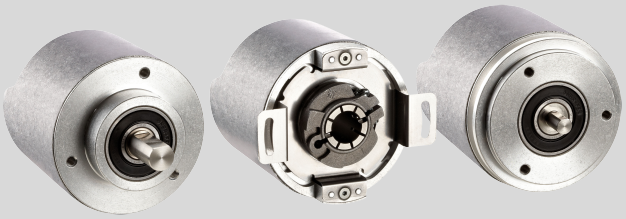
OPERATING INSTRUCTIONS

AFS/AFM60 EtherNet/IP

Absolute encoder

SICK
Sensor Intelligence.



Described product

AFS/AFM60 EtherNet/IP

Manufacturer

SICK AG
Erwin-Sick-Str. 1
79183 Waldkirch
Germany

Legal information

Original document

This document is an original document of SICK AG.

Contents

1 About this document

1.1 Scope

NOTE

These operating instructions apply to the AFS60/AFM60 EtherNet/IP absolute encoder with the following type designations:

- Singleturn encoder = AFS60A-xxIx262144
- Multiturn encoder = AFM60A-xxIx018x12

1.2 Purpose of this document

These operating instructions instruct the technical personnel of the machine manufacturer or machine operator in:

- Electrical installation
- Commissioning
- Parameterization
- Operation
- Maintenance

These operating instructions must be made available to all persons who work with the encoder.

The official and legal regulations for operating the encoder must always be complied with.

1.3 Target group

These operating instructions are intended for planning engineers, developers, and operators of plants and systems into which one or more AFS60/AFM60 EtherNet/IP absolute encoders are to be integrated. They are also intended for people who put the encoder into operation for the first time or who are in charge of maintenance.

These instructions are written for trained persons who are responsible for the installation, mounting and operation of the encoder in an industrial environment.

Only trained electricians are permitted to carry out work on the electrical system or electrical assemblies.

NOTICE

▶ Read the operating instructions carefully and ensure that you have understood the contents completely before you work with the encoder.

1.4 Further information

These operating instructions do not contain any information on mounting, technical data and dimensional drawings. These are enclosed separately with the device or available via the Internet: www.sick.com

The following information is available via the Internet:

- Mounting instructions
- Data sheets
- CAD data for drawings and dimensional drawings
- Certificates (such as the EU declaration of conformity)

1.5 Symbols and document conventions

Safety notes

DANGER
A safety note informs you of real-world specifications for safely mounting and installing the absolute encoder.
This is intended to protect you against accidents.

- ▶ Read the safety notes carefully and follow them.

Information on property damage/general advice

NOTICE
Indicates important information and possible property damage.

NOTE
Indicates useful tips and recommendations.

Instructions

- ▶ Instructions requiring specific action are indicated by an arrow. Carefully read and follow the instructions for action.

1. The sequence of instructions for action is numbered.
2. Numbered instructions for action are to be followed in the given order.

Status indicators

LED symbols describe the status of a diagnostics LED. Examples:

 The LED is illuminated continuously.

 The LED is flashing.

 The LED is off.

1.6 Abbreviations used

CIP	Common Industrial Protocol
CMR	Counts per Measuring Range
CNR_D	Customized Number of Revolutions, Divisor = denominator of the customized number of revolutions
CNR_N	Customized Number of Revolutions, Nominator = nominator of the customized number of revolutions
CPR	Counts Per Revolution
DHCP	Dynamic Host Control Protocol
DLR	Device Level Ring
EADK	EtherNet/IP Adapter Developers Kit = development environment for EtherNet/IP devices
EDS	Electronic Data Sheet
EEPROM	Electrically Erasable Programmable Read-only Memory
FPGA	Field Programmable Gate Array = electronic component that can be programmed to form an application-specific circuit

I/O	Input and Output Data (from the point of view of the master)
IP in TCP/IP	Internet Protocol
IP in EtherNet/IP	Industrial Protocol
MAC	Media Access Control
ODVA	Open DeviceNet Vendor Association
PLC	Programmable Logic Controller
TCP	Transmission Control Protocol
UDP	User Datagram Protocol = connectionless network protocol

2 Safety information

2.1 General notes

DANGER

Observe the following to ensure the safe use of the AFS/AFM60 EtherNet/IP as intended.

The encoder must be installed and maintained by trained, qualified personnel with knowledge of electronics, precision engineering, and controller programming. The relevant technical safety standards must be observed.

All persons entrusted with the installation, operation, or maintenance of the devices must follow the safety guidelines:

- The operating instructions must always be available and must be followed.
- Unqualified personnel must stay away from the system during installation and maintenance.
- The system must be installed in accordance with the applicable safety regulations and mounting instructions.
- The work safety regulations of the employers' liability insurance associations and trade associations in the respective country must be observed during installation.
- Failure to observe the relevant work safety regulations may lead to physical injury or cause damage to the system.
- The current and voltage sources in the encoder are designed in accordance with the applicable technical guidelines.

2.2 Intended use

The Absolute encoder AFS/AFM60 EtherNet/IP is a measuring device which is manufactured according to the recognized industrial regulations and which meets the quality requirements stipulated in ISO 9001:2008 as well as those relating to environmental management systems as defined in ISO 14001:2009.

An encoder is designed for mounting and can only be operated according to its intended function. For this reason, the encoder is not equipped with direct safety devices.

The system designer must provide measures to ensure the safety of persons and systems in accordance with the legal guidelines.

Due to its design, the AFS/AFM60 EtherNet/IP may only be operated within an EtherNet/IP network. The EtherNet/IP specifications and the guidelines for setting up an EtherNet/IP network must be observed.

In the event of any other usage or modification to the AFS/AFM60 EtherNet/IP (e.g., due to opening the housing during mounting and electrical installation) or in the event of changes made to the SICK software, any claims against SICK STEGMANN GmbH under the warranty will be rendered void.

2.3 Requirements for the qualification of personnel

The encoder must only be mounted, commissioned, and maintained by authorized personnel.

NOTE

Repair work on the encoder may only be performed by qualified and authorized service personnel from SICK STEGMANN GmbH.

The following qualifications are necessary for the various tasks:

Table 1: Authorized personnel

Task	Qualification
Mounting (see mounting instructions)	• Basic practical technical training • Knowledge of the current safety regulations in the workplace
Electrical installation and device replacement	• Practical electrical training • Knowledge of current electrical safety regulations • Knowledge of the operation and control of the devices in their particular application (e.g., industrial robots, storage and conveyor systems)
Commissioning, operation, and configuration	• Knowledge of the current safety regulations and of the operation and control of the devices in their particular application • Knowledge of automation systems (e.g. Rockwell ControlLogix controller) • Knowledge of EtherNet/IP • Knowledge of the use of automation software (e.g. with Rockwell RSLogix)

3 Product description

3.1 Product identification

NOTICE
The year of construction of the absolute encoder can be found on the device label or on the packaging label. Keep the packaging for this reason.

Solid shaft type code

Table 2: Solid shaft type code

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	F	S	6	0	A	-	S								

Table 3: Solid shaft type code - explanation

Position	Meaning	Description
1	Product family	A = Absolute
2	Technology	F = Optical scanning with high resolution and accuracy
3	Type	S = Singleturn M = Multiturn
4	Size	60 = Outer diameter approx. 60 mm
5		
6	Step count [1]	A = Number of steps per revolution max. = 262,144 (18 bit)
7	-	-
8	Type	S = Solid shaft
9	Mechanical design	1 = Servo flange, 6 x 10 mm 4 = Face mount flange, 10 x 19 mm 5 = Face mount flange, 10 x 19 mm round 6 = Face mount flange, 3/8" round 7 = Face mount flange, 3/8"
10	Electrical interface	I = EtherNet/IP
11	Connection type	B = 3 x M12, 4-pin, axial
12	Resolution	Singleturn: Number of steps per revolution, can be freely programed by customer: Type A = 2 ... 262144 (factory setting) Multiturn: 018x12, type A: 18 bit (singleturn) x 12 bit multiturn (factory setting)
13		
14		
15		
16		

[1] Number of steps per revolution of programmable devices: Singleturn: Between 4 ... 262144. Programmable via programming tool and Safety Designer configuration software (www.sick.com).

Blind hollow shaft type code

Table 4: Blind hollow shaft type code

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	F	S	6	0	A	-	B								

Table 5: Blind hollow shaft type code – explanation

Position	Meaning	Description
1	Product family	A = Absolute
2	Technology	F =

Position	Meaning	Description
3	Type	S = Singleturn M = Multiturn
4	Size	60 = Outer diameter approx. 60 mm
5		
6	Step count [1]	A = Number of steps per revolution max. = 262,144 (18 bit)
7	-	-
8	Type	B = Blind hollow shaft
9	Mechanical design	B = Blind hollow shaft 8 mm C = Blind hollow shaft 3/8" D = Blind hollow shaft 10 mm E = Blind hollow shaft 12 mm F = Blind hollow shaft 1/2" G = Blind hollow shaft 14 mm H = Blind hollow shaft 15 mm J = Blind hollow shaft 5/8" K = Blind hollow shaft 1/4"
10	Electrical interface	I = EtherNet/IP
11	Connection type	B = 3 x M12, 4-pin, axial
12	Resolution	Singleturn: Number of steps per revolution, can be freely programed by customer: Type A = 2 ... 262144 (factory setting) Multiturn: 018x12, type A: 18 bit (singleturn) x 12 bit multiturn (factory setting)
13		
14		
15		
16		

1) Number of steps per revolution of programmable devices: Singleturn: Between 4 ... 262144. Programmable via programming tool and Safety Designer configuration software (www.sick.com).

3.2 Specific features

Table 6: Special features of the encoder variants

Features	Singleturn encoder	Multiturn encoder
Absolute encoder in 60 mm design	X	X
Rugged nickel code disk for harsh ambient conditions	X	X
High accuracy and availability	X	X
Large ball bearing distance of 30 mm	X	X
High vibration resistance	X	X
Optimum concentricity	X	X
Compact design	X	X
Face mount flange, servo flange and blind hollow shaft	X	X
18 bit singleturn resolution (1 to 262,144 steps)	X	X
30 bit total resolution		X
12-bit multiturn resolution (1 to 4,096 revolutions)		X
Round axis functionality		X

Features	Singleturn encoder	Multiturn encoder
EtherNet/IP interface (according to IEC 61784-1)	X	X
Supports encoder profile 22h defined in the CIP (Common Industrial Protocol)	X	X
Device level ring (DLR)	X	X

3.3 Operating principle of the encoder

The Absolute encoder detects the position and speed of rotary axes and outputs the position in the form of a unique digital numerical value. Optical detection takes place via an internal code disk.

The AFS60 EtherNet/IP is a singleturn encoder

Singleturn encoders are used when one shaft revolution must be detected absolutely.

The AFM60 EtherNet/IP is a multiturn encoder

Multiturn encoders are used when more than one shaft revolution must be detected absolutely.

3.3.1 Scalable resolution

The steps per revolution or the total resolution can be scaled and adapted to the respective application.

The steps per revolution are scalable from 1 ... 262,144 in whole numbers. The total resolution of the AFM60 EtherNet/IP must be 2^n-fold the steps per revolution. This restriction is not relevant if the round axis functionality is activated.

3.3.2 Preset function

A preset value can be used to set the position value of the encoder. I. e. the encoder can be set to any position within the measuring range. This allows, for example, the zero position of the encoder to be aligned with the machine zero point.

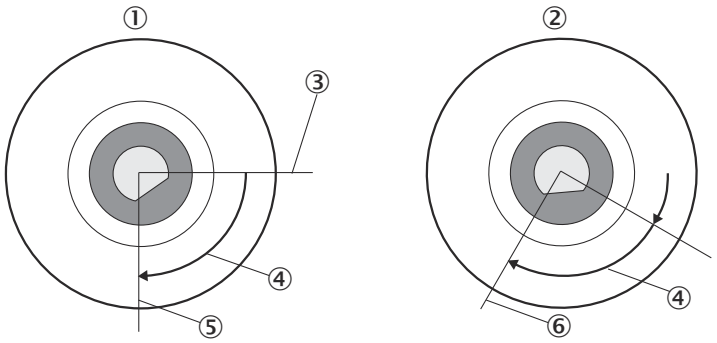


Figure 1: Setting a preset value

① Setting a preset value
② When switching on again
③ Actual position value
④ Offset
⑤ Position value after preset
⑥ Position value after switching on again

When the encoder is switched off, the offset – the delta between the real position value and the value specified by preset – is saved. When switching on again, the new calculated position value is formed from the new real position value and the offset. Even if the encoder was turned further during the switched-off state, the correct position value is output as a result.

3.3.3 Round axis functionality

The encoder supports the gear function for round axes. Here, the steps per revolution are set as a fraction (see "Preset function", page 40). This allows a number that is not 2^n-fold the steps per revolution or/and a decimal number (e.g. 12.5) to be configured as the total resolution.

NOTE
The output position value is calculated with a zero point correction, the set code sequence and the entered gear parameters.

Example with transmission ratio

A rotating table for filling bottles is to be controlled. The steps per revolution are specified by the number of fillers. There are nine fillers available. 1000 steps are required for precise measurement of the distance between two fillers.

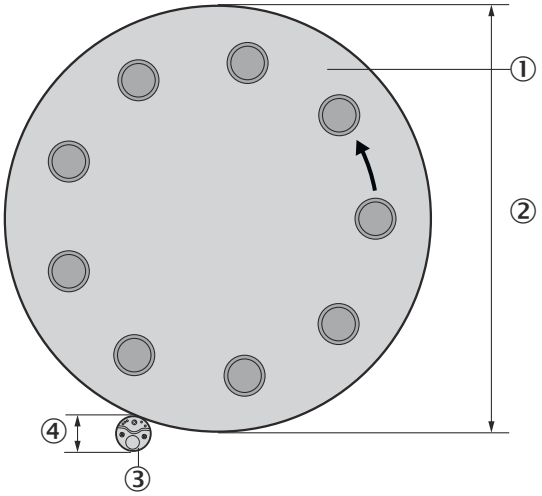


Figure 2: Example of position measurement on a rotating table with transmission ratio

① Rotating table with nine fillers
② Diameter of round table: 125 cm
③ Encoder mounted on an axis together with the drive wheel.
④ Diameter of drive wheel: 10 cm

The number of revolutions is given by the transmission ratio of the rotating table drive (125/10 = 12.5).

The total resolution is thus 9 × 1000 = 9000 steps, to be realized in 12.5 revolutions of the encoder. This ratio cannot be realized via the steps per revolution and the total resolution, since the total resolution is not 2^n-fold the steps per revolution.

The problem of the application can be solved with the round axis functionality. Here, the steps per revolution are disregarded. The total resolution and numerator and denominator of the number of revolutions are configured.

9000 steps are configured as the total resolution. The numerator of the number of revolutions is configured as 125, the denominator as 10 (125/10 = 12.5).

After 12.5 revolutions (i.e. after one complete revolution of the rotating table), the encoder reaches the total resolution of 9000.

