is strictly recommended to execute a potential equalization between operator and encoder (touching the encoder housing) in order to eliminate the danger of damaging the encoder by electrostatic discharge (ESD).

In order to perform a preset the push button must be pressed for a duration of at least three seconds and longest five seconds.

## 6.7. Cycle time and supported functionality

The encoder supports a minimal Requested Packet Interval (RPI) time of 1 millisecond. The position is sampled every 500 microseconds.



## 7. EtherNet/IP acyclic operation

### 7.1. Introduction

As an application layer EtherNet/IP uses the Common Industrial Protocol (CIP) released by the ODVA. CIP is transmitted as an “encapsulated” protocol in the data section of standard Ethernet frames. Depending on the assignment and type of the connection the data transmission mechanisms UDP/IP or TCP/IP are used.

CIP is an object-oriented protocol. The device characteristics are described by which have one or more instances. Each instance has one or more attributes. Attributes describe individual characteristics of objects such as parameter values.

In device profiles the ODVA defines which CIP objects and attributes have to be supported by a certain device class. In addition, optional and manufacturer-defined objects and attributes are also possible.

Data transmission of CIP messages in EtherNet/IP networks takes place by means of implicit and explicit messages. Typically, implicit messages are smaller data packages for time-critical data transmissions. When transmitting I/O data implicit connections with long-term viability are generally involved. I/O data is transmitted by means of UDP. Non time-critical data is transmitted by means of explicit messages. Examples of explicit messages are configuration or information data which use the TCP/IP transmission mechanisms.

More detailed information about the Common Industrial Protocol (CIP) or EtherNet/IP can be obtained from the ODVA ([www.odva.org](http://www.odva.org)).

### 7.2. Device Type 0x22 (encoder profile)

The EAx580 EtherNet/IP encoder family was designed according to profile 0x22 (encoder profile) of the ODVA.

### 7.3. CIP Object Model

The object model describes the used object classes of the encoder and their mutual relationship. This is defined in the device profile 0x22 (encoder profile) of the ODVA. Cyclic and acyclic data communication run in parallel and independently of each other.

### 7.4. Standard objects

The encoder supports the following standard objects:

- Identity Object (0x01)
- Message Router Object (0x02)
- Assembly Object (0x04)
- Connection Manager Object (0x06)
- Device Level Ring Object (0x47)
- Quality of Service Object (0x48)
- TCP/IP Interface Object (0xF5)
- Ethernet Link Object (0xF6)

### 7.5. Profile-specific objects

The encoder supports the following profile-specific object:

- Position Sensor Object (0x23)

## 7.6. Parameterization

The following parameters are members (instance attributes) of the Position Sensor Object (0x23).

### 7.6.1. Measuring units per revolution

The value for measuring units per revolution has to be within the range of 1 to maximum encoder resolution.

For details please refer to Instance Attribute 16 - Measuring Units per Span.

### 7.6.2. Total measuring range

Admissible values range from 2 to the product of the programmed resolution multiplied by the encoder's maximum number of revolutions.

For details please refer to Instance Attribute 17 - Total Measuring Range in Measuring Units.

### 7.6.3. Code sequence

Position data behaviour relates to the rotation direction of the shaft of the encoder when looking at the flange.

CW („clockwise“)	= ascending values with clockwise rotation
CCW („counter-clockwise“)	= ascending values with counter-clockwise rotation

For details please refer to Instance Attribute 12 - Direction Counting Toggle.

### 7.6.4. Scaling functionality

If scaling functionality is active the settings of „measuring units per revolution“ and „total measuring range“ are considered.

For details please refer to Instance Attribute 14 - Scaling Function Control.

### 7.6.5. Speed measuring unit

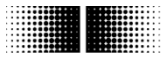
Please refer to Speed Measuring Unit.

### 7.6.6. Speed update period

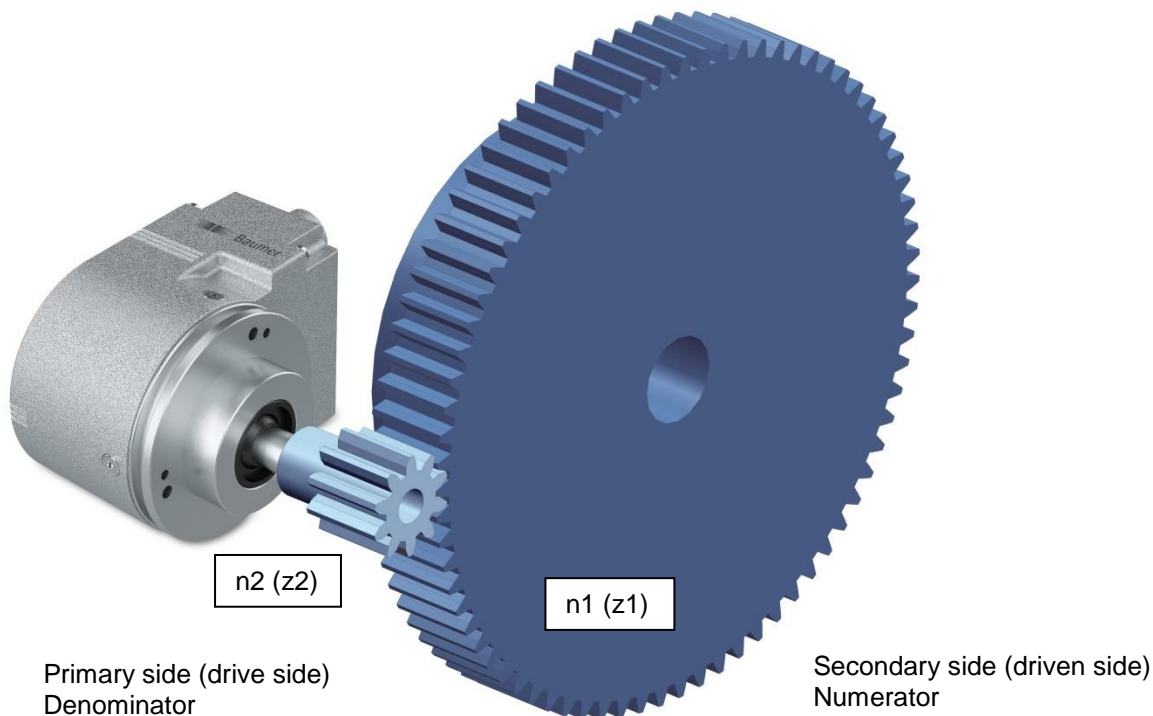
Please refer to Speed Update Period.

### 7.6.7. Speed filter depth

Please refer to Speed Filter Depth.

**7.6.8. Gear factor: activation**

With gear factor active the encoder is mechanically mounted on the primary side (drive side) of the gearbox while it outputs position data as if it was mounted on the secondary side (driven side) of the gearbox. The parameter „total measuring range“ always defines the number of requested steps for one revolution on the secondary side of the gear box.



$$\text{Gear factor } i = \frac{\text{Numerator}}{\text{Denominator}} = \frac{\text{Speed at drive side (n2)}}{\text{Speed at driven side (n1)}} = \frac{\text{Number of teeth at driven side (z1)}}{\text{Number of teeth at drive side (z2)}}$$

The values for numerator and denominator of the gear factor result directly from the number of teeth. In the example above the number of teeth at the driven side is 75. At the drive side the number of teeth is 10.

Parameter „measuring units per revolution“ is not set in the gear factor function. Instead the parameter „measuring units per revolution“ results from total measuring range, numerator and denominator.

$$\text{Measuring units per revolution} = \text{total measuring range} * \frac{\text{denominator}}{\text{numerator}}$$

**Example:**

The gear factor shall be 75:10 (means 7,5).

The resolution on the secondary side of the gearbox shall be “1 revolution = 10000 steps”.

The numerator is 75 and the denominator is 10. Only integer values are admissible for numerator and denominator. The total measuring range is 10000.

The encoder turns 7,5 revolutions for one revolution on the secondary side of the gearbox. The resulting value of „measuring units per revolution“ for the encoder is  $10000 / 7,5 = 1333,3333$ .

**Note:**

A change of [Instance Attribute 102 - Gear Factor Activation](#) clears internal position offsets (if any). So the current position reference is lost (see also [Preset function](#)).

The gear factor functionality is also called „numerator/denominator scaling“ or „round axis function“.

**7.6.9. Gear factor: numerator**

This parameter is only taken into account when gear factor functionality is active.

When using a reduction gear ( $n_2 < n_1$ ) the numerator of the gear factor is larger than the denominator.

**Note:**

For the “numerator” the following restrictions apply:

EAL580 MT encoder ST13 MT16, optical:	numerator $\leq$ 8192
EAL580 MT encoder ST18 MT13, optical:	numerator $\leq$ 4096
EAM580 MT encoder ST14 MT16, magnetic:	numerator $\leq$ 16384

**7.6.10. Gear factor: denominator**

This parameter is only taken into account when gear factor functionality is active.

When using a step-up gear ( $n_2 > n_1$ ) the denominator is larger than the numerator.

**7.6.11. Gear factor: parameterization**

Valid combinations of numerator, denominator and total measuring range result from the formula noted below. The parameter „measuring units per revolution“ must not exceed the maximum allowed values of the encoder. These values are different depending on the type of the encoder.

$$\text{Measuring units per revolution} = \text{total measuring range} * \frac{\text{denominator}}{\text{numerator}}$$

EAL580 MT encoder ST13 MT16, optical:	Measuring units per revolution $\leq$ 65536
EAL580 MT encoder ST18 MT13, optical:	Measuring units per revolution $\leq$ 524288
EAM580 MT encoder ST14 MT16, magnetic:	Measuring units per revolution $\leq$ 65536

### 7.6.12. Important note for multiturn encoder operation

„Endless operation“ is automatically supported where required.

Thus, there are no special requirements for the encoder parameters “total measuring range” and “measuring units per revolution” to stand in a certain ratio.

**When endless operation is active and the encoder is unpowered the encoder shaft may rotate max.  $\frac{1}{4}$  of the number of maximum possible revolutions.** For an encoder with max. 65536 ( $2^{16}$ ) revolutions (equals 16 „multiturn bits“) this would be 16384 ( $2^{14}$ ) revolutions. For an encoder with 13 „multiturn bits“ this then would be 2048 ( $2^{11}$ ) revolutions. If this number of revolutions is exceeded the encoder has to be referenced (execute [Preset function](#)) after every power-up. When endless operation is inactive the encoder shaft may rotate unlimitedly when the encoder is not powered (without influence on the position value).

How to determine if „Endless Operation“ is in use together with given parameters:

- multiply the encoder’s „max. possible revolutions“ (depending on encoder type: for 16 bits the value is 65536, for 13 bits the value is 8192) by the value of parameter „measuring units per revolution“
- divide this value by the value of parameter „total measuring range“
- if there is a division remainder endless operation is in use

#### Example for parameters without endless operation:

Maximum possible revolutions	65536	(16 bits multiturn)
Measuring units per revolution	3600	
Total measuring range	29491200	
Calculation:	$65536 \times 3600 / 29491200 = 8$ (no remainder)	

#### Example for parameters with endless operation:

Maximum possible revolutions	65536	(16 bits multiturn)
Measuring units per revolution	3600	
Total measuring range	100000	
Calculation:	$65536 \times 3600 / 100000 = 2359$ (remainder 29600)	

### 7.6.13. Preset

Please refer to [Preset function](#).

### 7.6.14. Parameterization sequence

Please proceed in the following sequence:

- write [Instance Attribute 12 - Direction Counting Toggle](#)
- write [Instance Attribute 17 - Total Measuring Range in Measuring Units](#)
- write [Instance Attribute 16 - Measuring Units per Span](#)
- use “Apply Service” of Position Sensor Object (0x23)
- move shaft of encoder to desired reference position
- write [Instance Attribute 19 - Preset Value](#)

See [Position Sensor Object \(0x23\)](#) for details.





### 7.6.15. Parameterization behaviour

The following chapter applies for all parameters of the Position Sensor Object marked with a “(c)” (see [Instance Attributes](#)).

#### *Parameterization via Configuration Assembly*

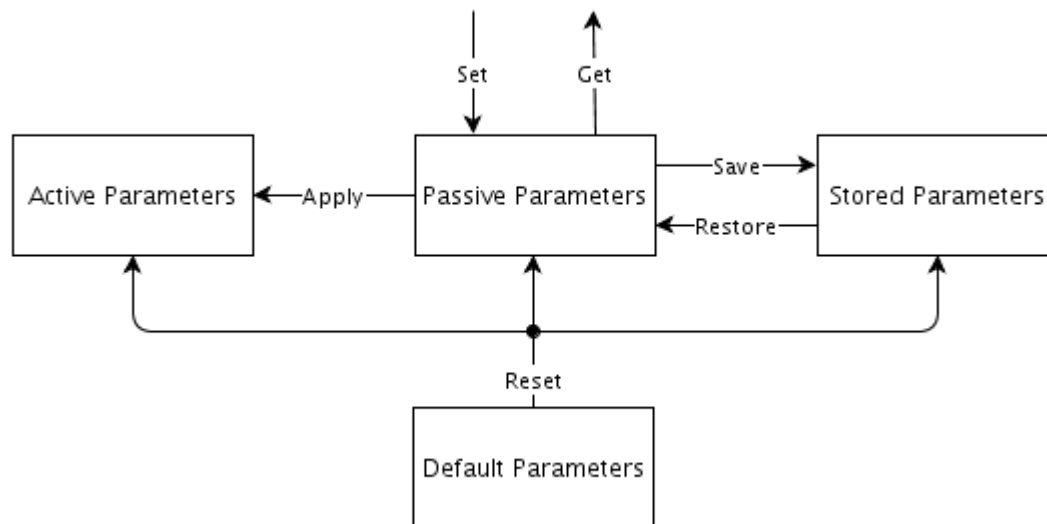
A parameterization during start up is possible via the configuration assembly (see [Instance \(105\) - Configuration](#)). All parameters of the assembly are immediately applied and stored during connection establishment.

#### *Parameterization via Acyclic Services*

Depending on Attribute 110 (see [Instance Attribute 110 - Set Parameter Non-Volatile](#)) the acyclic encoder parameterization changes. Both parameter behaviours are described the next chapters.

#### **Default Parameter Behaviour**

If attribute 110 has the value 0 (factory setting) the following parameter behaviour is valid:



The encoder works with the “Active Parameters”. The transitions can be executed by the according services described in [Position Sensor Object \(0x23\)](#).

Transition	Service
Set	Set_Attribute_Single (see <a href="#">Instance Services</a> )
Get	Get_Attribute_Single (see <a href="#">Instance Services</a> )
Apply	Apply_Attributes (see <a href="#">Class Services</a> )
Reset	Reset (see <a href="#">Class Services</a> )
Save	Save (see <a href="#">Class Services</a> )
Restore	Restore (see <a href="#">Class Services</a> )

#### **Alternative Parameter Behaviour**

If attribute 110 is set to 1 the alternative parameter behaviour is applied. Parameters become active and are stored immediately at the time when they are set. The following transitions (services) are no longer accessible:

- Apply
- Save
- Restore

The parameterization has to be done as described in [Parameterization sequence](#).

## 7.7. Error Handling

Reported errors of the encoder are mapped to warning and alarm bits.

- If an error appears one or more corresponding warning and/or alarm bits are set.
- If an error disappears one or more corresponding warning and/or alarm bits are reset.

The warning and alarm bits are mapped to a defined device status. The device status is evaluated depending on a priority.

### 7.7.1. Warning and alarm bits

Warning and alarm bits are available in [Instance Attribute 47 – Warnings](#) and [Instance Attribute 44 – Alarms](#).

### 7.7.2. Mapping table

The following table describes the reaction to certain diagnosis events in instance attributes 44 and 47.