Example without transmission ratio

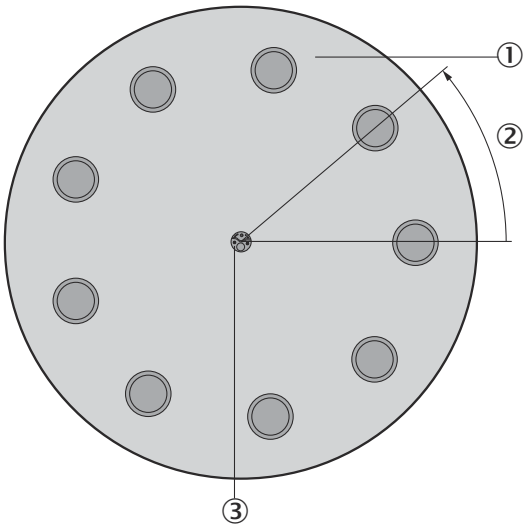


Figure 3: Example of position measurement on a rotating table without transmission ratio

① Rotating table with nine fillers
② 1000 steps
③ Encoder

The encoder is mounted directly on the shaft of the rotating table. The transmission ratio is 1:1.

The rotating table has 9 fillers. The encoder is to be configured so that it starts counting with 0 at a filler position and counts up to 999 until the next filler position.

1000 steps are configured as the total resolution.

1 is configured as the numerator of the number of revolutions, 9 as the denominator (1/9 revolutions = 1000).

After 1/9 revolutions of the encoder shaft there are 1000 steps, then the encoder starts counting again at 0.

3.4 Integration in EtherNet/IP

3.4.1 EtherNet/IP architecture

EtherNet/IP and thus also the AFS60/AFM60 EtherNet/IP uses Ethernet as transmission technology.

The network components are usually integrated in a **star or line structure**.

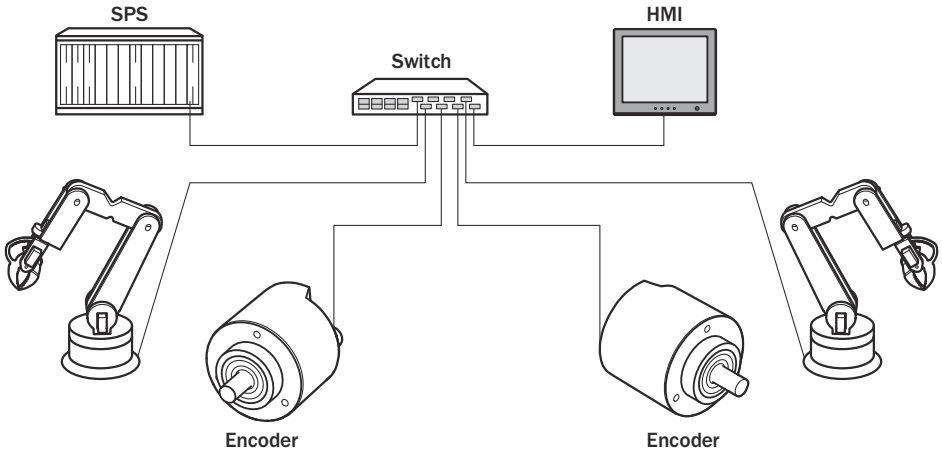


Figure 4: Example of an EtherNet/IP network in a star structure

However, to achieve greater availability and reduce the wiring work required, the system can also be integrated in a **device level ring** (**DLR**).

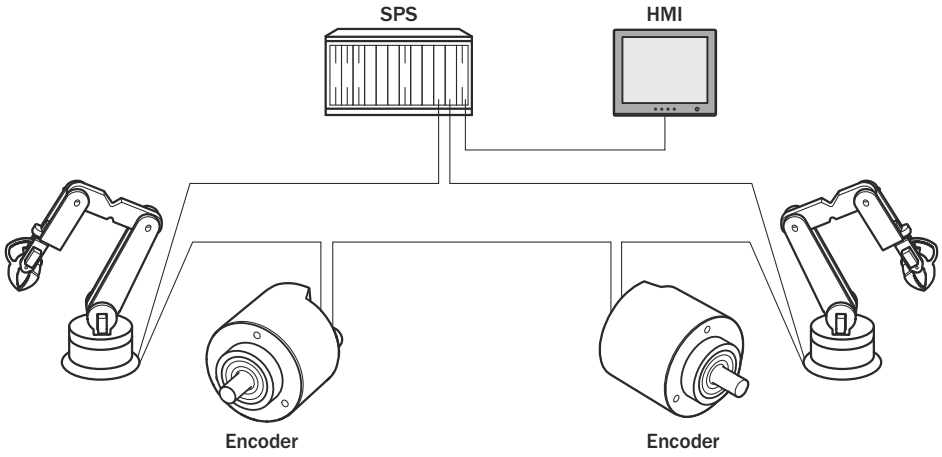


Figure 5: Example of an EtherNet/IP network in a device level ring

3.4.2 Communication in EtherNet/IP

MAC address

Each encoder is assigned a globally unique MAC address as device identification at the factory. This serves for the identification of the Ethernet node. This 6-byte device identification cannot be changed and consists of the following components:

- 3 byte ident number
- 3 byte device identifier

TCP/IP and UDP/IP

EtherNet/IP uses TCP/IP or UDP/IP for communication.

The IP address is necessary for identification. This is permanently entered for the encoder via address switches or obtained via DHCP server.

If the IP address is fixed, only the least significant byte can be set. 192.168.1.xxx is fixed.

In addition, the subnet mask (default = 255.255.255.0) and, if necessary, a gateway must be configured in the network.

Implicit messaging is used in EtherNet/IP for real-time communication between the controller and the encoder. Implicit messaging establishes a connection between exactly two devices within the CIP, e.g. to transmit I/O data such as position, speed, etc. from the encoder to the controller (see "Position sensor object", page 27). Implicit messaging uses **UDP/IP** over port 2222. It thus uses fast data throughput.

Explicit messaging is used in EtherNet/IP for communication that does **not** need to take place in real time. Explicit messaging uses **TCP/IP**, it is used, e.g., to transmit parameters from the controller to the encoder (see "Assembly object", page 22).

Common industrial protocol (CIP)

EtherNet/IP uses the CIP at the process level. This protocol is used to control processes in a similar way to how FTP is used to send files, for example.

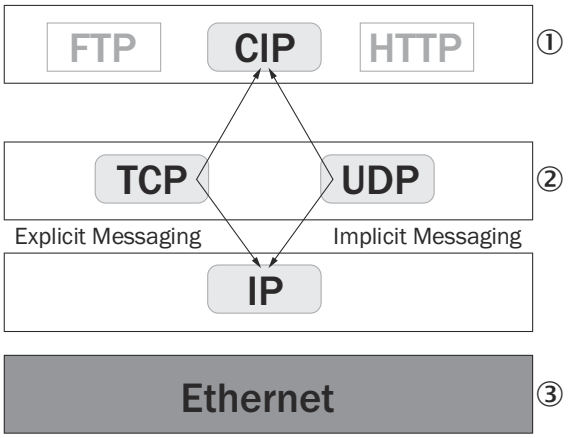


Figure 6: CIP and other services

① Process level
② Communication levels
③ Physical level

The absolute encoder complies with the guidelines of the EtherNet/IP protocol according to IEC 61784-1 and those of encoder profile 22h.

The encoder is an I/O adapter within the EtherNet/IP. It receives and sends explicit messages and implicit messages cyclically or on request.

EtherNet/IP communication

EtherNet/IP is based on the standard Ethernet frame. This contains the Ethernet header, the Ethernet data and the Ethernet trailer. The MAC addresses of the receiver (destination address) and the source (source address) are contained in the Ethernet header.

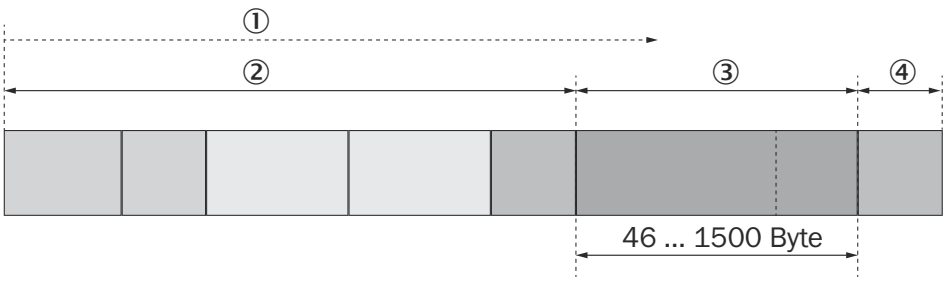


Figure 7: Ethernet frame

① Transmission order
② Header

③ Data field
④ Trailer

The Ethernet data field consists of different protocols that are nested within each other:

- The IP datagram is transported in the user data of the Ethernet data field.
- TCP segment or UDP datagram are transported in the user data of the IP datagram.
- The CIP protocol is transported in the user data of the TCP segment or UDP datagram.

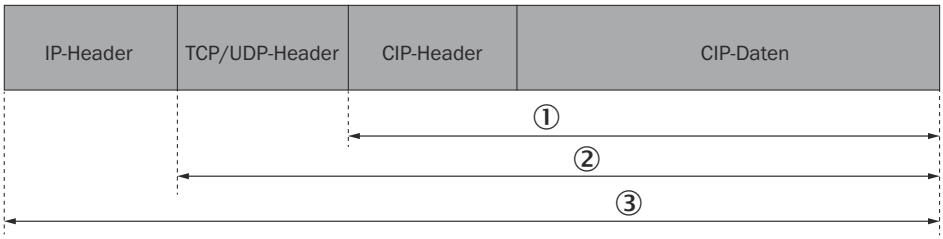


Figure 8: Ethernet data field

① CIP protocol
② TCP segment or UDP datagram
③ IP datagram

3.5 CIP object model

For network communication, EtherNet/IP uses an “object model”, in which all the functions and data of a device are defined.

The most important terms are explained below:

Class A class contains related objects of a device, organized into instances.

Instance An instance consists of various attributes, which describe the properties of this instance. Different instances in a class have the same services and the same attributes. They can, however, have different attribute values.

Attribute Attributes represent the data which a device makes available via EtherNet/IP. This data contains the current values of a configuration or an input, for example. Typical attributes are, e.g., configuration or status information.

Service Services are used to access classes or attributes of a class and to generate certain events. These services perform specified actions, e.g., reading attributes.

Table 7: Example CIP object model

	Class	Instance	Attribute	Value
Code	23h	1h	0Ah	3FFFFFFFh
Designation	Position sensor object	Class has one instance	Current position value	Example

3.5.1 Supported classes

The encoder supports the following classes of encoder profile 22h:

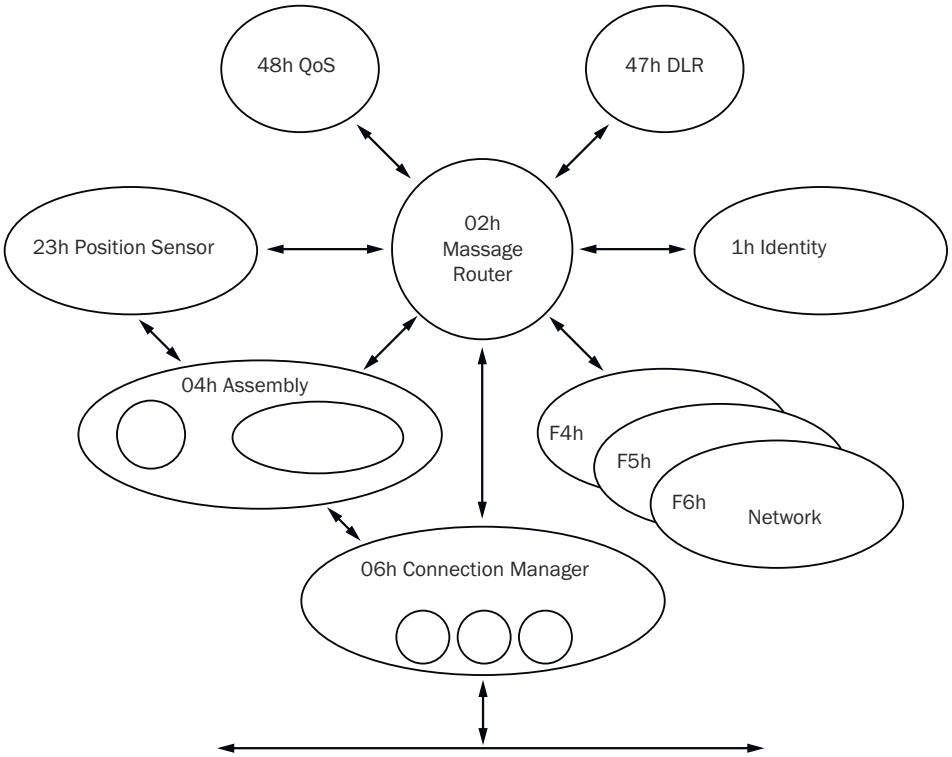


Figure 9: Supported classes

Table 8: Supported classes

Class code	Class	Description	Access	Instances
01h	Identity object	Contains all device-specific data (e.g., ID, device type, device status, etc.).	Get	1
02h	Message router object	Contains all supported class codes of the encoder and the max. number of connections.	Get	1
04h	Assembly object	Combines the data of multiple objects into a single object. Supplies e.g. the position value of the encoder.	Get	7
06h	Connection manager object	Contains connection-specific attributes for triggering, transport, connection type, etc.	Get	1
23h	Position sensor object	Contains all attributes for programming the encoder parameters such as scaling.	Set/Get	1
F4h	Port object	Contains the available ports, port name and node address.	Get	1
F5h	TCP/IP interface object	Contains the attributes for TCP/IP, such as IP address, subnet mask, and gateway or reference for the IP address via DHCP or hardware switch.	Set/Get	1
F6h	Ethernet link object	Contains connection-specific attributes, such as transmission speed, interface status, and MAC address.	Get	3
47h	Device level ring (DLR) object	Contains status and configuration attributes of the DLR protocol.	Get	1

Class code	Class	Description	Access	Instances
48h	Quality of service (QoS) object	Contains mechanisms for processing data flows with different priorities.	Get	1

3.5.2 Identity object

The device information or parameters are retrieved via the instances.

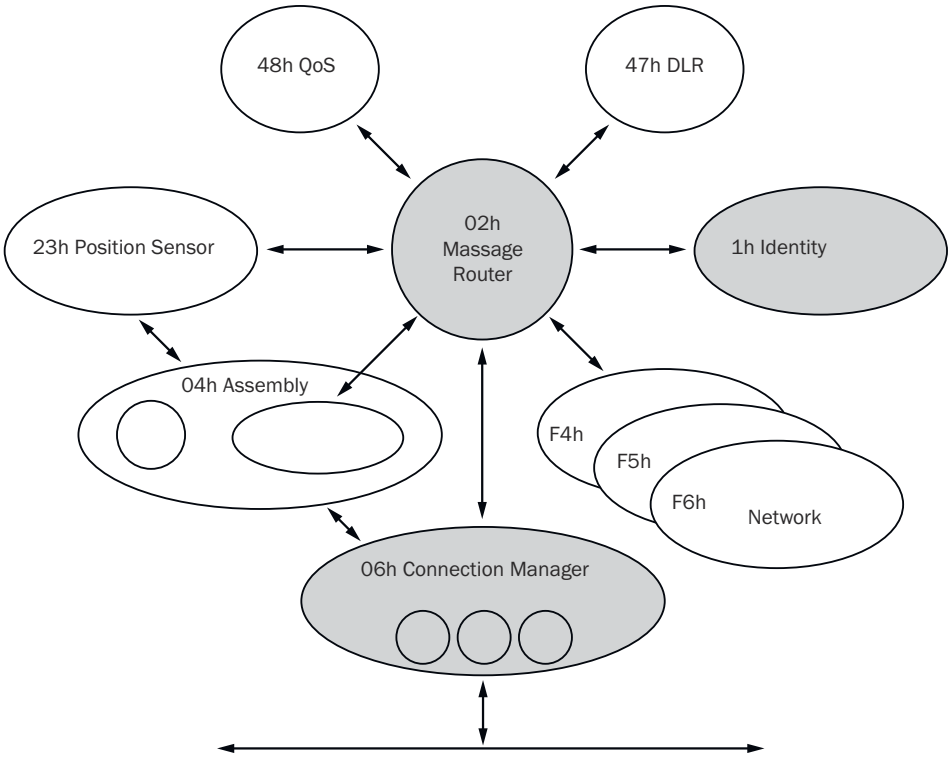


Figure 10: Connections for the identity object

Table 9: Class services of the identity object

Service code	Service	Description
01h	Get_Attribute_All	Returns the values of all attributes.
0Eh	Get_Attribute_Single	Returns the values of an attribute.

Table 10: Class attributes of the identity object

Attribute ID	Access	Description	Data type	Default value
1	Get	Object revision index	Uint	0001h
2	Get	Highest instance number in this class	Uint	0001h
3	Get	Number of object instances in this class	Uint	0001h
4	Get	Optional attributes list	STRUCT	-
6	Get	Highest class attribute ID that appears	Uint	0007h
7	Get	Highest instance attribute implemented	Uint	0075h

NOTE

The class attribute 5 has not been implemented.

Table 11: Instance services of the identity object

Service code	Service	Description
01h	Get_Attribute_All	Returns the values of all attributes.
0Eh	Get_Attribute_Single	Returns the values of an attribute.
05h	Reset	Resets the device: 0 = The device is reinitialized (power on). 1 = The device is reinitialized (power on) and reset to factory settings.

Table 12: Instance attributes of the identity object

Attribute ID	Access	Name	Description	Data type	Default value
01h	Get	Vendor ID	Manufacturer ID 0328h = SICK	Uint	0328h
02h	Get	Device type	Device profile 22h = Encoder	Uint	0022h
03h	Get	Product code	Manufacturer-specific product code 03h = Singleturn 04h = Multiturn	Uint	
04h	Get	Revision	Contains the firmware revision number in the format XX.XX	STRUCT	
	Get	Major revision	Front part of the revision number e.g. 01 (depending on the release)	Uint	01h
	Get	Minor revision	Rear part of the revision number e.g. 02 (depending on the release)	Uint	02h
05h	Get	Status	Device status flags	WORD	see table 13
06h	Get	Serial number	Serial number in the format YY.WW.xxxx Y = Year W = Week x = Sequential number E.g. 0E.34.0001 (depending on the release)	UDInt	0E340001h
07h	Get	Product name	Product name	Short_ String	AFx60A-Eth/IP
68h	Get	Vendor	Version of the firmware in the FPGA (e.g. 1.2.0)	UDInt	00010200h

Table 13: Bits of the “Status” instance attribute

Bit	Name	Description	Default value
0	Owned	0 = No connection with the master 1 = Connection established with the master	0
1	-	Reserved	0
2	Configured	0 = Device with standard configuration 1 = No standard configuration	0
3	-	Reserved	0
4 ... 7	Extended device Status field	Manufacturer-specific status bits	see table 14

Bit	Name	Description	Default value
8	Minor recoverable status	0 = No error 1 = Error that can be reset (device not in error status)	0
9	Minor unrecoverable status	0 = No error 1 = Error that can be reset (device not in error status)	0
10	Major recoverable status	0 = No major error 1 = Major error that can be reset (device in error status)	0
11	Major unrecoverable status	0 = No major error 1 = Major error that cannot be reset (device in error status)	0
12 ... 15	-	Reserved	0000

Table 14: Bits 4 to 7 of the "Status" instance attribute

Possible combinations Bits 4 ... 7	Description
0000	Device in self-test
0001	Firmware update in progress
0010	At least one connection error
0011	No I/O connection established
0100	Configuration in non-volatile memory (EEPROM) failed
0101	Major error, bit 10 or bit 11 = 1
0110	At least one connection in "Run" operating mode
0111	At least one connection present, all in "Idling" operating mode
1000 ... 1111	Reserved

3.5.3 Assembly object

The assembly object enables data attributes from different objects to be grouped together into one single object. The absolute encoder supports only static compilation of attributes, which is why the number of instances is fixed.

Table 15: Class services of the assembly object

Service code	Service	Description
01h	Get_Attribute_All	Returns the values of all attributes.
0Eh	Get_Attribute_Single	Returns the values of an attribute.

Table 16: Class attributes of the assembly object

Attribute ID	Access	Description	Data type	Default value
1	Get	Object revision index	Uint	0002h
2	Get	Highest instance number in this class	Uint	006Ah
3	Get	Number of object instances in this class	Uint	0007h
6	Get	Highest class attribute ID that appears	Uint	0007h
7	Get	Highest instance attribute implemented	Uint	0004h

NOTE
The class attributes 4 and 5 have not been implemented.

The encoder supports only **input** and **listen-only** connections.

Table 17: Instance services of the assembly object

Service code	Service	Description
01h	Get_Attribute_All	Returns the values of all attributes.
0Eh	Get_Attribute_Single	Returns the values of an attribute.

Table 18: Instance attributes of the assembly object

Instance	Attribute ID	Access	Description	Bits	Bytes
1	3	Get	Position value	32	4
2	3	Get	Position value Warning and alarm flags	32 8	5
3	3	Get	Position value Speed	32 32	8
4 ... 5	-	-	-	-	-
100	3	Set/Get	Configuration data	224	28
101	3	Get	Error Position value	32 32	8
102	3	Get	Error Position value Warning and alarm flags	32 32 8	9
103	3	Set/Get	Error Position value Speed	32 32 32	12
101WS	3	Get	Error Position value	32 32	8
102WS	3	Get	Error Position value Warning and alarm flags	32 32 8	9
103WS	3	Set/Get	Error Position value Speed	32 32 32	12
110	3	Set/Get	Dummy instance, for the configuration data of a listen-only connection	0	0

NOTE

- Instances 4 and 5 from encoder profile 22h are not implemented.
- Instances 100 to 110 are manufacturer-specific assemblies.
- If instances 101, 102 and 103 are used, then configuration assembly 100 is activated. If instances 101WS, 102WS, and 103WS are used, then configuration assembly 100 is **not** activated.

I/O assembly

The I/O data is retrieved/output via the instances.

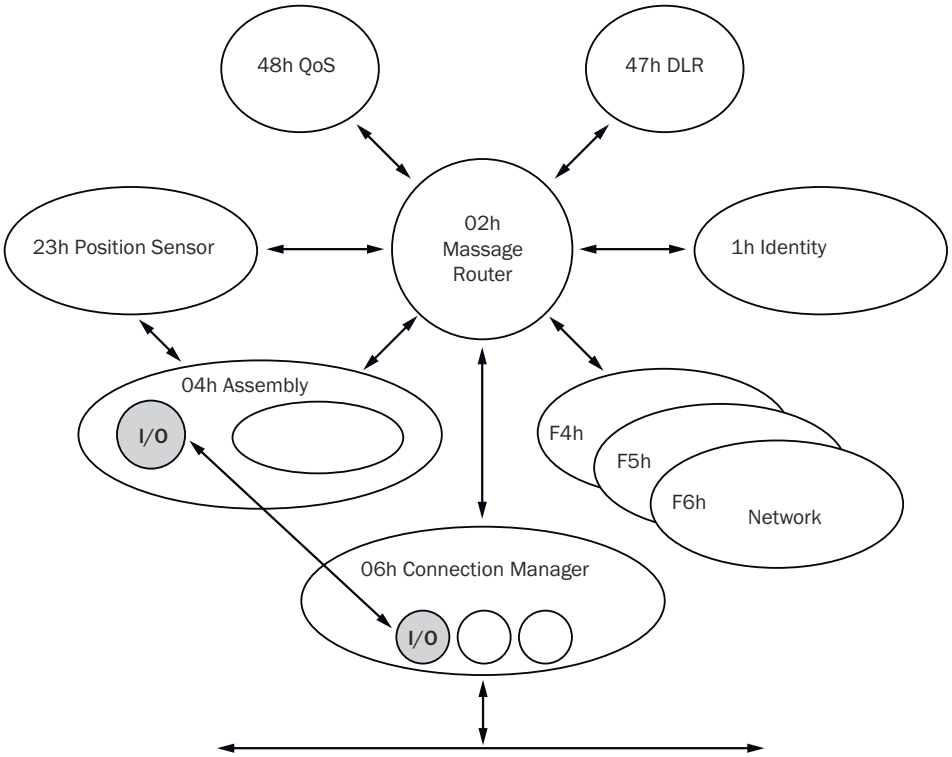


Figure 11: Connections for the I/O assembly

Table 19: Data format of the attributes of the I/O assembly

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	Position value (least significant byte)							
	1	Position value							
	2	Position value							
	3	Position value (most significant byte)							
2	0	Position value (least significant byte)							
	1	Position value							
	2	Position value							
	3	Position value (most significant byte)							
	4							Warning	Alarm
3	0	Position value (least significant byte)							
	1	Position value							
	2	Position value							
	3	Position value (most significant byte)							
	4	Speed value (least significant byte)							
	5	Speed value							
	6	Speed value							
	7	Speed value (most significant byte)							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
101/ 101WS	0	Fault header (least significant byte, see table 33, page 108)							
	1	Fault header							
	2	Fault header							
	3	Fault header (most significant byte)							
	4	Position value (least significant byte)							
	5	Position value							
	6	Position value							
	7	Position value (most significant byte)							
102/ 102WS	0	Fault header (least significant byte)							
	1	Fault header							
	2	Fault header							
	3	Fault header (most significant byte)							
	4	Position value (least significant byte)							
	5	Position value							
	6	Position value							
	7	Position value (most significant byte)							
	8							Warning	Alarm
103/ 103WS	0	Fault header (least significant byte, see table 33, page 108)							
	1	Fault header							
	2	Fault header							
	3	Fault header (most significant byte)							
	4	Position value (least significant byte)							
	5	Position value							
	6	Position value							
	7	Position value (most significant byte)							
	8	Speed value (least significant byte)							
	9	Speed value							
	10	Speed value							
	11	Speed value (most significant byte)							

Configuration assembly

The encoder can be configured via the configuration assembly.

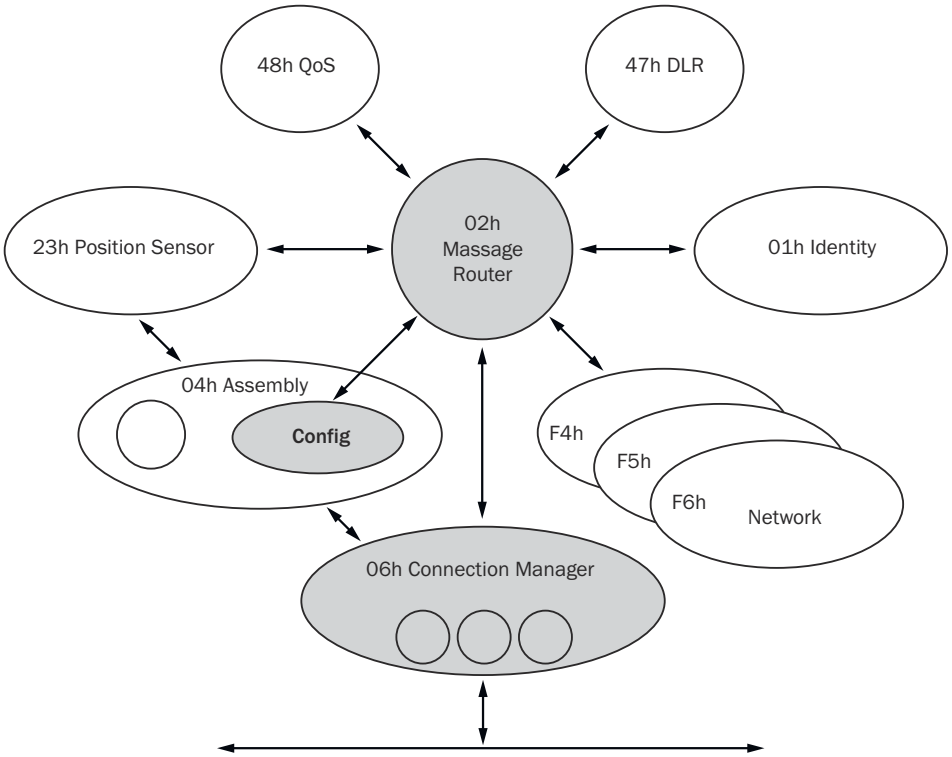


Figure 12: Connections for the configuration assembly

NOTE

- If the encoder is included as a generic module, then the configuration assembly can be activated or not **independently of** the I/O assembly instances.
- If the EDS file (electronic data sheet) of the encoder is used, then **depending on** the instances of the I/O assembly, the configuration assembly is activated or not:
  - Active with instances 101, 102 and 103.
  - Not active with instances 101WS, 102WS, and 103WS.
- When the configuration assembly is activated, it must not be empty. Otherwise, the control may output an error.

Table 20: Data format of the configuration assembly attributes

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
100	0	Not used							
	1	Not used							
	2	Not used							
	3	Not used							
	4	Steps per revolution CPR (least significant byte)							
	5	CPR							
	6	CPR							
	7	CPR (most significant byte)							
	8	Total resolution CMR (least significant byte)							
	9	CMR							
	10	CMR							
	11	CMR (most significant byte)							
	12	Not used							cw/ccw [1]

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	13	Not used							scf [2]
	14	Not used							raf [3]
	15	Not used							
	16	Numerator of the number of revolutions CNR_N (least significant byte)							
	17	CNR_N							
	18	CNR_N							
	19	CNR_N (most significant byte)							
	20	Denominator of the number of revolutions CNR_D (least significant byte)							
	21	CNR_D							
	22	CNR_D							
	23	CNR_D (most significant byte)							
	24	Speed measurement unit (least significant byte)							
	25	Speed measurement unit (most significant byte)							
	26	Not used							
	27	Not used							

1 cw = clockwise
ccw = counterclockwise
2 scf = scaling function
3 raf = round axis functionality

NOTE

- The structure of the configuration assembly is fixed.
- During initialization of the encoder, it reads the data from the controller.
- The **heartbeat connection point** for input connections of the PLC, i.e. for the output of the encoder, must be set to 198 (see figure 59, page 68).
- The **heartbeat connection point** for **listen-only connections** must be set to 199.