Diagnosis event	Bit in Position Sensor Object (instance attribute 47)	Bit(s) in Position Sensor Object (instance attribute 44)
	Warnings	Alarms
Temperature sensor communication	15	-
Light control reserve error	1	0 and 12
Internal CRC error	-	0 and 13
Internal frame error	-	0 and 14
Speed violation	0	0
Internal CRC error (configuration)	-	1 and 15
Operation hours counter error	14	-
Encoder runtime data error	-	15
Parameter data error	-	15
Low battery voltage	4	-
Magnetic sampling error	-	0 and 12
Magnetic sampling warning	13	-
Position measurement trigger error	-	-
CRC error of EtherNet/IP-related data	-	15

### 7.7.3. Mapping of warning bits to device status

The following table shows the mapping of the warning bits (see [Instance Attribute 47 – Warnings](#) of Position Sensor Object) to the device status (see [Attribute 5 \(Status\)](#) of Identity Object).

Warning Bit(s)	Device Status
0 (Frequency exceeded)	Minor Recoverable Fault
1 (Light control reserve)	Minor Unrecoverable Fault
2-3	Not Supported
4 (Battery charge)	Minor Unrecoverable Fault
5-12	Not Supported / Reserved
13 (Sampling warning)	Minor Recoverable Fault
14 (Checksum warning)	Minor Recoverable Fault
15 (Temperature sensor warning)	Minor Recoverable Fault

### 7.7.4. Mapping of alarm bits to device status

The following table shows the mapping of the alarm bits (see [Instance Attribute 44 – Alarms](#) of Position Sensor Object) to the device status (see [Attribute 5 \(Status\)](#) of Identity Object).

Alarm Bit(s)	Device Status
0 (Position error)	Major Recoverable Fault
1 (Diagnostic error)	Major Unrecoverable Fault
2-11	Not Supported
12 (Sampling error)	Major Recoverable Fault
13 (Internal CRC error)	Major Recoverable Fault
14 (Internal frame error)	Major Recoverable Fault
15 (Checksum Error)	Major Unrecoverable Fault

### 7.7.5. Device Status Prioritization

The system status (see [Attribute 5 \(Status\)](#) of Identity Object) is set depending on the most serious device status which is currently set in the warning and alarm bits (see [Instance Attribute 44 – Alarms](#) and [Instance Attribute 47 – Warnings](#) of Position Sensor Object). The priority of possible values for the device status is as follows:

1. Major Unrecoverable Fault
2. Major Recoverable Fault
3. Minor Unrecoverable Fault
4. Minor Recoverable Fault
5. Fault Free (no warning or alarm bits are set)

## 7.8. Firmware update via webserver

The EAx580 EtherNet/IP encoder supports a webserver. By using this webserver it is possible to update the firmware of the EAx580 EtherNet/IP encoder.

## 7.9. Identity Object (0x01)

This object provides identification of and general information about the device.

### 7.9.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x01	Get_Attributes_All	Returns the contents of the instance or class attributes defined in the object definition
0x05	Reset	See table "Reset service" below

#### Reset service

The device supports reset types 0 and 1.

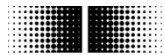
Reset Type	Description
0	Emulated power cycle reset (software restart)
1	<p>A customer factory reset is executed followed by an emulated power cycle reset. The encoder is reset to factory defaults.</p> <p>Caution: Even communication parameters (e.g. IP address) are reset to factory defaults.</p>

### 7.9.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.	1
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level.	1
6	Get	Maximum ID Number Class Attributes	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device.		7
7	Get	Maximum ID Number Instance Attributes	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device		9

### 7.9.3. Instance Services

Service Code	Service Name	Description	Link
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute	-



## 7.9.4. Instance Attributes

Attr ID	Access Rule	NV	Name	Data Type	Description of Attribute	Value
1	Get	NV	Vendor ID	UINT	Identification of each vendor by number	468
2	Get	NV	Device Type	UINT	Indication of general type of product	0x22
3	Get	NV	Product Code	UINT	Identification of a particular product of an individual vendor	
4	Get	NV	Revision	Struct of:	Revision of the item the Identity Object represents	
			Major Revision	USINT		
			Minor Revision	USINT		
5	Get	V	Status	WORD	Summary status of device	Default: 0 (depending on connection state)
6	Get	NV	Serial Number	UDINT	Serial number of device	
7	Get	NV	Product Name	SHORT_STRING	Human readable identification	
8	Get	V	State	USINT	Present state of the device as represented by the state transition diagram	Default: 0 (depending on device status)
9	Get	NV	Configuration Consistency Value	UINT	Contents identify configuration of device	0

### Attribute 5 (Status)

The meaning of the bits of attribute 5 is described in the following table.

Bit(s)	Description
0-3	Not Supported (= 0)
4-7	Extended Device Status (see table below)
8	Minor Recoverable Fault: TRUE indicates the device detected a problem with itself, which is thought to be recoverable. The problem does not cause the device to go into one of the faulted states
9	Minor Unrecoverable Fault: TRUE indicates the device detected a problem with itself, which is thought to be unrecoverable. The problem does not cause the device to go into one of the faulted states.
10	Major Recoverable Fault: TRUE indicates the device detected a problem with itself, which caused the device to go into the Major Recoverable Fault state.
11	Major Unrecoverable Fault: TRUE indicates the device detected a problem with itself, which caused the device to go into the Major Unrecoverable Fault state.
12-15	0

The meaning of bits 4 to 7 ("Extended Device Status") of attribute 5 is described in the following table.

Value	Description
0	Self-Testing or unknown (not supported)
1	Firmware update in progress (not supported)
2	At least one faulted I/O connection (not supported)
3	No I/O connections established
4	Non-Volatile configuration bad (not supported)
5	Major fault (Bit 10-11 or'ed)
6	At least one I/O connection in run mode (not supported, no run idle data in assemblies)
7	At least one I/O connection established, all in idle mode
8	0
9	Reserved
10-15	0 (not supported)

### Attribute 8 (State)

The meaning of the bits of attribute 8 is described in the following table.

Value	Description
0	Not existent
1	Device Self Testing
2	Standby
3	Operational
4	Major Recoverable Fault
5	Major Unrecoverable Fault
6-254	Reserved
255	Not supported

### 7.10. Message Router Object (0x02)

The Message Router Object provides a messaging connection point through which a client may address a service to any object class or instance residing in the physical device. Since the message router does not have any class or instance attributes there are no services supported.

### 7.11. Assembly Object (0x04)

The Assembly Object binds attributes of multiple objects which allows data to or from each object to be sent or received over a single connection. Assembly Objects can be used to bind input data or output data. The terms "input" and "output" are defined from the network's point of view. An input will produce data on the network and an output will consume data from the network.

#### 7.11.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute

#### 7.11.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is two (02). If updates that require an increase in this value are made, then the value of this attribute increases by 1.	2



### 7.11.3. Instance Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x10	Set_Attribute_Single	Modifies an attribute value (only for consumed assemblies, therefore no attribute to use on)

### 7.11.4. Instance Attributes

Attr ID	Access Rule	NV	Name	Data Type	Description of Attribute
3	Set	V	Data	ARRAY of BYTE	Data of the according assembly instance.
4	Get	NV	Size	UINT	Number of bytes in Attribute 3

### 7.11.5. Instances

The device supports the following assembly instances.

Instance	Attr ID	Access	Description	Bits	Bytes
1	3	Get	Input Assembly 1: Position	32	4
2	3	Get	Input Assembly 2: Position + Warning/Alarm Flags	32 8	5
3	3	Get	Input Assembly 3: Position + Velocity	32 32	8
110	3	Get	Input Assembly 110: Position + Velocity + Warning/Alarm Flags	32 32 8	9
105	3	Get	Config Assembly 105: Encoder Parameter	256	32
107	3	Get	Config Assembly 107: Empty	0	0

### Forward Open Assembly Check

In case of a second IO connection establishment (Forward Open) it is checked that the new connection does not change the current encoder parameter (through [Instance \(105\) - Configuration](#)). In case of differing parameters the following result code is sent and the connection is rejected.