3.5.4 Position sensor object

The position sensor object contains all attributes of the encoder. Explicit messages can be used to retrieve or set all parameters.

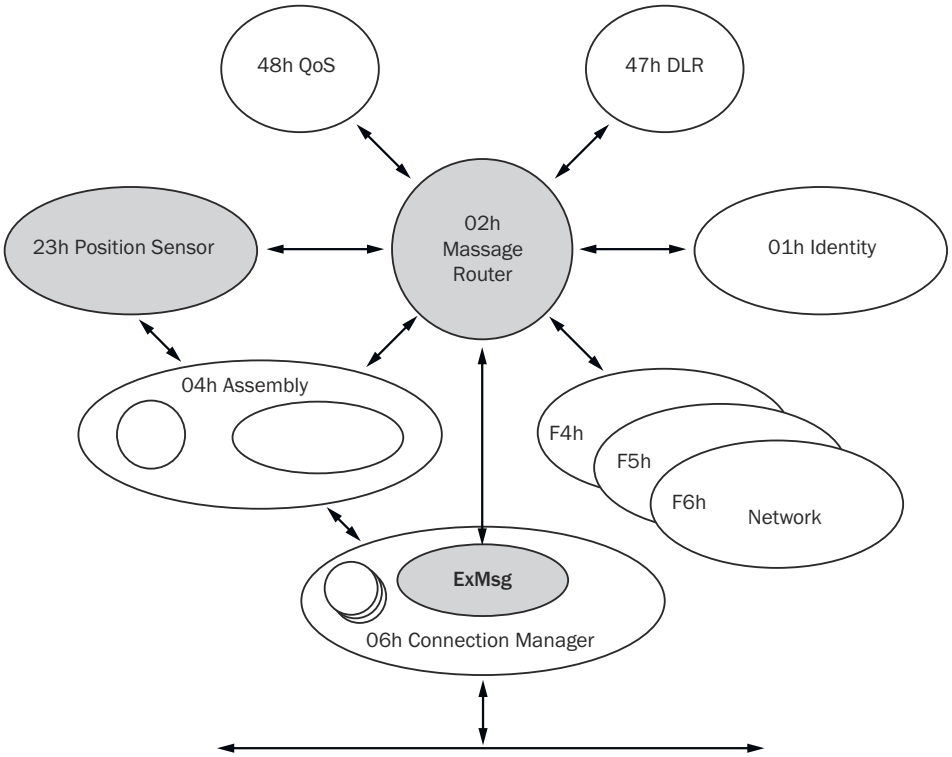


Figure 13: Connections for explicit messages to the position sensor object

Table 21: Class services of the position sensor object (to the class attributes)

Service code	Service	Description
05h	Reset	Resets the encoder to factory settings (see "Saving and resetting configuration", page 36).
0Eh	Get_Attribute_Single	Returns the values of an attribute.
15h	Restore	Restores all parameters last saved in non-volatile memory (see "Saving and resetting configuration", page 36).
16h	Save	Saves parameters to the non-volatile memory (see "Saving and resetting configuration", page 36).

Table 22: Class attributes of the position sensor object

Attribute ID	Access	Description	Data type	Default value
1	Get	Object revision index	Uint	0002h
2	Get	Highest instance number in this class	Uint	0001h
3	Get	Number of object instances in this class	Uint	0001h
4	Get	Optional attributes list	STRUCT	-
5	Get	Optional services list	STRUCT	-
6	Get	Highest class attribute ID that appears	Uint	0064h
7	Get	Highest instance attribute implemented	Uint	-
100	Get	The version of firmware	Array	AFx_aa.bb.dd.mm.yy

Table 23: Instance services of the position sensor object (for the instance attributes)

Service code	Service	Description
0Eh	Get_Attribute_Single	Returns the values of an attribute
10h	Set_Attribute_Single	Sets the value of an attribute

Table 24: Instance attributes of the position sensor object

Attribute ID	Access	V/NV [1]	Name	Description	Data type	Min. max. (default value)
01h	Get	V	Number of attributes	Number of attributes in this class	Uint	0000h FFFFh
02h	Get	V	Attribute list	List of supported attributes	Array of bytes	-
0Ah	Get	V	Position value signed	Current position value	DInt	-
0Bh	Get	NV	Position sensor type	01h = Singleturn 02h = Multiturn	Uint	0001h 0002h (0002h)
0Ch	Set	NV	Direction counting	Code sequence 0 = Clockwise 1 = Counterclockwise	BOOL	(0)
0Dh	Set	NV	Commissioning diagnostic control	Encoder self-test 0 = Off 1 = On	BOOL	(0)
0Eh	Set	NV	Scaling Function Control	Scaling 0 = Off 1 = On	BOOL	(0)
0Fh	Set	NV	Position format	Format of position measurement 1001h = steps	ENG UInt	(1001h)
10h	Set	NV	Counts per range	Number of steps per revolution (CPR)	UDInt	00000001h 00040000h (00040000h)
11h	Set	NV	Total measuring range	Total resolution (CMR)	UDInt	00000001h 40000000h (4,096 × attribute 10h)
12h	Set	NV	Position measuring increment	Smallest resolution (always 1)	UDInt	00000001h 00000001h
13h	Set	NV	Preset value	Preset value	DInt	00000000h Attribute 11h - 1 (00000000h)
15h	Get	NV	Position status register	Indicates if and how the limit set by attributes 16h and 17h is undercut/exceeded. Bit 0 = Out of range Bit 1 = Over range Bit 2 = Under range Bit 3 ... 7 = Reserved	Byte	(00h)
16h	Set	NV	Position low limit	Lower limit of the position [2]	DInt	00000000h 3FFFFFFFh (00000000h)
17h	Set	NV	Position high limit	Upper limit of the position [2]	DInt	00000000h 3FFFFFFFh (3FFFFFFFh)
18h	Get	V	Velocity value	Current speed The format is determined by attributes 19h and 1Ah.	DInt	00000000h XXXXXXXXh [3]

Attribute ID	Access	V/NV [1]	Name	Description	Data type	Min. max. (default value)
19h	Set	NV	Velocity format	Speed unit 1F04h = counts/s 1F05h = counts/ms 1F0Eh = turns/s 1F0Fh = turns/min 1F10h = turns/h	ENG UInt	(1F0Fh)
1Ah	Set	NV	Velocity resolution	Minimum resolution of the speed measurement	DUInt	(00000001h)
1Bh	Set	NV	Minimum velocity setpoint	Lower/upper limit of the speed in turns/min. [4] If the speed undercuts/exceeds this value, the warning flag (attribute 2Fh) is set.	DInt	(-12,000)
1Ch	Set	NV	Maximum velocity setpoint		DInt	(+12,000)
1Dh	Get	V	Acceleration value	Current acceleration. The format is determined by attributes 1Eh and 1Fh.	DInt	00000000h FFFFFFFFh
1Eh	Set	NV	Acceleration format	Acceleration unit 0810h = counts/ms^2 0811h = counts/s^2 0812h = turns/s^2 0813h = rad/s^2	ENG UInt	(0810h)
1Fh	Set	NV	Acceleration resolution	Minimum resolution of the acceleration measurement	DUInt	(1)
20h	Set	NV	Minimum acceleration setpoint	Lower/upper limit of acceleration in counts/ms^2. [5] If the acceleration undercuts/exceeds this value, the warning flag (attribute 2Fh) is set.	DInt	(C0000001h)
21h	Set	NV	Maximum acceleration setpoint		DInt	(3FFFFFFFh)
29h	Get	V	Operating status	Operational status of the encoder Bit 0: Direction 0 = Counting up 1 = Counting down Bit 1: Scaling 0 = Off 1 = On Bit 2 ... 4: Reserved Bit 5: Diagnostics on/off 0 = Off 1 = On Bit 6, 7: Reserved	Byte	
2Ah	Get	NV	Physical resolution span	Physical resolution per revolution = 18 bits	UDInt	(40000h)
2Bh	Get	NV	Physical resolution Number of span	Physical number of revolutions 0001h = Singleturn 1000h = Multiturn	Uint	(0001h) or (1000h)
2Ch	Get	V	Alarms	Bit field with flags for alarms and errors (see table 34, page 109)	WORD	-
2Dh	Get	NV	Supported alarms	Supported alarms and errors	WORD	3003h
2Eh	Get	V	Alarm flag	0 = No alarm/error 1 = Alarm/error	BOOL	-

Attribute ID	Access	V/NV [1]	Name	Description	Data type	Min. max. (default value)
2Fh	Get	V	Warnings	Bit field with flags for warnings (see table 35, page 110)	WORD	-
30h	Get	NV	Supported warnings	Supported warnings	WORD	67C3h
31h	Get	V	Warning flags	0 = No warning 1 = Warning	BOOL	-
32h	Get	NV	Operating time	Stored operating time in 0.1 h = 6 min	UDInt	0
33h	Get	NV	Offset value	Offset value is calculated when initializing the preset function.	DInt	00000000h
64h	Get	V	Temperature value	Current temperature with ± 5° accuracy -40 to +100 °C or -40 to +212 °F	INT	F060h 2710h
65h	Set	NV	Temperature value format	Temperature unit 1200h = °C (Celsius) 1201h = °F (Fahrenheit)	ENG UINT	(1200h)
66h	Set	NV	Temperature resolution	Smallest resolution of temperature (°C/100 or °F/100)	UDINT	(00000064h)
67h	Set	NV	Minimum temperature setpoint	Lower/upper limit of temperature in °C. [6] If the temperature undercuts/exceeds this value, the warning flag (attribute 2Fh) is set.	INT	F060h - (F060h = -4,000)
68h	Set	NV	Maximum temperature setpoint		INT	- 2710h (2710h = +10,000) or (52D0h = +21,200)
69h	Get	V	Fault header	see table 33, page 108	DWORD	(00000000h)
6Ah	Set	V	Special encoder functionalities	Bit field with flags for special encoder functions Bit 0: Slave sign of life (On/Off) Bit 1 ... 7: Not used Bit 8 ... 15: Update factor (2 ... 127) Bit 16: Storage mode (Auto (0)/Manual (1)) Bit 17 ... 23: Not used Bit 24 ... 31: Activation/deactivation of FTP firmware update functionality (default: active (1))	DWORD	(01000500h)
6Bh	Get	NV	Encoder motion time	Stored movement time in seconds (will be increased when moving).	UDINT	-
6Ch	Get	NV	Encoder operating time	Stored operating time in seconds (is increased as soon as the encoder is in operation).	UDINT	-
6Dh	Get	NV	Max. velocity	Highest speed that the encoder has reached since commissioning. [7]	UDINT	-
6Eh	Get	NV	Max. acceleration	Highest acceleration the encoder has experienced since commissioning. [8]	UDINT	-

Attribute ID	Access	V/NV [1]	Name	Description	Data type	Min. max. (default value)
6Fh	Get	NV	Max. Temp	Operating temperature in °C/100	UDINT	-4,000
70h	Get	NV	Min. temp	Lowest operating temperature reached in °C/100	UDINT	10,000
71h	Get	NV	Number of start-ups	Number of start-ups (power on) of the encoder	UDINT	-
72h	Get	V	LED current value	Momentary internal LED current of the sensor in µA	UINT	200 25,000 (0)
73h	Get	NV	Max. current value	Maximum internal LED current of the sensor system in µA	UINT	200
74h	Get	NV	Min. current value	Minimum internal LED current of the sensor system in µA	UINT	25,000
75h	Get	V	Direction change counter	Number of changes in the direction of rotation (the numerator increases when the encoder changes the direction of rotation).	UDINT	0
76h	Get	V	Revolution counter forward	Number of clockwise starts (The numerator increases when the encoder moves clockwise).	UDINT	0
77h	Get	V	Revolution counter backwards	Number of counterclockwise starts (The numerator increases when the encoder moves counterclockwise).	UDINT	0
78h	Get	V	Power supply voltage	Current supply voltage in mV	UINT	9,500 30,500 (24,000)
79h	Get	V	Max. power supply voltage	Maximum supply voltage in V (stored in EEPROM).	UINT	0 33 (0)
7Ah	Get	V	Preset offset value	Offset value calculated from the preset value. [9]	DINT	(00000000)
7Dh	Set	NV	Endless shaft functionality	Activates round axis functionality 0 = Off 1 = On	BOOL	(0)
7Eh	Set	NV	Number of revolutions, nominator	Numerator for the number of revolutions	UDINT	1 2,048 (2,048)
7Fh	Set	NV	Number of revolutions, divisor	Denominator for the number of revolutions	UDINT	1 65,535 (1)
80h	Set	NV	Velocity filter integration time	Number of measured values from which a mean value is formed (scanning frequency of the position as the basis for calculating the speed = 1 kHz). [10]	UDINT	0 128 (1)

Attribute ID	Access	V/NV [1]	Name	Description	Data type	Min. max. (default value)
81h	Set	NV	Velocity filter bandwidth	Bandwidth of the low pass filter in Hz [10] 0 = Deactivated	UDINT	0 1000 (100)
82h	Set	NV	Acceleration filter integration time	Number of measured values from which a mean value is formed (scanning frequency of the position as the basis for calculating the acceleration = 1 kHz).	UDINT	0 128 (1)
83h	Set	NV	Acceleration filter bandwidth	Bandwidth of the low pass filter in Hz 0 = Deactivated	UDINT	0 1000 (100)
84h	Set	NV	Velocity acceleration hysteresis	Hysteresis for the speed limits (attributes 1Bh and 1Ch). The unit depends on attribute ID 19h.	UDINT	0 3FFFFFFF (0)
85h	Set	NV	Velocity acceleration scaled	Activates the use of the scaled values for speed and acceleration. 1 = Scaling on 0 = Scaling off (speed and acceleration are calculated based on the physical/unscaled values). After changing the value, a restart is required to activate the change.	UDINT	0 1 (1)
86h	Set	V	Motion time limit	Limit of movement time in seconds	UDINT	00000000h FFFFFFFFh (630,720,000)
87h	Set	V	Power time limit	Operating time limit in seconds	UDINT	00000000h FFFFFFFFh (630,720,000)
88h	Set	V	Direction changes limit	Limit of the number of changes in the direction of rotation	UDINT	00000000h FFFFFFFFh (1,000,000)
89h	Set	V	Starts in cw limit	Limit of the number of clockwise starts	UDINT	00000000h FFFFFFFFh (1,000,000)
8Ah	Set	V	Starts in ccw limit	Limit of the number of counterclockwise starts	UDINT	00000000h FFFFFFFFh (1,000,000)
8Bh	Set	V	Reset fault header bit 15	Resets bit 15 in the Fault header (see table 33, page 108).	Byte	(00h)

1 V = Volatile (volatile data), NV = Non-volatile (non-volatile data).
2 Area monitoring can be implemented with the lower and upper limit of the position. It is not an electronic cam.
3 The maximum speed depends on the “solid shaft” or “blind hollow shaft” mechanical interface used (see data sheet).
4 The unit changes with the velocity format (attribute ID 19h). The limits must then be converted accordingly, e.g. 12,000 turns/min = 200 turns/s.
5 The unit changes with the acceleration format (attribute ID 1Eh). The limits must then be converted accordingly, e.g. 2 counts/ms^2 = 2,000,000 counts/s^2.
6 The unit changes with the temperature value format (attribute ID 65h). The limits must then be converted accordingly.
7 The value is output in the format defined in attribute ID 19h.
8 The value is output in the format defined in attribute ID 1Eh.

[9] With normal scaling = Physical position; with round axis functionality = Physical position + range offset.
[10] The calculation of the speed can be adapted to the dynamic conditions of different applications via the settings of the parameter values for the filters. For example, for applications with low speed and dynamics, the following parameter values can be a good reference: 80h = 8 / 81h = 50.

Filter for the speed (attributes 80h and 81h) or the acceleration (attributes 82h and 83h)

The filters are used to smooth the raw speed or acceleration values.

NOTE
The filters are each applied in the following order:

- Integration time filter for speed (80h) or acceleration (82h)
- Low pass filter for speed (81h) resp. acceleration (83h)

The filter with attribute 80h forms a mean value from the speed measured values. The filter with attribute 82h forms a mean value from the acceleration measured values:

- With a configured value of 1, the mean value is formed from 2 measured values.
- With a configured value of 128, the mean value is formed from 129 measured values.

The filter with attribute 81h forms a low pass for the speed measured values. The filter with attribute 83h forms a low pass for the acceleration measured values:

- This is configured to 100 Hz at the factory. I.e. only speed or acceleration values ≤ 100 Hz are considered.

3.6 Integration and configuration options

The encoder can be integrated in EtherNet/IP in various ways and can be configured independently of the integration.

3.6.1 Integration in EtherNet/IP

The encoder can be integrated into EtherNet/IP:

- As a generic module (see "Integration of the encoder as generic module", page 67):
  All module settings must be entered manually.
- Using an EDS file (see "Integration and configuration using an EDS file", page 52):
  The module settings of the absolute encoder are already predefined.

3.6.2 Configuration

The following options are available for configuring the encoder:

- The configuration assembly
- The controller tags in the controller organizer
- The web server integrated in the encoder

Case 1: When integrated as generic module

If the encoder has been integrated as a generic module, then it can be configured depending on the **connection parameters** entered.

- If the configuration assembly is **activated** under Connection Parameters, then the configuration assembly is to be used for configuration (see "Module settings", page 68).
  In addition, the parameters that are not included in the configuration assembly can be configured with the web server integrated in the encoder.
- If the configuration assembly is **not activated** under Connection Parameters, then the web server can be used to configure all parameters (see "Configuration using the integrated web server", page 94).

NOTE
If the configuration assembly is active, the parameters entered there overwrite those which have been configured via the web server.

Case 2: When integrating using the EDS file

If the encoder was integrated using the EDS file, then it can be configured depending on the selected instances of the I/O assemblies (see table 18, page 23).

- If instances 101, 102 or 103 are used, then the configuration parameters can be configured in the **controller tags** . In addition, the parameters that are not included in the configuration assembly can be configured with the web server.
- If instances 101WS, 102WS or 103WS are used, then the web server can be used to configure the parameters.

Case 3: When using ladder routine for configuration mapping

A ladder routine for mapping the configuration data is available for the absolute encoder (see "Installing the ladder routine", page 56).

If the ladder routine is used for mapping and if instances 101WS, 102WS or 103WS are used (see table 18, page 23), then the encoder can be configured both from the controller (in the controller tags) and using the web server.

NOTE
In cases 1 and 2, the parameters are configured offline and written to the encoder and activated when switching to online mode.

If the ladder routine is used (case 3), then changes to the configuration also take effect immediately in online mode!

Parameter changes via the web server are immediately applied and displayed on the controller side. Parameter changes via the controller are applied immediately. However, to display it in the web browser, the corresponding page must be refreshed.

DANGER
Before changing the configuration, check whether there is any danger from the machine or system in which the encoder is integrated!

The ladder routine offers the option of changing parameter data of the encoder during operation, i.e. **while the control is in online mode**.

Changing the configuration therefore has a direct effect on the data output of the encoder. This could cause an unexpected reaction that could endanger people or damage the system or other objects.

NOTE
The configuration should only be changed when the encoder is at a standstill.

3.7 Parameterizable functions

3.7.1 Saving and resetting configuration

The configuration memory of the absolute encoder is divided into three parts. The following table shows the functions of the memory types.

Table 25: Configuration memory – functions of the different memory types

Memory type	Function
Volatile memory	The encoder works during operation with the values in the volatile memory. Changed parameters are first written to the volatile memory. These are lost when the power is turned off.
Non-volatile memory	When switching on, the encoder loads the values from the non-volatile memory into the volatile memory.
Factory settings	Contains the factory preset values.

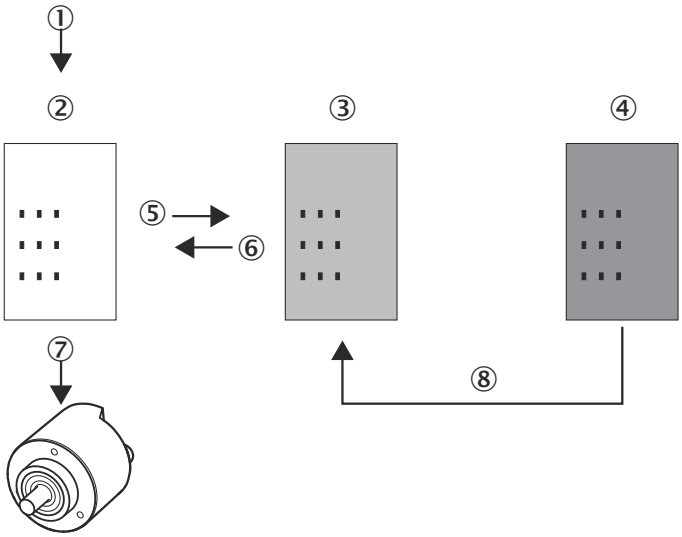


Figure 14: Configuration memory

① Parameterization
② Volatile memory
③ Non-volatile memory
④ Saved factory settings
⑤ Save
⑥ Restore
⑦ Acts on encoder
⑧ Reset + data 01h

Reset: Resetting to factory settings

- ▶ Set the address switches to 888 (see figure 19, page 44).
- ▶ Press the preset pushbutton for longer than 5 seconds.

Or:

- ▶ In the position sensor object class (23h), in the class service with the service code for **Reset** (05h), set the value for Data to 01h (see table 21, page 28).
  The parameters of the position sensor object are reset to the factory settings. table 26 shows which parameters are reset to which value.

DANGER

Before using the Reset function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Reset function leads to a reset of the parameters of the position sensor object to the factory settings, which can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE

The Reset function should only be used when the encoder is at a standstill.

Restore: Resetting to the values in the non-volatile memory

Each time the encoder is switched on, the values of the position sensor object are read from the non-volatile memory.

- If the parameter is to be read from the non-volatile memory during operation, then the **Restore** class service (service code 15h) of the position sensor object must be used (see table 21, page 28).

Save: Saving parameters to non-volatile memory

- Use the **Save** class service (service code 16h) from the position sensor object (see table 21, page 28).
  The parameters are saved to the non-volatile memory. The following table shows which parameters are saved.

Table 26: Parameters that are saved or reset

Attribute ID in the position sensor object	Parameter	Factory setting
0Ch	Code sequence	cw
0Eh	Scaling	Off
10h	Increments per revolution	262,144
11h	Total resolution	1,073,741,824
13h	Preset value	0
16h	Lower position limit	0
17h	Upper position limit	1,073,741,823
19h	Speed unit	turns/min
1Bh	Lower speed limit	-12,000
1Ch	Upper speed limit	12,000
1Eh	Acceleration unit	counts/ms^2
20h	Lower limit of acceleration	-1,073,741,823
21h	Upper limit of acceleration	1,073,741,823
65h	Temperature unit	°C
7Dh	Round axis functionality	Off
7Eh	Numerator for the number of revolutions	2,048
7Fh	Denominator for the number of revolutions	1
80h	Number of measured values from which a mean value is formed	1
81h	Bandwidth of the low pass filter	100

Attribute ID in the position sensor object	Parameter	Factory setting
82h	Number of measured values from which a mean value is formed	1
83h	Bandwidth of the low pass filter	100
84h	Hysteresis for the speed limit values	0
85h	Hysteresis for the acceleration limit values	0
86h	Limit of movement time in seconds	630,720,000
87h	Operating time limit in seconds	630,720,000
88h	Limit of the number of changes in the direction of rotation	1,000,000
89h	Limit of the number of clockwise starts	1,000,000
8Ah	Limit of the number of counterclockwise starts	1,000,000

NOTE

The following parameters are not reset:

- Movement time
- Uptime
- Lower limit of the temperature
- Upper limit of the temperature
- Maximum voltage supply

3.7.2 IP address

The IP address is required for identification of the encoder in the EtherNet/IP. This is obtained for the encoder from a DHCP server (see "Assigning the IP address via DHCP", page 46) or permanently entered via address switches (see "IP address setting", page 44).

- If the IP address is obtained via DHCP, then any address range is possible.
- If the IP address is set via address switches, then the address range is set to 192.168.1.xxx.

3.7.3 Slave sign of life

The absolute encoder supports the slave sign-of-life functionality.

It is transmitted in bit 30 of the Fault header. It is used so that the controller can determine whether the encoder is in operation even if the position data does not change (e.g. at a standstill).

The bit changes its value in the configured update cycle.

The update cycle is formed from the requested packed interval (RPI) and an update factor. The RPI can be between 5 and 750 ms:

Update cycle = RPI × update factor × 6

The update factor is determined with attribute 6Ah of the position sensor object (see table 24, page 29).

The supported value depends on the RPI time of the encoder connection. The update cycle should be at least twice as long as the RPI (so 1500 ms for RPI = 750 ms).

3.7.4 Code sequence

The code sequence determines at which direction of rotation, starting from a viewing direction on the shaft, the position value increases.

- Clockwise = Increasing position value when shaft is rotated clockwise
- Counterclockwise = Increasing position value when shaft is rotated counterclockwise

DANGER
Before using the Code sequence function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Code sequence function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Code sequence function should only be used when the encoder is at a standstill.

3.7.5 Scaling

Scaling allows the steps per revolution or the total resolution to be scaled.

NOTE
Only if the **Scaling** parameter (attribute ID 0Eh of the position sensor object) is configured to **Enable**, are the entered values for the steps per revolution or the total resolution applied.

DANGER
Before using the Scaling function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Scaling function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Scaling function (steps per revolution or total resolution) should only be used when the encoder is at a standstill.

3.7.6 Increments per revolution

The resolution of the absolute encoder is max. 262,144 steps per revolution. The resolution is scalable from 1 ... 262,144 in whole numbers.

NOTE
The parameter is not used if the round axis functionality (see "Round axis functionality", page 40) is activated.

3.7.7 Total resolution/measuring range

The total resolution, i.e. the measuring range of the AFM60 EtherNet/IP, is max. 1,073,741,824 steps. The total resolution must be 2^n-fold the steps per revolution.

Table 27: Examples for total resolution

Increments per revolution	n	Total resolution
1,000	3	8,000

Increments per revolution	n	Total resolution
8,179	5	261,728
2,048	11	4,194,304

NOTE
This restriction is not relevant if the round axis functionality (see "Round axis functionality", page 40) is activated.

3.7.8 Preset function

The Preset function is used to set the encoder to a predefined start position. With the help of a preset value, the encoder can be set to any position within the measuring range.

The preset value can be set in the following way:

- With the help of the preset pushbutton
- With an acyclic explicit message
  Here the preset value is passed as an attribute (13h) of the position sensor object.
- With the help of the integrated web server and the ladder routine

DANGER
Before using the Preset function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Preset function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Preset function should only be used when the encoder is at a standstill.

3.7.9 Speed measurement unit

The parameter defines the unit with which the speed is transmitted.

Possible units are:

- counts/s [1)]
- counts/ms [1)]
- turns/s
- turns/min
- turns/h

The factory setting is **turns/min**.

[1)] Depending on the configured resolution.
Example: Resolution = 2,000 steps; the encoder rotates 0.5 times per second = 1,000 counts/s or 1 counts/ms.

3.7.10 Round axis functionality

The round axis functionality removes the restriction that the total resolution must be 2^n-fold the steps per revolution. The shaft is considered an **endless shaft**.

The steps per revolution are not configured directly, but the numerator and denominator for the number of revolutions are determined.

The following requirements must be met or attributes must be set:

- Attribute ID 0Eh, scaling to 1
- Attribute ID 7Dh, round axis functionality set to 1

Number of revolutions, numerator

The numerator (attribute ID 7Eh, nominator “CNR_N”) is scalable from 1 ... 2,048 in whole numbers. Factory setting for the numerator is 2,048.

Number of revolutions, denominator

The denominator (attribute ID 7Fh, divisor “CNR_D”) is scalable from 1 ... 65,535 in whole numbers. Factory setting for the denominator is 1.

Resulting total resolution

The resulting total resolution (attribute ID 11h, total resolution “CMR”) is scalable from 1 ... 536,870,912 in whole numbers.

The following restrictions must be observed:

- CMR ≤ ÷ (CNR_N ÷ CNR_D) x physical resolution (PhysRes)
- CNR_N ÷ CNR_D ≤ 1/2
- The resulting total resolution CMR of the round axis functionality can thus be at most half the physical resolution (PhysRes) of the encoder (0.5 x 1,073,741,824 = 536,870,912).

NOTE
The round axis functionality is only supported by the multiturn encoder.

NOTE
When the round axis functionality is activated, the possible value range for monitoring the position limits is automatically limited to the total measuring range (CMR) of the round axis functionality.

NOTE
If a preset value was already set for the current position before the round axis functionality was activated, this position value will change when the round axis functionality is activated.

DANGER
Before using the Round axis functionality function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Round axis functionality function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Round axis functionality function should only be used when the encoder is at a standstill.

3.8 Operating elements and status indicators

The absolute encoder has five LEDs.

Three of the LEDs signal the operational status (Net, Mod and Encoder), two signal the status of the Ethernet interface (Link 1 and Link 2).

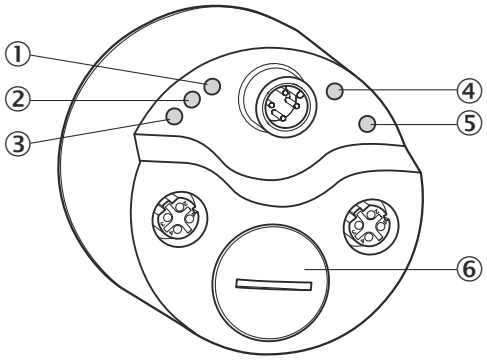


Figure 15: Position of the LEDs, the address switches and the preset pushbutton

① Net
② Mod
③ Link 1
④ Encoder
⑤ Link 2
⑥ Screw cover

The LEDs are multicolored. see "Error and status indications of the LEDs", page 106 for the meaning of the signals.