General Result Code	Extended Status
Object State Conflict (0x0C)	0

In the following details about the instances are described.

### Instance (1) - Position Value

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Position LSB							
1	Position							
2	Position							
3	Position MSB							

Value Mapping:

Data Component Name	Mapped Class	Mapped Attribute	Details
Position Value	Position Sensor	Position Value (3)	<u>Instance Attribute 3 - Position Value Unsigned</u>

### Instance (2) - Position + Warning/Alarm Flags

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Position LSB							
1	Position							
2	Position							
3	Position MSB							
4	0	0	0	0	0	0	Warning Flag	Alarm Flag

Value Mapping:

Data Component Name	Mapped Class	Mapped Attribute	Details
Position Value	Position Sensor	Position Value (3)	<u>Instance Attribute 3 - Position Value Unsigned</u>
Warning Flag	Position Sensor	Warning Flag (49)	<u>Instance Attribute 49 - Warning Flag</u>
Alarm Flag	Position Sensor	Alarm Flag (46)	<u>Instance Attribute 46 - Alarm Flag</u>

### Instance (3) - Position + Velocity

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Position LSB							
1	Position							
2	Position							
3	Position MSB							
4	Velocity LSB							
5	Velocity							
6	Velocity							
7	Velocity MSB							

Value Mapping:

Data Component Name	Mapped Class	Mapped Attribute	Details
Position Value	Position Sensor	Position Value (3)	<a href="#">Instance Attribute 3 - Position Value Unsigned</a>
Velocity Value	Position Sensor	Velocity Value (24)	<a href="#">Instance Attribute 24 - Velocity Value</a>

### Instance (110) - Position + Velocity + Warning/Alarm Flags

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Position LSB							
1	Position							
2	Position							
3	Position MSB							
4	Velocity LSB							
5	Velocity							
6	Velocity							
7	Velocity MSB							
8	0	0	0	0	0	0	Warning Flag	Alarm Flag

Value Mapping:

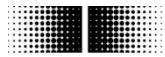
Data Component Name	Mapped Class	Mapped Attribute	Details
Position Value	Position Sensor	Position Value (3)	<a href="#">Instance Attribute 3 - Position Value Unsigned</a>
Velocity Value	Position Sensor	Velocity Value (24)	<a href="#">Instance Attribute 24 - Velocity Value</a>
Warning Flag	Position Sensor	Warning Flag (49)	<a href="#">Instance Attribute 49 - Warning Flag</a>
Alarm Flag	Position Sensor	Alarm Flag (46)	<a href="#">Instance Attribute 46 - Alarm Flag</a>

### Instance (105) - Configuration

Byte	Parameter	Details
0	Direction Counting Toggle	<a href="#">Instance Attribute 12 - Direction Counting Toggle</a>
1	Scaling Function Control	<a href="#">Instance Attribute 14 - Scaling Function Control</a>
2-5	Measuring Units per Span	<a href="#">Instance Attribute 16 - Measuring Units per Span</a>
6-9	Total Measuring Range	<a href="#">Instance Attribute 17 - Total Measuring Range in Measuring Units</a>
10-11	Velocity Format	<a href="#">Instance Attribute 25 - Velocity Format</a>
12-15	Velocity Resolution	<a href="#">Instance Attribute 26 - Velocity Resolution</a>
16	Velocity Sample Rate	<a href="#">Instance Attribute 100 - Velocity Sample Rate</a>
17	Velocity Filter	<a href="#">Instance Attribute 101 - Velocity Filter</a>
18-19	Gear Factor Activation	<a href="#">Instance Attribute 102 - Gear Factor Activation</a>
20-23	Gear Factor Numerator	<a href="#">Instance Attribute 103 - Gear Factor Numerator</a>
24-27	Gear Factor Denominator	<a href="#">Instance Attribute 104 - Gear Factor Denominator</a>
28-31	Preset Request Value	<a href="#">Instance Attribute 107 - Preset Request Value</a>

For invalid parameter combinations the following data is transmitted. The “Extended Status” contains the byte number of the parameter which is responsible for failing.

Affected Parameter	Description	General Result Code	Extended Status (Byte index of the parameter in configuration assembly)
-	No error	-	-
Rotating Direction (CCW)	Parameter CCW invalid	0x03 (Invalid Parameter Value)	0
Scaling	Parameter Scaling invalid	0x03 (Invalid Parameter Value)	1
Steps per Revolution	Minimum value violated	0x03 (Invalid Parameter Value)	2
Steps per Revolution	Maximum value violated	0x03 (Invalid Parameter Value)	3
Total Measurement Range (TMR)	Minimum value violated	0x03 (Invalid Parameter Value)	6



Total Measurement Range (TMR)	Maximum value violated	0x03 (Invalid Parameter Value)	7
Total Measurement Range (TMR)	Parameter "TMR" > ("steps per revolution" * "physical multiturn range")	0x03 (Invalid Parameter Value)	8
Gear Factor Mode	Maximum value violated	0x03 (Invalid Parameter Value)	18
Gear Factor Numerator	Minimum value violated	0x03 (Invalid Parameter Value)	20
Gear Factor Numerator	Maximum value violated	0x03 (Invalid Parameter Value)	21
Gear Factor Denominator	Minimum value violated	0x03 (Invalid Parameter Value)	24
Gear Factor Denominator	Maximum value violated	0x03 (Invalid Parameter Value)	25
Gear Factor Numerator Gear Factor Denominator	$((\text{TMR} * \text{"Gear Factor Denominator"}) / \text{"Gear Factor Numerator"}) > 2^{(32 - [\text{"physical multiturn length"}])}$	0x03 (Invalid Parameter Value)	26
Velocity Unit	Maximum value violated	0x03 (Invalid Parameter Value)	10
Velocity Resolution Steps	Parameter invalid	0x03 (Invalid Parameter Value)	12
Velocity Resolution Turns	Parameter invalid	0x03 (Invalid Parameter Value)	13
Velocity Update Period	Minimum range violated	0x03 (Invalid Parameter Value)	16
Velocity Filter Depth	Minimum value violated	0x03 (Invalid Parameter Value)	17
Set Preset Request Value	Maximum value violated ( $\geq$ TMR)	0x03 (Invalid Parameter Value)	28
Set Preset 32Bit	Maximum value violated ( $\geq$ TMR)	-	255 (general fault code)

## Instance (107) - Empty Configuration

The device supports an empty configuration assembly which is used for each connection.

### Remark:

The empty configuration assembly is needed to check that no parameters are changed at a second IO connection.

## 7.12. Connection Manager Object (0x06)

The Connection Manager Class allocates and manages the internal resources associated with both I/O and Explicit Messaging Connections. The specific instance generated by the Connection Manager Class is referred to as a Connection Instance or a Connection Object.

### 7.12.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute

### 7.12.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01). If updates that require an increase in this value are made, then the value of this attribute increases by 1.	1
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level.	1

### 7.12.3. Instance Services

Instance services for the connection manager object are not supported.

### 7.12.4. Instance Attributes

Instance attributes for the connection manager object are not supported.

### 7.13. Position Sensor Object (0x23)

Encoder parameters are defined in the “Position Sensor Object” class. Depending on their access rule the parameters can be read or written via explicit messaging (acyclic messages).

#### 7.13.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x05	Reset	See below.
0x0D	Apply_Attributes	See <a href="#">Default Parameter Behaviour</a>
0x15	Restore	See <a href="#">Default Parameter Behaviour</a>
0x16	Save	See <a href="#">Default Parameter Behaviour</a>

#### **Reset service**

The device supports reset types 0 and 1.

Reset Type	Description
0	Emulated power cycle reset (software reset)
1	The reset of the encoder parameters (see <a href="#">Default Parameter Behaviour</a> ) is executed followed by an emulated power cycle reset.