The following control elements are located under the screw cover:

- Address switch
- Preset pushbutton

4 Commissioning

4.1 Electrical installation

DANGER
Risk of injury from electrical voltage.
Disconnect the system from the voltage supply to prevent the system from starting unintentionally.

- Before starting work on the system, ensure that it is and remains in a de-energized state during electrical installation.

Connecting male and female connectors are required for electrical installation (see data sheet of the absolute encoder).

4.1.1 Encoder connections

The encoder connections are located on the rear side.

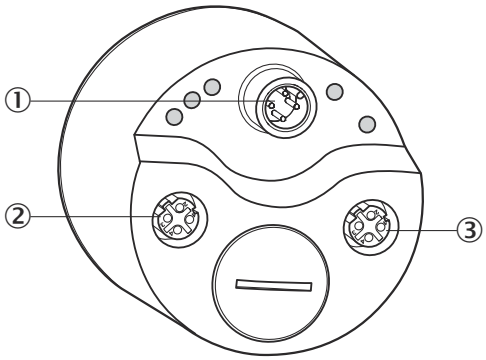


Figure 16: Position of the AFS60/AFM60 EtherNet/IP connections

① Voltage supply
② Ethernet port 1
③ Ethernet port 2

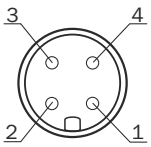


Figure 17: Ethernet port M12 × 4, female connector

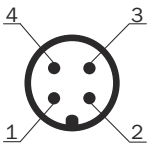


Figure 18: Voltage supply M12 × 4, male connector

NOTE
Two Ethernet ports are used if the absolute encoder is integrated in a DLR or a line topology (see figure 5, page 16).

Table 28: Pin assignment of the voltage supply connection

PIN	Signal	Wire color [1]	Function
1	V_S	Brown	Supply voltage 10 ... 30 V DC
2	-	White	Do not use
3	GND	Blue	0 V DC (ground)

PIN	Signal	Wire color [1]	Function
4	-	Black	Do not use

1 When using pre-assembled cables.

NOTE

Pin 2 and 4 must **not be connected**; this can lead to destruction of the absolute encoder.

Table 29: Pin assignment of the Ethernet port 1 and 2 connections

PIN	Signal	Wire color [1]	Function
1	TxD+	White/orange	Ethernet
2	RxD+	White/gray	Ethernet
3	TxD-	Orange	Ethernet
4	RxD-	Green	Ethernet

1 When using pre-assembled cables.

NOTE

- ▶ **Connect the screen to the encoder housing.**
- ▶ Observe the maximum lengths of cable.
- ▶ Mount all cables with strain relief.

4.2 Settings on the hardware

The following elements for adjustment are located under the screw cover:

- Three address switches
- Preset pushbutton

▶ Open the screw cover using a screwdriver for slotted screws with a blade width of min. 10.0 mm.

4.2.1 IP address setting

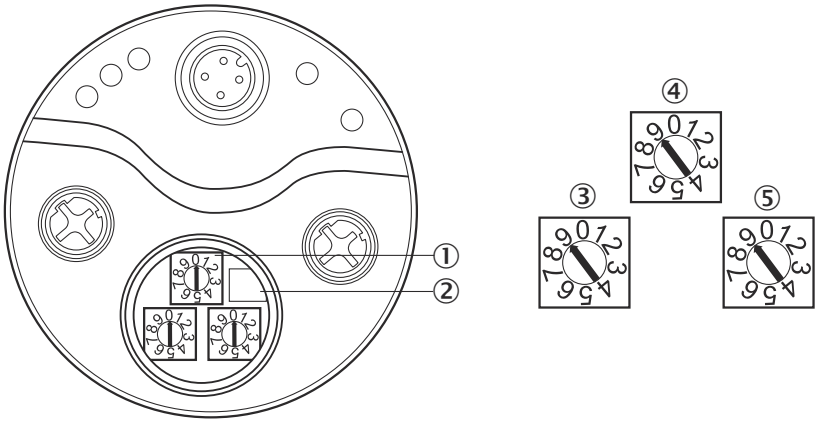


Figure 19: Address switch and preset pushbutton

① Address switch
② Preset pushbutton
③ Hundreds place
④ Tens place
⑤ Ones place

Table 30: Address switch – meaning of the adjustable values

Value	Meaning
888	The encoder obtains its IP address from a DHCP server.
001 ... 254	Fixed IP address. Only the least significant byte (1 ... 254) can be changed. • Address range 192.168.1.xxx is fixed • Subnet mask 255.255.255.0 is fixed • Gateway address 0.0.0.0 is fixed
000/999	The encoder loads the IP address from non-volatile memory when switched on.

Fixed IP address via address switch

- ▶ Use the left address switch to set the hundreds place of the address.
- ▶ Use the middle address switch to set the tens place of the address.
- ▶ Use the right address switch to set the ones place of the address.

Obtaining the IP address via DHCP

1. Switch off the encoder.
2. Set the address switches to 888.
3. Switch the encoder back on.

The absolute encoder now obtains its IP address from a DHCP server and stores it in non-volatile memory. Detailed step-by-step instructions for assigning the IP address via DHCP can be found in the chapter 5.2.2.

If necessary, deactivate the DHCP function in the encoder (see "Freezing the assigned IP address", page 49).

The following procedure ensures that the encoder retains the IP address assigned via DHCP even after it is switched on again:

- ▶ Set the address switches to 000.
  From now on, the encoder loads the IP address from the non-volatile memory every time it is switched on.

4.2.2 Triggering a preset value with the preset pushbutton

To trigger the preset value, press the preset pushbutton.
The value from attribute 13h of the position sensor object is used as the new position value (see table 24, page 29).

DANGER
Before using the Preset function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Preset function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE

- The Preset function should only be used when the encoder is at a standstill.
- The preset value must be within the configured measuring range.
- Do **not** press the preset pushbutton for longer than 5 seconds, as this may reset the encoder to the factory settings.

5 Configuration using a PLC

The encoder can be integrated into an Allen-Bradley control system from Rockwell as well as into other systems whose controls have an EtherNet/IP communication interface.

NOTE

- All software notes are in English.
- All software notes refer to the RSLogix 5000 software. For the example project below, the Allen-Bradley **ControlLogix Controller 1756-L61** with **RSLogix 5000** control system was used. It is assumed that the hardware has already been installed.

5.1 Delivery state

The encoder is delivered with the following parameters:

- Code sequence = Clockwise
- Scaling = Not activated
- Steps per revolution = 262,144
- Total resolution of the AFS60 EtherNet/IP = 262,144
- Total resolution of the AFM60 EtherNet/IP = 1,073,741,824
- Preset = 0
- Unit of speed measurement = turns/min
- Round axis functionality = not activated
- Numerator for the number of revolutions (round axis functionality) = 2,048
- Denominator for the number of revolutions (round axis functionality) = 1
- Position of the address switches = 999 (meaning see table 30, page 45)

5.2 IP address of the encoder

5.2.1 Without DHCP server

If the IP address of the encoder has been permanently entered via the address switches (see "IP address setting", page 44), then this IP address must be used in the controller.

NOTE

This limits the address range to 192.168.1.xxx. Only if the IP address is obtained via DHCP is any address range possible.

5.2.2 Assigning the IP address via DHCP

1. Switch off encoder (supply voltage off).
2. Set value “888” via the 3 address switches.
3. Connect the encoder to a laptop/PC on which a DHCP server (e.g. BootP) is installed.
4. Switch on encoder (supply voltage on).
5. Check IP address range on the laptop/PC and set if necessary (same range in which the PLC/controller and the encoder should be located).
6. Open DHCP server (e.g. BootP).

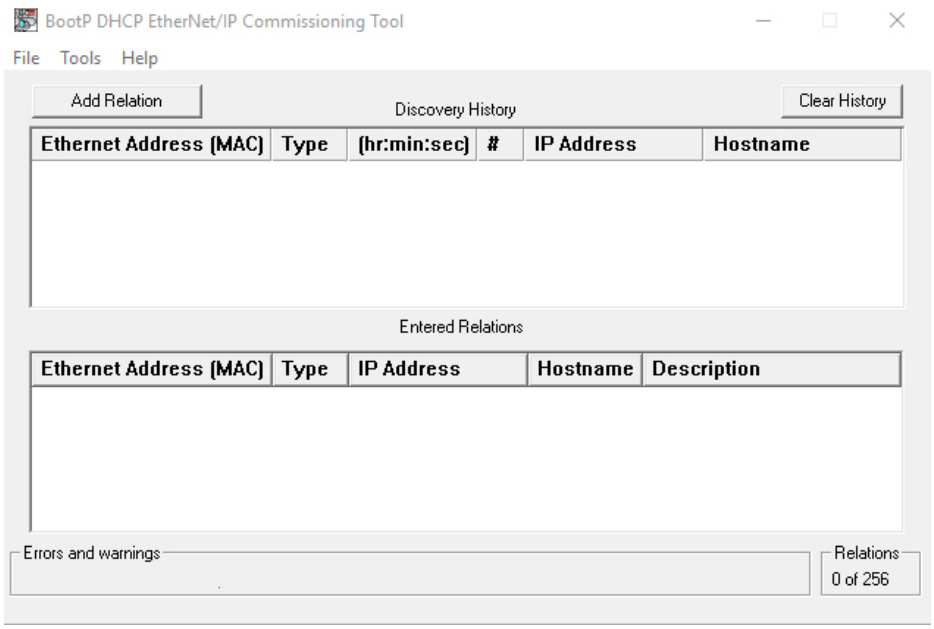


Figure 20: Empty window in BootP

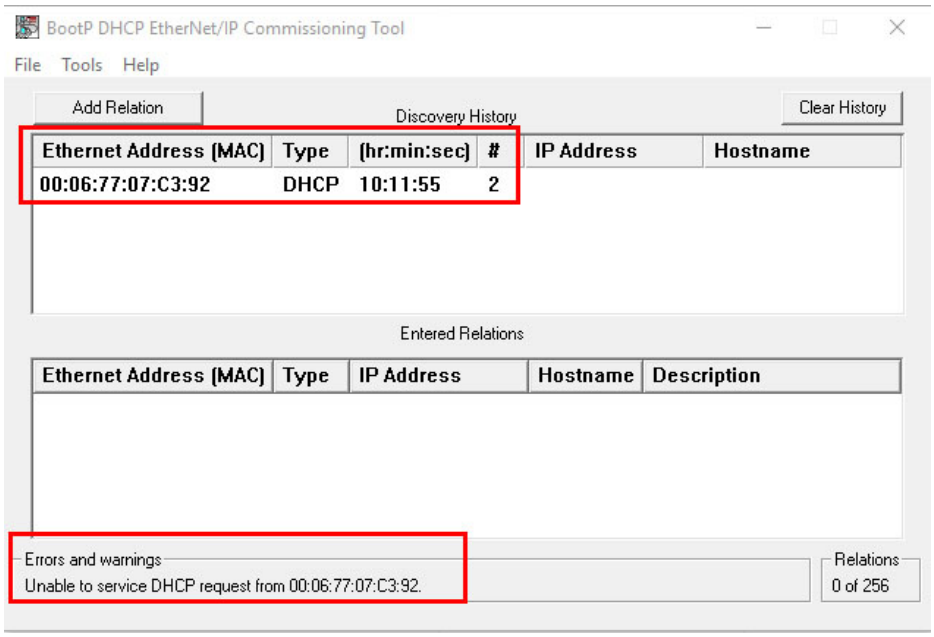


Figure 21: Contact structure of the encoder

✓ As soon as contact with the encoder is established via the Ethernet address (MAC), but there is no link to the specific IP address yet, the message “Unable to service DHCP request” is displayed in BootP.
This means that the encoder has started to periodically send DHCP requests to get an IP address assigned to its MAC address. However, the server (laptop/PC) cannot yet respond accordingly.

7. Add encoder to the list via the **Add Relation** button.
8. Assign the last three digits of the IP address (client IP Address) for the encoder. Use the same IP address range as specified by the server (laptop/PC) (server IP address).

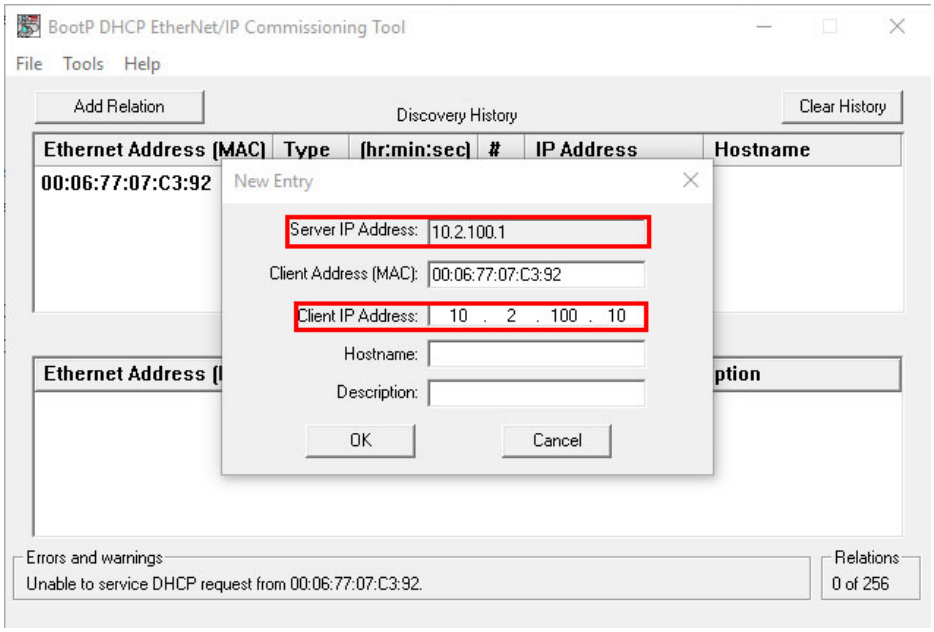


Figure 22: IP address assignment

✓ Once the encoder has been assigned an IP address in BootP, BootP starts to respond and sends the corresponding IP address to the encoder (via its MAC address).
This typically takes up to 30 seconds.

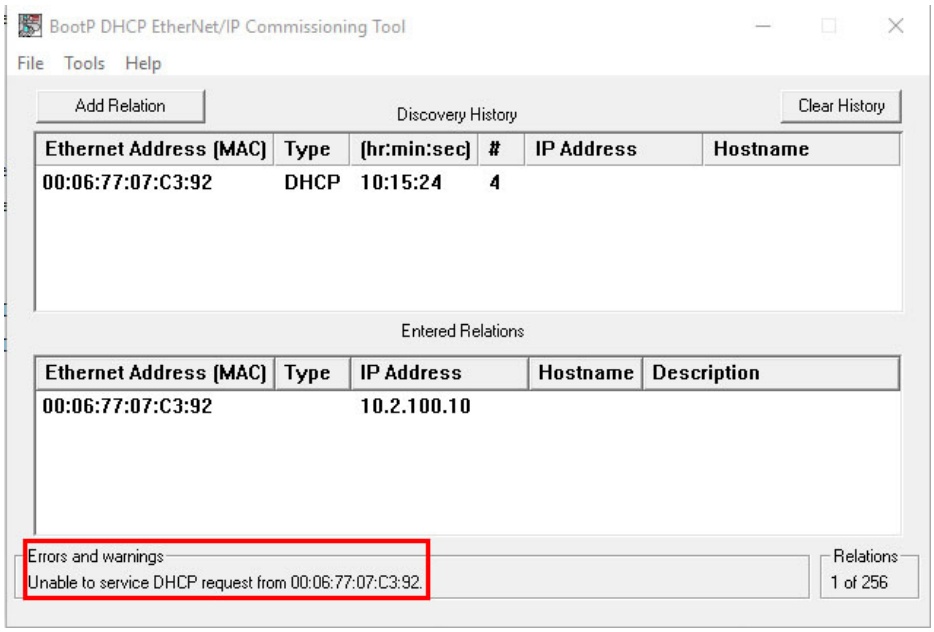


Figure 23: Sending the IP address to the encoder

✓ As soon as the encoder has correctly received the assigned IP address, the status message at **Errors and warnings** changes accordingly.

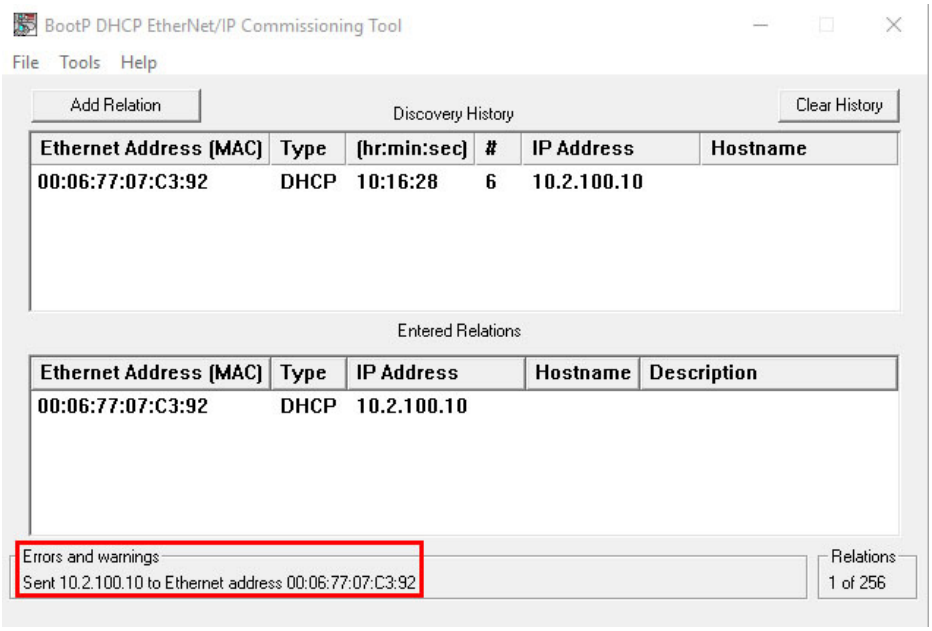


Figure 24: IP address assigned successfully

5.2.3 Freezing the assigned IP address

1. Switch off encoder (supply voltage off).
2. Set value “000” via the three address switches.
3. Switch on encoder (supply voltage on).

Deactivating DHCP server

4. Click on the encoder address line in the lower window area to activate the **Disable BOOTP/DHCP** button.
5. Click the **Disable BOOTP/DHCP** button to deactivate the encoder’s DHCP server.

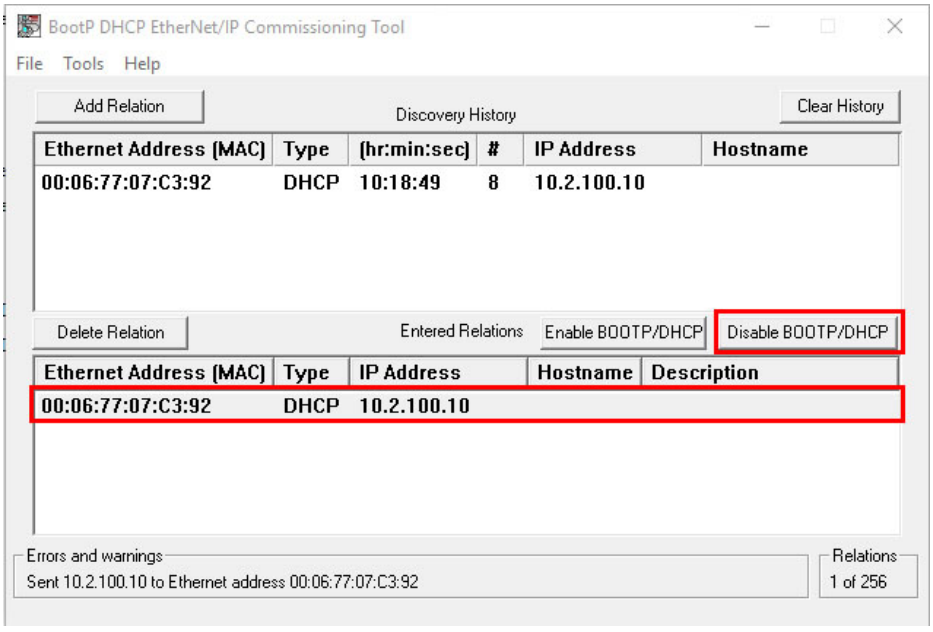


*Figure 25: **Disable BOOTP/DHCP** button*

6. If the button does not respond to this request, the **Disable BOOTP/DHCP** option can alternatively be activated via the **Disable BootP/DHCP** menu option under **Tools**.

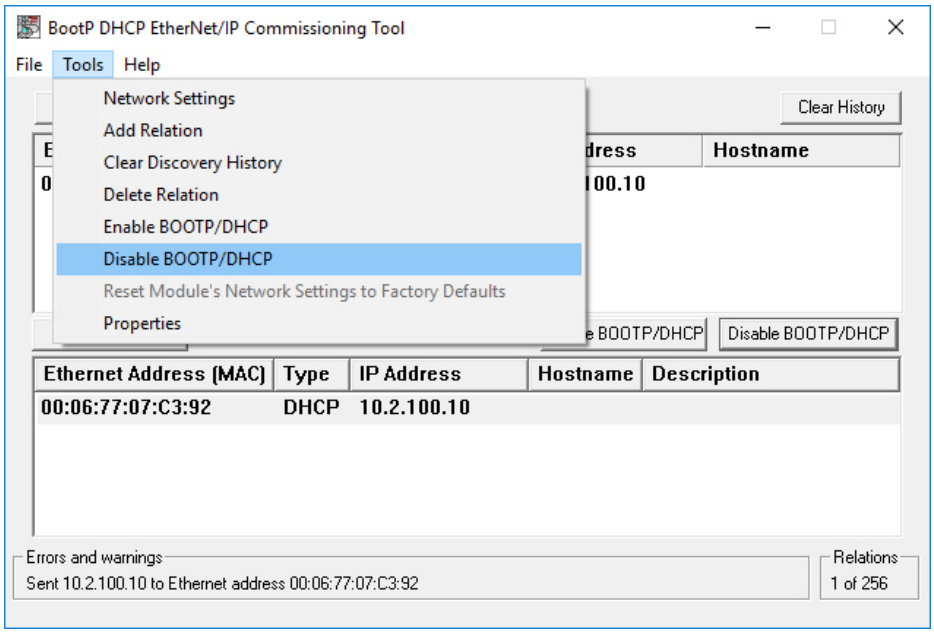


*Figure 26: Alternative activation of the **Disable BOOTP/DHCP** button*

✓ If the **Disable DHCP** command was successful (see status message at **Errors and warnings**), then the encoder is ready for use in the network with the newly assigned IP address.

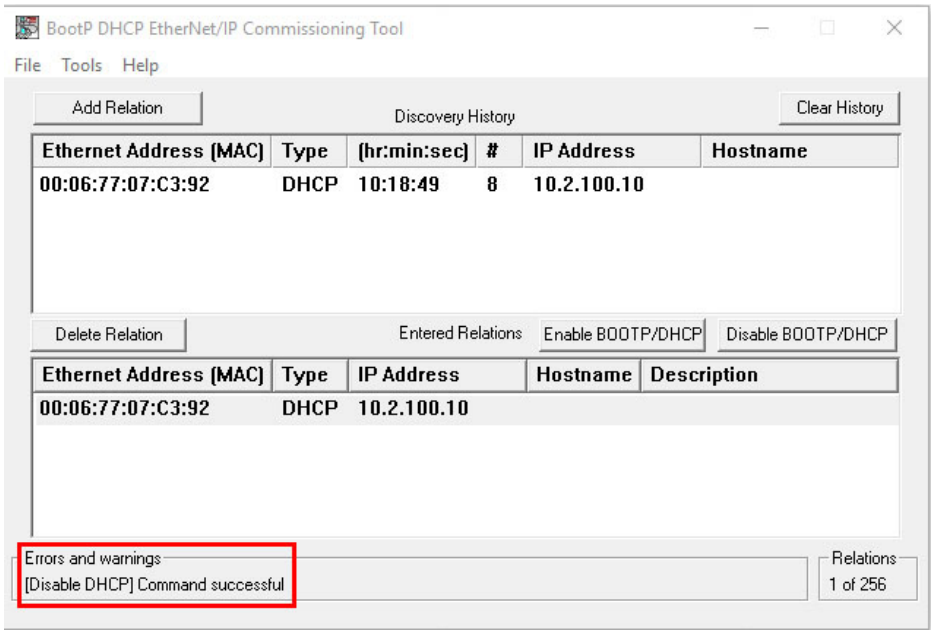


*Figure 27: **Disable DHCP** status message*

✓ Leaving the address switches in the “000” position ensures that the encoder always reads the IP address correctly assigned via DHCP from the encoder’s EEPROM at every restart.

5.2.4 Checking integration into EtherNet/IP via RSLinx Classic

The **RSLinx Classic** tool can be used to check again whether the control detects the set IP address.

1. Start **RSLinx Classic** (usually in the start menu of your PC/notebook under **Rockwell Software** → **RSLinx** → **RSLinx Classic**).
2. In the program, click on the **RSWho** button.

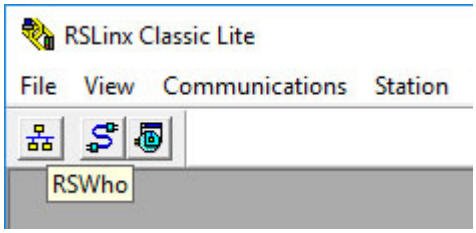


Figure 28: RSWho button in RSLinx Classic

3. Open the **AB_ETHIP-1** → **Ethernet** path.

✓ The encoder is visible under its IP address.

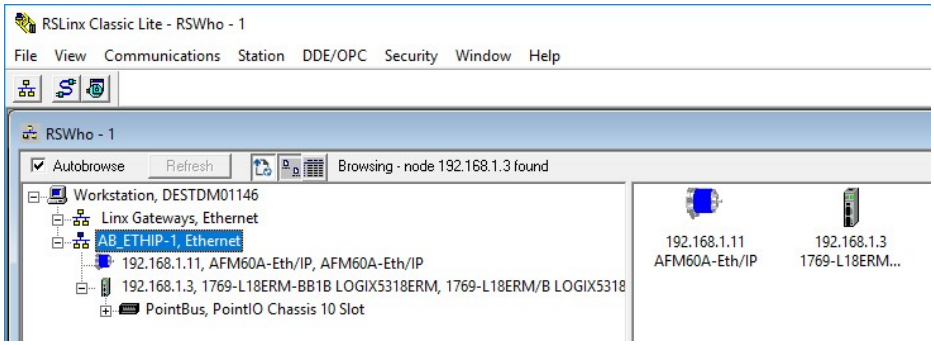


Figure 29: Encoder in the AB_ETHIP-1 of RSLinx Classic path

5.3 Creating a project in the control software

1. Start the **RSLogix 5000** control software (usually in the start menu of your PC/notebook under **Rockwell Software** → **RSLogix 5000 Enterprise Series** → **RSLogix 5000**).
2. In the **File** menu, open a new project with the **New...** command
3. Project the hardware.

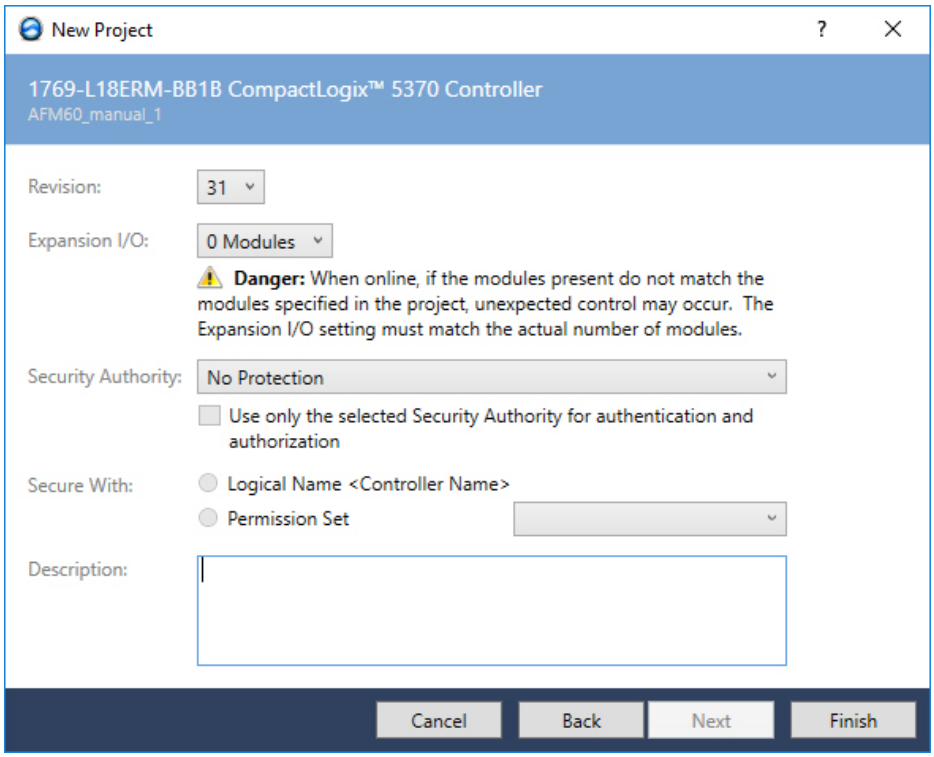


Figure 30: Projecting the hardware

4. Click **OK** .
✓ The **RSLogix 5000 [Name]** window opens.

NOTE
Type and Chassis Type must match your controller.