### 7.13.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	-	2
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level.	1
3	Get	Number of Instances	UINT	Number of object instances currently created at this class level of the device.	The number of object instances at this class hierarchy level.	1
4	Get	Optional attribute list	Struct of	List of optional instance attributes utilized in an object class implementation.	A list of attribute numbers specifying the optional attributes implemented in the device for this class.	
		number of attributes	UINT	Number of attributes in the optional attribute list.	The number of attribute numbers in the list.	8
		optional attributes	ARRAY of UINT	List of optional attribute numbers	The optional attribute numbers.	3,11,14,16,17,19,42,50
6	Get	Maximum ID Number Class Attributes	UINT	The attribute ID number of the last class attribute of the class definition implemented in the device.		7
7	Get	Maximum ID Number Instance Attributes	UINT	The attribute ID number of the last instance attribute of the class definition implemented in the device		See <a href="#">Instance Attributes</a>

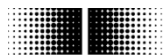
### 7.13.3. Instance Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x10	Set_Attribute_Single	Modifies an attribute value



### 7.13.4. Instance Attributes

Attribute ID	Access Rule	NV	Name	Data Type	Description of Attribute	Byte in Configuration Data	Details
3 <sup>(I)</sup>	Get	V	Position Value Unsigned	UDINT	Current Position		<a href="#">Instance Attribute 3 - Position Value Unsigned</a>
11	Get	NV	Position Sensor Type	UINT	Specifies the device type		<a href="#">Instance Attribute 11 - Position Sensor Type</a>
12 <sup>(C)</sup>	Set	NV	Direction Counting Toggle	BOOL	Defines the direction of increasing 'Position Value'	0	<a href="#">Instance Attribute 12 - Direction Counting Toggle</a>
14 <sup>(C)</sup>	Set	NV	Scaling Function Control	BOOL	Physical resolution span (attribute 42) is converted to a numerical value	1	<a href="#">Instance Attribute 14 - Scaling Function Control</a>
16 <sup>(C)</sup>	Set	NV	Measuring Units per Span	UDINT	Number of distinguishable steps per one complete span. Less than or equal to Physical Resolution Span (attribute 42)	2-5	<a href="#">Instance Attribute 16 - Measuring Units per Span</a>
17 <sup>(C)</sup>	Set	NV	Total Measuring Range in Measuring Units	UDINT	Steps over the total measuring range, only used for rotary encoders	6-9	<a href="#">Instance Attribute 17 - Total Measuring Range in Measuring Units</a>
19	Set	NV	Preset Value	DINT	Output position value is set to Preset Value		<a href="#">Instance Attribute 19 - Preset Value</a>
24 <sup>(I)</sup>	Get	V	Velocity Value	DINT	Current speed where the format of this value is defined in attributes 25 & 26		<a href="#">Instance Attribute 24 - Velocity Value</a>
25 <sup>(C)</sup>	Set	NV	Velocity Format	ENGUINT	Format of the velocity attributes	10-11	<a href="#">Instance Attribute 25 - Velocity Format</a>
26 <sup>(C)</sup>	Set	NV	Velocity Resolution	UDINT	Specifies the smallest incremental change of the Velocity Value attribute 24	12-15	<a href="#">Instance Attribute 26 - Velocity Resolution</a>
42	Get	NV	Physical Resolution Span	UDINT	Number of distinguishable steps per one complete span		<a href="#">Instance Attribute 42 - Physical Resolution Span</a>
44	Get	V	Alarms	WORD	Indicates a malfunction has occurred that could lead to an incorrect position value or require user intervention		<a href="#">Instance Attribute 44 - Alarms</a>
45	Get	NV	Supported Alarms	WORD	Information about supported Alarms		<a href="#">Instance Attribute 45 - Supported Alarms</a>
46 <sup>(I)</sup>	Get	V	Alarm Flag	BOOL	Indicates that an alarm error occurred		<a href="#">Instance Attribute 46 - Alarm Flag</a>
47	Get	V	Warnings	WORD	Internal parameters exceeded		<a href="#">Instance Attribute 47 - Warnings</a>
48	Get	NV	Supported Warnings	WORD	Information about supported Warnings		<a href="#">Instance Attribute 48 - Supported Warnings</a>
49 <sup>(I)</sup>	Get	V	Warning Flag	BOOL	Indicates that a warning error occurred		<a href="#">Instance Attribute 49 - Warning Flag</a>
50	Get	NV	Operating Time	UDINT	Stores operating time for the encoder in tenths of an hour		<a href="#">Instance Attribute 50 - Operating Time</a>
51	Get	NV	Offset Value	DINT	The Offset Value is calculated by the preset function. The position value is shifted with the calculated value.		<a href="#">Instance Attribute 51 - Offset Value</a>
100 <sup>(C)</sup>	Get / Set	NV	Velocity Sample Rate	USINT	Velocity sample rate in milliseconds	16	<a href="#">Instance Attribute 100 - Velocity Sample Rate</a>



101 <sup>(C)</sup>	Get / Set	NV	Velocity Filter	USINT	Number of measurements for the calculation of the moving average	17	<a href="#">Instance Attribute 101 - Velocity Filter</a>
102 <sup>(C)</sup>	Get / Set	NV	Gear Factor Activation	UINT	Enables or disables the gear factor functionality	18-19	<a href="#">Instance Attribute 102 - Gear Factor Activation</a>
103 <sup>(C)</sup>	Get / Set	NV	Gear Factor Numerator	UDINT	Adaption of the Gear Factor Numerator	20-23	<a href="#">Instance Attribute 103 - Gear Factor Numerator</a>
104 <sup>(C)</sup>	Get / Set	NV	Gear Factor Denominator	UDINT	Adaption of the Gear Factor Denominator	24-27	<a href="#">Instance Attribute 104 - Gear Factor Denominator</a>
105	Get	NV	Diagnostic Logbook	ARRAY of USINT	Returns the diagnostic logbook		<a href="#">Instance Attribute 105 - Diagnostic Logbook</a>
106	Get	V	Temperature	SINT	Returns the current temperature		<a href="#">Instance Attribute 106 - Device Temperature</a>
107 <sup>(C)</sup>	Get / Set	NV	Preset Request	UDINT	Preset Request Value (Preset button)	28-31	<a href="#">Instance Attribute 107 - Preset Request Value</a>
108	Get	V	Raw Position Value	UDINT	Raw Position Value		<a href="#">Instance Attribute 108 - Raw Position</a>
109	Get	NV	Number of Spans	UDINT	Number of turns		<a href="#">Instance Attribute 109 - Number of Spans</a>
110	Get / Set	NV	Set Parameter Non-Volatile	USINT	Controls the kind of setting parameters		<a href="#">Instance Attribute 110 - Set Parameter Non-Volatile</a>

<sup>(C)</sup> Attributes are mapped in Configuration Data Assembly Instance, see [Configuration data](#). Editing the value of any attribute has to be applied by “Apply Service” to be effective.

<sup>(I)</sup> Attributes are mapped in Input Data Assembly Instances, see [Input Data](#).

The minimum, maximum and default values can be found in the EDS files.

The following chapters contain more details on the instance attributes of the Position Sensor Object.

### *Instance Attribute 3 - Position Value Unsigned*

This instance attribute contains the absolute position of the sensor. A zero correction of the preset function is taken into consideration in the displayed value. This means that if the value was modified by an executed preset the effect of the preset is considered. The unit of the position value is increments or scanning steps or counts.

### *Instance Attribute 11 - Position Sensor Type*

This instance attribute can have one of the following values:

- 01 – Singleturn absolute encoder
- 02 – Multiturn absolute encoder

### *Instance Attribute 12 - Direction Counting Toggle*

This instance attribute sets the rotating direction. It can be cw (clockwise) or ccw (counterclockwise). The instance attribute can have one of the following values:

- If bit is not set the position value increases if the shaft is rotated clockwise (looking at the shaft).
- If bit is set the position value increases if the shaft is rotated counterclockwise (looking at the shaft).

### *Instance Attribute 14 - Scaling Function Control*

This instance attribute is used to enable or disable the scaling of the position calculation. The instance attribute can have one of the following values:

- If bit is not set scaling of the position value is disabled.
- If bit is set scaling of the position value is enabled.

### *Instance Attribute 16 - Measuring Units per Span*

This instance attribute contains the desired singleturn resolution within the range from 1 to maximum encoder resolution. The attribute defines the number of distinguishable steps per revolution of the sensor. Values between 1 and the maximum resolution of the encoder per revolution (attribute 42) are admissible. The following values are possible.

Value range	1..8192 (for "MT16ST13" device variant) 1..16384 (for "MT16ST14" device variant) 1..262144 (for "MT13ST18" device variant) all other values not allowed
Default value	8192 (for "MT16ST13" device variant) 16384 (for "MT16ST14" device variant) 262144 (for "MT13ST18" device variant)

### *Instance Attribute 17 - Total Measuring Range in Measuring Units*

This instance attribute defines the total number of distinguishable steps over the entire measurement range.

The minimum possible value is calculated as:

*Minimum value of attribute 17 = value of attribute 16*

The maximum possible value is calculated as:

*Maximum value of attribute 17 = value of attribute 16 x value of attribute 109*

Reparameterization deletes the previous offset value (attribute 51) so that the previous position reference is lost. The following values are possible.

Value range	2..536870912 (for "MT16ST13" device variant) 2..1073741824 (for "MT16ST14" device variant) 2..2147483648 (for "MT13ST18" device variant)
Default value	536870912 (for "MT16ST13" device variant) 1073741824 (for "MT16ST14" device variant) 2147483648 (for "MT13ST18" device variant)

The total measurement range (TMR) indicates the maximum steps the encoder delivers. The maximum possible position value is TMR - 1.

### *Instance Attribute 19 - Preset Value*

This instance attribute contains the desired absolute preset value. Writing this object executes a preset. The encoder position will immediately be set to the absolute position value given in instance attribute 19. The encoder internally calculates a preset offset value (attribute 51) which is being stored in a non-volatile memory.

*Preset value (attribute 19) = position value (attribute 3) + offset value (attribute 51)*

**Note: The preset function should only be used when the encoder is at a standstill.**

The preset can be selected in a range between zero and a value smaller than the set overall measurement range (attribute 17).

For more details please see chapter [Parameterization sequence](#).

### *Instance Attribute 24 - Velocity Value*

This instance attribute contains the current velocity value of the encoder.

### *Instance Attribute 25 - Velocity Format*

This instance attribute contains the format in which the current velocity value of the encoder is displayed.

The following two values are possible:

- 0x1F05 (counts per millisecond, CPMS)
- 0x1F0F (revolution per minute, rpm)

The device supports counts per millisecond (0x1F05) with a predefined factor (per 10 milliseconds, 100 milliseconds or 1000 milliseconds). The factor is adjusted via attribute 26.

### *Instance Attribute 26 - Velocity Resolution*

This instance attribute contains the factor for the velocity format (attribute 25). The factor is called "Supported resolutions" in the following table.

Unit	Supported resolutions
0x1F05 (counts per ms)	10, 100, 1000
0x1F0F (turns per minute)	1

### *Instance Attribute 42 - Physical Resolution Span*

This instance attribute contains the physical resolution of the encoder. It can be read out in the form of scanning steps per revolution. The value can also be interpreted as number of distinguishable steps per one complete span. For rotary devices a span equals one revolution.

Possible values are:

8192 = 0x2000 (for "MT16ST13" device variant)

16384 = 0x4000 (for "MT16ST14" device variant)

262144 = 0x40000 (for "MT13ST18" device variant)

### *Instance Attribute 44 – Alarms*

This instance attribute contains 16 bits. Some of them represent alarm events. Others are reserved by CIP. The following alarms are supported:

Bit	Description
0	Position error
1	Diagnostic error
2-11	Reserved by CIP
12	Sampling error
13	Internal CRC error
14	Internal frame error
15	Checksum error

An alarm is set if the encoder has detected a status which can result in an incorrect encoder position. As soon as an alarm status is detected the relevant bit is set to logical high.

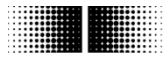
### *Instance Attribute 45 - Supported Alarms*

This instance attribute contains the information which alarms are supported by the encoder. The following alarms are supported:

Bit	Description	Supported
0	Position error	Yes
1	Diagnostic error	Yes
2-11	Reserved by CIP	-
12	Sampling error	Yes
13	Internal CRC error	Yes
14	Internal frame error	Yes
15	Checksum error	Yes

### *Instance Attribute 46 - Alarm Flag*

This instance attribute indicates that an alarm has occurred. This attribute is the logical OR of all alarm bits in Instance Attribute 44 – Alarms.



## *Instance Attribute 47 – Warnings*

This instance attribute contains 16 bits. Some of them represent warning events. Others are reserved by CIP or not supported. The following warnings are supported:

Bit	Description
0	Frequency Exceeded
1	Light Control Reserve
2	Not Supported = 0
3	Not Supported = 0
4	Battery Charge
5	Not Supported = 0
6	Not Supported = 0
7	Not Supported = 0
8	Not Supported = 0
9	Not Supported = 0
10	Not Supported = 0
11-12	Reserved By CIP
13	Sampling Warning
14	Checksum Warning
15	Temperature Sensor Warning

In contrast to alarms warnings do not indicate an incorrect position. The warning flag (attribute 49) is also set with each warning.

### *Instance Attribute 48 - Supported Warnings*

This instance attribute contains the information which warnings are supported by the encoder. The following warnings are supported:

Bit	Description	Supported
0	Frequency Exceeded	Yes
1	Light Control Reserve	Yes
2	CPU Watchdog	No
3	Operating Time Limit Warning	No
4	Battery Charge	Yes
5	Reference Point	No
6	Minimum Velocity Flag	No
7	Maximum Velocity Flag	No
8	Minimum Acceleration Flag	No
9	Maximum Acceleration Flag	No
10	Position Limits Exceeded	No
11-12	Reserved By CIP	No
13	Sampling Warning	Yes
14	Checksum Warning	Yes
15	Temperature Sensor Warning	Yes

### *Instance Attribute 49 - Warning Flag*

This instance attribute indicates that a warning has occurred. This attribute is the logical OR of all warnings bits in [Instance Attribute 47 – Warnings](#).

### *Instance Attribute 50 - Operating Time*

This instance attribute provides the total operation time of the device in tenths of an hour (6 minutes).

### *Instance Attribute 51 - Offset Value*

This instance attribute contains the preset offset of the encoder. The value of this attribute is calculated when [Instance Attribute 19 - Preset Value](#) is written or when a preset is triggered via the push button.



### *Instance Attribute 100 - Velocity Sample Rate*

This instance attribute contains the update period for the speed calculation in milliseconds.

Value range	1 to 255 (milliseconds)
Default value	16 (milliseconds)

The speed calculation is done cyclically with the update period. With a higher update period the speed value jitter can be reduced. The shortest update time is 1 millisecond.

### *Instance Attribute 101 - Velocity Filter*

This instance attribute represents the speed filter depth. The number of elements of the arithmetic average filter for the speed calculation is configured with this instance attribute.

Value range	0x01 to 0xFF
Default value	0x05

### *Instance Attribute 102 - Gear Factor Activation*

This instance attribute is used to enable or disable the gear factor functionality.

Value range	<p>0x0000: gear factor functionality is inactive</p> <p>Instance Attribute 103 - Gear Factor Numerator and Instance Attribute 104 - Gear Factor Denominator are ignored.</p> <p>0x0001: gear factor functionality is active</p> <p>Instance Attribute 103 - Gear Factor Numerator and Instance Attribute 104 - Gear Factor Denominator are used and not ignored.</p> <p>All other values are not allowed.</p>
Default value	0x0000

### *Instance Attribute 103 - Gear Factor Numerator*

This instance attribute is used to set the numerator of the gear factor of the encoder.

Value range	1..8192 (for "MT16ST13" device variant) 1..4096 (for "MT16ST14" device variant) 1..16384 (for "MT13ST18" device variant)  All other values not allowed.
Default value	8192 (for "MT16ST13" device variant) 4096 (for "MT16ST14" device variant) 16384 (for "MT13ST18" device variant)

### *Instance Attribute 104 - Gear Factor Denominator*

This instance attribute is used to set the denominator of the gear factor of the encoder.

Value range	1..65535 (for "MT16ST13", "MT16ST14" and "MT13ST18" device variants)  All other values not allowed.
Default value	1 (for "MT16ST13", "MT16ST14" and "MT13ST18" device variants)

### *Instance Attribute 105 - Diagnostic Logbook*

The logbook contains diagnostic data.

### *Instance Attribute 106 - Device Temperature*

This instance attribute contains the device temperature value in the value range from -128 to 127 degrees Celsius.

### *Instance Attribute 107 - Preset Request Value*

This instance attribute contains the value which will be used as preset value when a preset is triggered by the push button.

Value range	0 to total measuring range -1  Content of <u>Instance Attribute 17 - Total Measuring Range in Measuring Units</u>
-------------	---

### *Instance Attribute 108 - Raw Position*

This instance attribute contains the raw position value (without the influence of scaling).

### *Instance Attribute 109 - Number of Spans*

This instance attribute contains the maximum number of distinguishable revolutions.

Default value	65536 = 0x10000 (for "MT16ST13" device variant) 65536 = 0x10000 (for "MT16ST14" device variant) 8192 = 0x2000 (for "MT13ST18" device variant)
---------------	---

The physical measurement range is calculated as follows:

*Physical measurement range = attribute 42 (Physical Resolution Span) x attribute 109 (Number of Spans)*

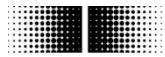
### *Instance Attribute 110 - Set Parameter Non-Volatile*

This instance attribute defines the kind of setting parameters. Depending on the value one of the following behaviours is active: Default Parameter Behaviour / Alternative Parameter Behaviour. The parameter is set non-volatile.

### 7.13.5. Error codes (Position Sensor Object)

When a write access to the Position Sensor Object was not successful the encoder sends a negative response. The following table contains possible error codes for this case.

Affected Parameter	Description	General Result Code	Extended Status
-	No error	-	0
Rotating Direction (CCW)	Parameter invalid	0x9 (Invalid attribute value)	12
Scaling	Parameter invalid	0x9 (Invalid attribute value)	14
Steps per Revolution (MeasUnits)	Minimum value violated	0x9 (Invalid attribute value)	16
Steps per Revolution (MeasUnits)	Maximum value violated	0x9 (Invalid attribute value)	16
Total Measurement Range (TMR)	Minimum value violated	0x9 (Invalid attribute value)	17
Total Measurement Range (TMR)	Maximum value violated	0x9 (Invalid attribute value)	17
Total Measurement Range (TMR)	Parameter "TMR" > ("steps per revolution" * "physical multiturn range")  See note below	0x3 (Invalid parameter value)	17
Gear Factor Mode	Maximum value violated	0x9 (Invalid attribute value)	102
Gear Factor Numerator	Minimum value violated	0x9 (Invalid attribute value)	103
Gear Factor Numerator	Maximum value violated	0x9 (Invalid attribute value)	103
Gear Factor Denominator	Minimum value violated	0x9 (Invalid attribute value)	104
Gear Factor Denominator	Maximum value violated	0x9 (Invalid attribute value)	104
Gear Factor Numerator Gear Factor Denominator	$((\text{TMR} * \text{"Gear Factor Denominator"}) / \text{"Gear Factor Numerator"}) > 2^{(32 - [\text{number of multiturn bits}])}$  The number of multiturn bits is for example 16 for variant "MT16ST13".	0x9 (Invalid attribute value)	102
Velocity Unit	Maximum value violated	0x9 (Invalid attribute value)	25



Velocity Resolution Steps	Parameter invalid	0x9 (Invalid attribute value)	26
Velocity Resolution Turns	Parameter invalid	0x9 (Invalid attribute value)	26
Velocity Update Period	Minimum range violated	0x9 (Invalid attribute value)	100
Velocity Filter Depth	Minimum value violated	0x9 (Invalid attribute value)	101
Set Preset Request Value	Maximum value violated ( $\geq$ TMR)	0x9 (Invalid attribute value)	107
Set Preset 32Bit	Maximum value violated ( $\geq$ TMR)	0x9 (Invalid attribute value)	19

Note:

The physical multiturn range is equal to the value of Instance Attribute 109 - Number of Spans.

## 7.14. Device Level Ring Object (0x47)

The Device Level Ring (DLR) Object provides the configuration and status information interface for the DLR protocol. The DLR protocol is a layer 2 protocol that enables the use of an Ethernet ring topology. The DLR Object provides the CIP application-level interface to the protocol.

One instance of the DLR Object is implemented for each pair of DLR ring ports supported.

### 7.14.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute

### 7.14.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is three (03).	3

### 7.14.3. Instance Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x01	Get_Attributes_All	Returns the contents of the instance or class attributes defined in the object definition

#### 7.14.4. Instance Attributes

Attr ID	Access Rule	NV	Name	Data Type	Description of Attribute	Value
1	Get	V	Network Topology	USINT	Current network topology mode	0 indicates "Linear" 1 indicates "Ring"
2	Get	V	Network Status	USINT	Current status of network	0 indicates "Normal" 1 indicates "Ring Fault" 2 indicates "Unexpected Loop Detected" 3 indicates "Partial Network Fault" 4 indicates "Rapid Fault/Restore Cycle"
10	Get	NV	Active Supervisor Address	STRUCT of:	IP and/or MAC address of the active ring supervisor	
				UDINT	Supervisor IP Address	A Value of 0 indicates no IP Address has been configured for the device
				ARRAY of 6 USINTs	Supervisor MAC Address	Ethernet MAC address
12	Get	NV	Capability Flags	DWORD	Describes the DLR capabilities of the device	Bit 0: 0 Bit 1: 1 Bit 2-4: 0 Bit 5: 0 Bit 6: 0 Bit 7: 1 Bit 8-31: 0

### 7.15. Quality of Service Object (0x48)

Quality of Service (QoS) is a general term that is applied to mechanisms used to treat traffic streams with different relative priorities or other delivery characteristics. Standard QoS mechanisms include IEEE 802.1D/Q (Ethernet frame priority) and Differentiated Services (DiffServ) in the TCP/IP protocol suite.

The QoS Object provides a means to configure certain QoS-related behaviours in EtherNet/IP devices.

The QoS Object is required for devices that support sending EtherNet/IP messages with non-zero DiffServ code points (DSCP), or sending EtherNet/IP messages in 802.1Q tagged frames.

#### 7.15.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x01	Get_Attributes_All	Returns the contents of the instance or class attributes defined in the object definition
0x10	Set_Attribute_Single	Modifies an attribute value

#### 7.15.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is one (01).	1
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device	The largest instance number of a created object at this class hierarchy level	1

#### 7.15.3. Instance Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x10	Set_Attribute_Single	Modifies an attribute value by applying and saving the value to the non-volatile memory.



#### 7.15.4. Instance Attributes

Attr ID	Access Rule	NV	Name	Data Type	Description of Attribute	Value
1	Get/Set	NV	802.1Q Tag Enable	USINT	Enables or disables sending 802.1Q frames on CIP and IEEE 1588 messages	0
4	Get/Set	NV	DSCP Urgent	USINT	DSCP value for CIP transport class 0/1 Urgent priority messages	Default: 55
5	Get/Set	NV	DSCP Scheduled	USINT	DSCP value for CIP transport class 0/1 Scheduled priority messages	Default: 47
6	Get/Set	NV	DSCP High	USINT	DSCP value for CIP transport class 0/1 High priority messages	Default: 43
7	Get/Set	NV	DSCP Low	USINT	DSCP value for CIP transport class 0/1 low priority messages	Default: 31
8	Get/Set	NV	DSCP Explicit	USINT	DSCP value for CIP explicit messages (transport class 2/3 and UCM) and all other EtherNet/IP encapsulation messages	Default: 27

## 7.16. TCP/IP Interface Object (0xF5)

The TCP/IP Interface Object provides the mechanism to configure a device's TCP/IP network interface. Examples of configurable items include the device's IP address, network mask and gateway address.

The TCP/IP Interface Object provides an attribute that identifies the link-specific object for the associated physical communications interface. The link-specific object is generally expected to provide link-specific counters as well as any link-specific configuration attributes.

### 7.16.1. Class Services

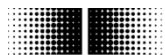
Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x01	Get_Attributes_All	Returns the contents of the instance or class attributes defined in the object definition
0x10	Set_Attribute_Single	Modifies an attribute value

### 7.16.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is four (04).	4
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device	The largest instance number of a created object at this class hierarchy level	1

### 7.16.3. Instance Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x10	Set_Attribute_Single	Modifies an attribute value by applying and saving the value to the non-volatile memory



## 7.16.4. Instance Attributes

Attr ID	Access Rule	NV	Name	Data Type	Description of Attribute	Value
1	Get	V	Status	DWORD	Interface Status	Default: 2
2	Get	NV	Configuration Capability	DWORD	Interface capability flags	Default: 0x000000B5 (DHCP, not stored)  See <a href="#">Instance Attribute 2 - Configuration Capability</a>
3	Get/Set	NV	Configuration Control	DWORD	Interface control flags	Default: 2, if rotary switch = 0  Default: 0, if rotary switch = 1
4	Get	NV	Physical Link Object	STRUCT of:	Path to physical link object	
			Path size	UINT	Size of Path	Default: 0  Number of 16 bit words in Path
			Path	Padded EPATH	Logical segments identifying the physical link object	Default: ""  The path is restricted to one logical class segment and one logical instance segment. The max. size is 12 bytes.
5	Get/Set	NV	Interface Configuration	STRUCT of:	TCP/IP network interface configuration	
			IP Address	UDINT	The device's IP address.	Default: None  Value of 0 indicates that no IP address has been configured.
			Network Mask	UDINT	The device's network mask	Default: 255.255.255.0  Value of 0 indicates that no network mask address has been configured.
			Gateway Address	UDINT	Default gateway address	Default: 0  Value of 0 indicates that no gateway address has been configured.
			Name Server	UDINT	Primary name server	Default: 0  Value of 0 indicates that

						no name server address has been configured.
			Name Server 2	UDINT	Secondary name server	Default: 0  Value of 0 indicates that no secondary name server address has been configured.
			Domain Name	STRING	Default domain name	Default: "" (ASCII characters)  Maximum length is 48 characters. A length of 0 indicates that no Domain Name is configured.
6	Get/Set	NV	Host Name	STRING	Host name	Default: "" (ASCII characters.)  Maximum length is 64 characters. A length of 0 indicates that no Host Name is configured.
10	Get/Set	NV	SelectAcd	BOOL	Activates the use of ACD	Default: 1  Enable ACD: 1 Disable ACD: 0
11	Get/Set	NV	LastConflictDetected	STRUCT of:	Structure containing information related to the last conflict detected	ACD Diagnostic Parameters
			AcdActivity	USINT	State of ACD activity when last conflict detected	Default: 0
			RemoteMAC	Array of 6 USINT	MAC address of remote node from the ARP PDU in which a conflict was detected	Default: 0
			ArpPdu	ARRAY of 28 USINT	Copy of the raw ARP PDU in which a conflict was detected.	Default: 0 * 28 ARP PDU
13	Get/Set	NV	Encapsulation Inactivity Timeout	UINT	Number of seconds of inactivity before TCP connection or DTLS session is close	Default: 120  0 = Disable 1-3600 = timeout in seconds

*Instance Attribute 2 - Configuration Capability*

Bit(s)	Called	Definition	Value
0	BOOTP Client	1 (TRUE) indicates that the device is capable of obtaining its network configuration via BOOTP.	1
1	DNS Client	1 (TRUE) indicates that the device is capable of resolving host names by querying a DNS server.	0
2	DHCP Client	1 (TRUE) indicates that the device is capable of obtaining its network configuration via DHCP.	1
3	DHCP-DNS Update	Set to 0	0
4	Configuration Settable	1 (TRUE) indicates that the Interface Configuration attribute is settable.	1
5	Hardware Configurable	1 (TRUE) indicates that the IP Address member of the Interface Configuration attribute can be obtained from hardware settings	1
6	Interface Configuration Change Requires Reset	1 (TRUE) indicates that the device requires a restart in order for a change to the Interface Configuration attribute to take effect. If this bit is FALSE a change in the Interface Configuration attribute will take effect immediately.	0
7	AcdCapable	(1) TRUE indicate that the device is ACD capable	1
8-31	Reserved	Reserved for future use and shall be set to zero.	0

## 7.17. Ethernet Link Object (0xF6)

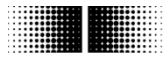
The Ethernet Link Object maintains link-specific counters and status information for an IEEE 802.3 communications interface. Each device supports exactly one instance of the Ethernet Link Object for each IEEE 802.3 communications interface on the module.

### 7.17.1. Class Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x01	Get_Attributes_All	Returns the contents of the instance or class attributes defined in the object definition
0x10	Set_Attribute_Single	Modifies an attribute value

### 7.17.2. Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description	Semantics of Values	Value
1	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is four (04).	4
2	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device	The largest instance number of a created object at this class hierarchy level	2
3	Get	Number of Instances	UINT	Number of object instances currently created at this class level of the device	The number of object instances at this class hierarchy level. This basically relates to the number of Ethernet ports the device supports	2



### 7.17.3. Instance Services

Service Code	Service Name	Description
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute
0x10	Set_Attribute_Single	Modifies an attribute value by applying and saving the value to the none volatile memory
0x4C	Get_and_Clear	Gets then clears the specified attribute

### 7.17.4. Instance Attributes

The details of the instance attributes can be found in chapter 5-5 of specification “The CIP Networks Library Volume 2”, Edition 1.23.

## **8. Troubleshooting – Frequently Asked Questions – FAQ**

### **8.1. FAQ: Project Work**

#### **8.1.1. Where do I get an encoder manual?**

The manual is available for download at [www.baumer.com](http://www.baumer.com) (freeware „Adobe Reader®“ required). Make sure the manual is the right one for your encoder by verifying the table at the beginning of the manual. You will find the encoder type on the product label (e.g. EAL580-xxx.xxEN-13160.x).

Should your encoder not be on the list please contact Baumer.

#### **8.1.2. Where do I get the applicable EDS file?**

The EDS file is available at [www.baumer.com](http://www.baumer.com). To find out if the EDS file is applicable for your encoder please use the table at the beginning of the manual. You will find the encoder type on the product label (e.g. EAL580-xxx.xxEN-13160.x).

Should your encoder not be on the list please contact Baumer.

### **8.2. FAQ: Operation**

#### **8.2.1. What is the significance of the LEDs provided at the encoder?**

The encoder integrates several LEDs indicating activity status of the encoder. Both link/activity LEDs visualize encoder activity on the bus, meaning data communication on each of the two Ethernet ports. In particular upon commissioning and in case of error the LEDs provide first information on the system status. For details on the respective status please refer to chapter [Diagnostic LEDs](#).

#### **8.2.2. How to adapt the resolution?**

The encoder resolution in steps per turn („measuring units per turn“) is programmed within the project by corresponding parameterization. Usually this is performed by the PLC's engineering tool (for example Studio 5000). The encoder resolution is programmable in individual steps between the maximum limit (encoder-specific, for example 8192 for EAL580-xxx.xxEN-13160.x) and the minimum limit of 1 step/turn.



## 9. Appendix A

### 9.1. Software Change Log (from firmware V1.003 to V1.004)

The following table contains important changes of firmware version (V1.004 compared to V1.003).

Change	Description
[Parameterization] Different parameterization behaviour possible (Instance Attribute 110)	By the new Instance Attribute 110 the behaviour of the acyclic encoder parametrization can be configured.  Details can be found here: <ul style="list-style-type: none"><li>- <a href="#">Parameterization behaviour</a></li><li>- <a href="#">Instance Attribute 110 - Set Parameter Non-Volatile</a></li></ul>