You can then integrate and configure the encoder in the project in three ways:

- Using an EDS file (see "Integration and configuration using an EDS file", page 52)
- Using the function block (see "Function block", page 66)
- As generic module (see "Integration of the encoder as generic module", page 67)

Due to the fast and easy integration, we recommend integrating the encoder using an EDS file.

Please note that with older encoders before DateCode (YYWW) 1535 (firmware version before 2.1.8), only integration as generic module is possible.

5.4 Integration and configuration using an EDS file

The EDS file (electronic data sheet) contains all information about the parameters and the operating modes of the absolute encoder. The absolute encoder can be configured and put into operation using the EDS file.

5.4.1 Requirements

- An Allen-Bradley control system with **RSLogix 5000** control software V22 or later is used (or another controller that allows integration using an EDS file).
- The encoder is integrated into the EtherNet/IP network (see "IP address of the encoder", page 46).
- The EDS file has been integrated into the control software using the Rockwell Hardware Installation Tool.

5.4.2 Setting up communication

1. Right-click the **Ethernet** symbol and select the **New Module...** command.

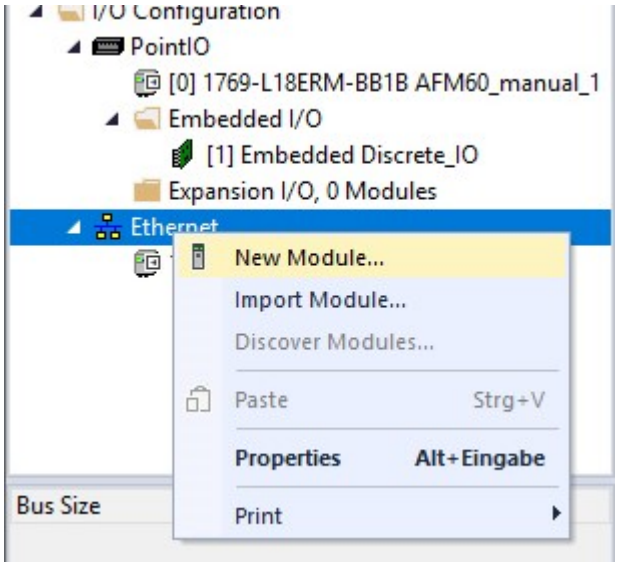


Figure 31: Integrating encoder using EDS

✓ The **Select Module Type** dialog opens.

2. Select the respective encoder type in the **Catalog** tab.

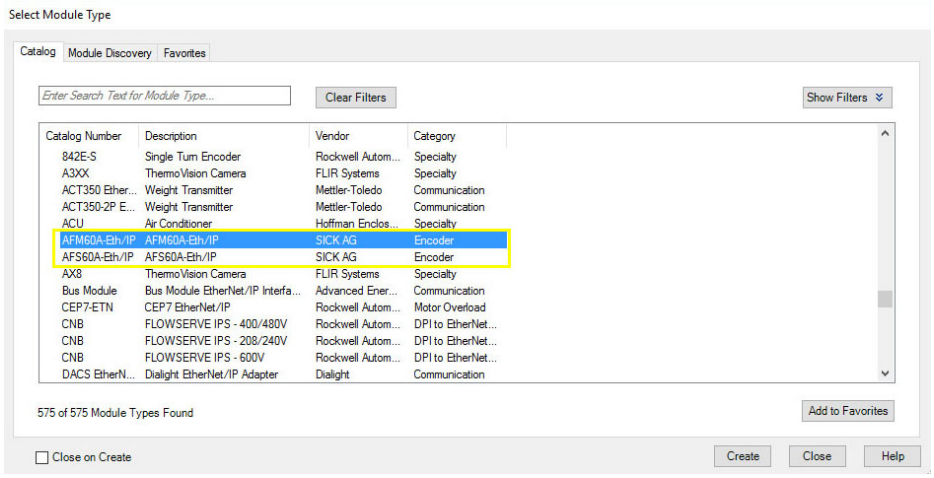


Figure 32: Select module

Depending on the connected type, the following designation is displayed:

- **AFS60A-Eth/IP** for the AFS60 EtherNet/IP
- **AFM60A-Eth/IP** for the AFM60 EtherNet/IP

3. Click **OK** .

✓ The **Module Properties [Modulname]** dialog opens.

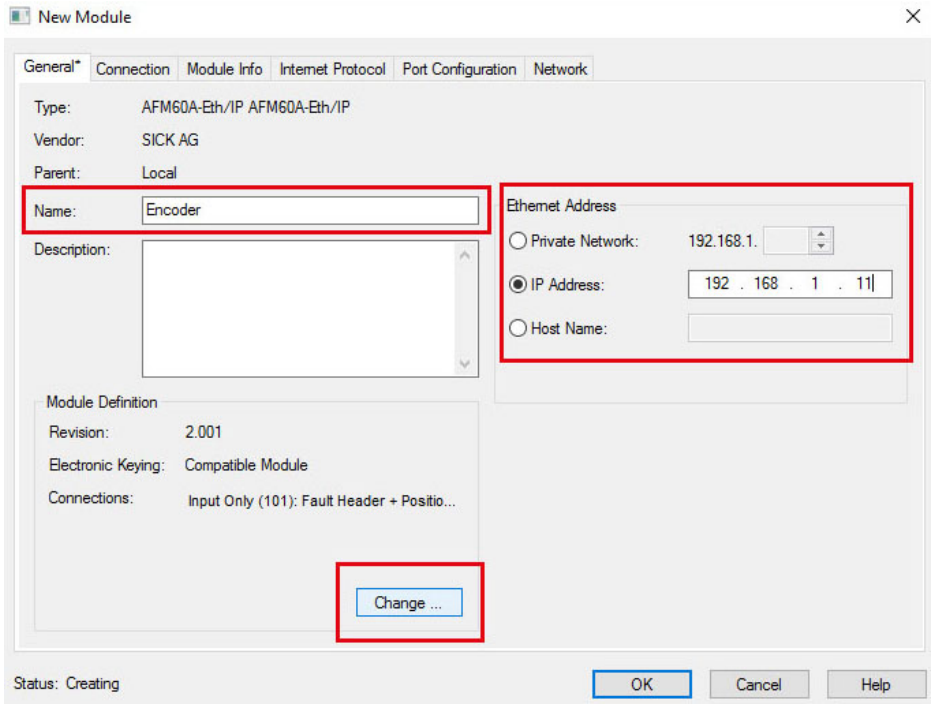


Figure 33: Entering module properties

4. Enter a name in the **Name** field (freely selectable) and enter the IP address defined for the encoder in the **IP Adress** field (see "IP address of the encoder", page 46).
   In the **Module Definition** area, the **Input Only (101)** default connection is displayed as **Connections** . This is instance 101 of the assembly object (see table 18, page 23).
5. If you want to change this instance, click on **Change...**.

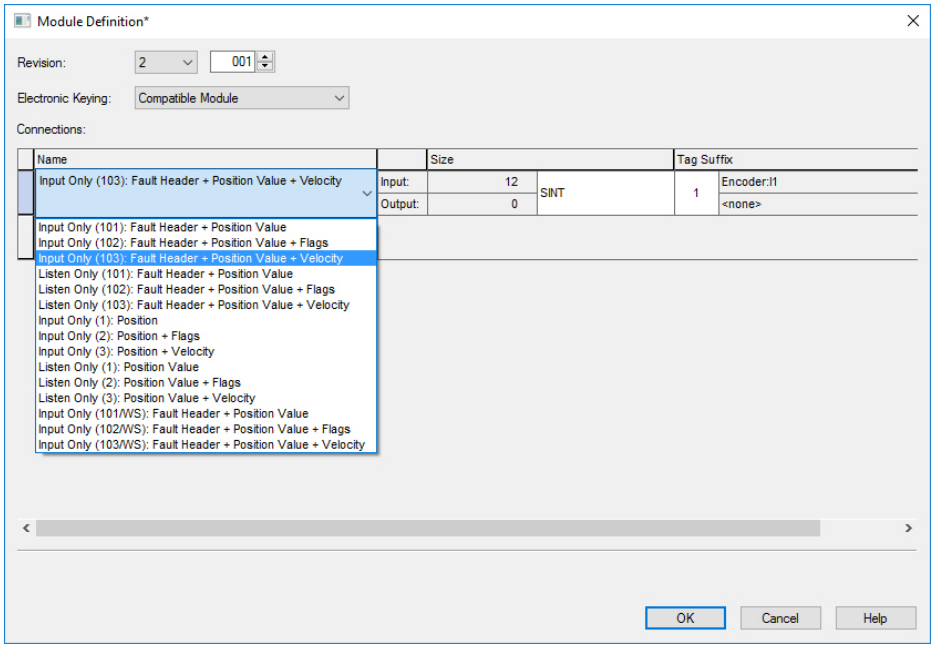


Figure 34: Changing connections

6. For example, select Input Only 103. This instance contains errors, position value and speed of the encoder.

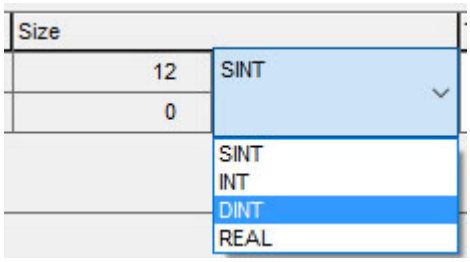


Figure 35: Changing data format

7. Under **Size**, select the **DINT** data format.
8. Then click on **OK** .

Checking communication

The data received by the controller from the encoder can be displayed in order to check that communication between the controller and the encoder is working correctly.

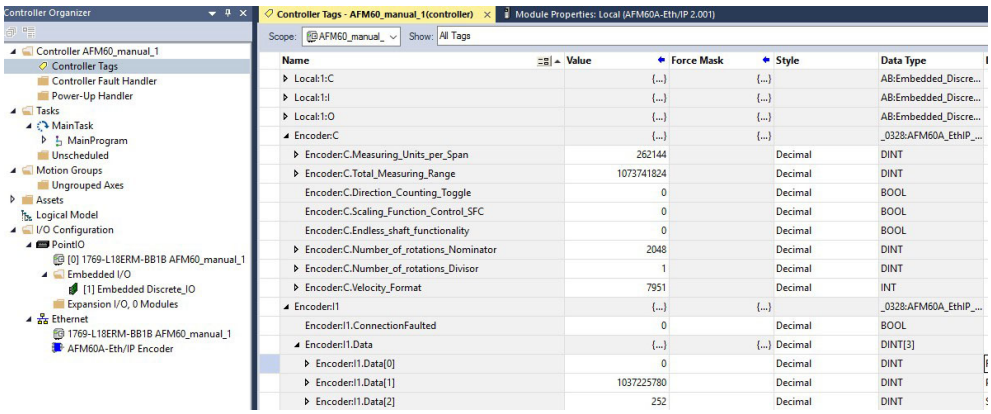


Figure 36: Checking communication

1. In **Controller Organizer**, open the **Controller test** → **Controller Tags** folder.
2. In **Controller Tags**, in the **Name** column, open the **AFx60_EIP:I1** → **AFx60_EIP:I1.Data** item.

Displayed data in the example:

- **AFx60_EIP:I1.Data[0]**: Fault header: 0
- **AFx60_EIP:I1.Data[1]**: Position: 1037225780
- **AFx60_EIP:I1.Data[2]**: Speed: 252

5.4.3 Configuration

Encoder:C	{...}	{...}		_0328:AFM60A_EthIP_...
Encoder:C.Measuring_Units_per_Span	262144		Decimal	DINT
Encoder:C.Total_Measuring_Range	1073741824		Decimal	DINT
Encoder:C.Direction_Counting_Toggle	0		Decimal	BOOL
Encoder:C.Scaling_Function_Control_SFC	0		Decimal	BOOL
Encoder:C.Endless_shaft_functionality	0		Decimal	BOOL
Encoder:C.Number_of_rotations_Nominator	2048		Decimal	DINT
Encoder:C.Number_of_rotations_Divisor	1		Decimal	DINT
Encoder:C.Velocity_Format	7951		Decimal	INT

Figure 37: Encoder configuration

1. In **Controller Tags**, in the **Name** column, open the **AFx60_EIP:C** item.
2. Enter the parameters of the encoder (see "Parameterizable functions", page 36).

5.5 Installing the ladder routine

Two so-called ladder routines are available for integration of the web server. The ladder routine is used to map the configuration data between the controller and the web server.

Use the following ladder routine depending on the selected instance:

- SickAFx_A101WS_A103WS_FB_Enc1_GetSet.L5X for instances 101WS and 103WS
- SickAFx_A102WS_FB_Enc1_GetSet.L5X for instance 102WS

Requirements for the installation of the ladder routine are:

- Installation file of the ladder routine, downloaded from the encoder's web server (see "Ladder routine", page 103).
- Correct installation of the current EDS file (see "Integration and configuration using an EDS file", page 52).
- Selection of instance 101WS, 102WS or 103WS when configuring the encoder module.

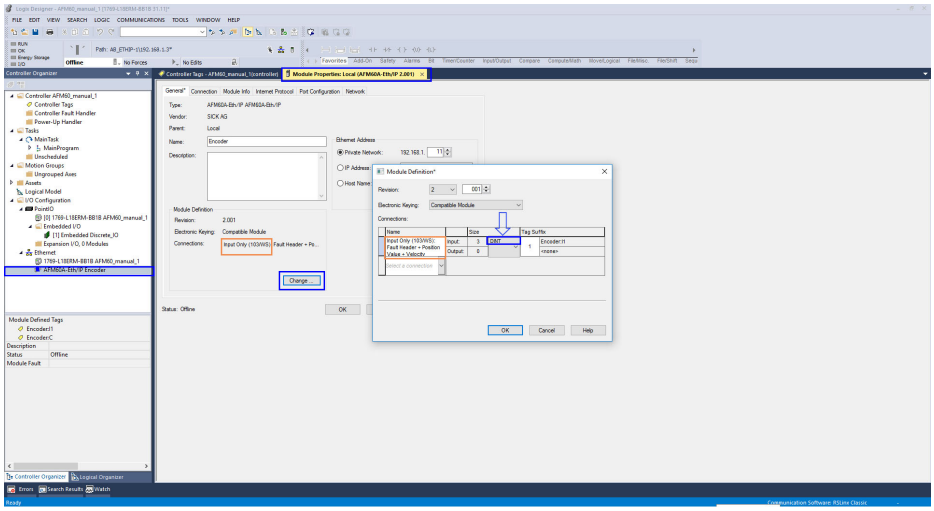


Figure 38: Selection of the instance (in the example 103WS)

- An executable project with the absolute encoder in the **RSLogix 5000**.

The following steps must be performed:

- The ladder routine must be imported and some parameters must be configured during the import.
- The ladder routine must be integrated as a SubRoutine into the MainRoutine of your project.
- The encoder can then be configured both from the controller (in the controller tags) and using the web server.

NOTE

If multiple encoders are used, the routine must be imported several times and given its own unique so-called **Final Name** during import. In addition, the **Tag References** must be uniquely named for each encoder.

5.5.1 Import of the ladder routine

1. In the of **MainProgram** context menu, select the **Import Routine...** command.

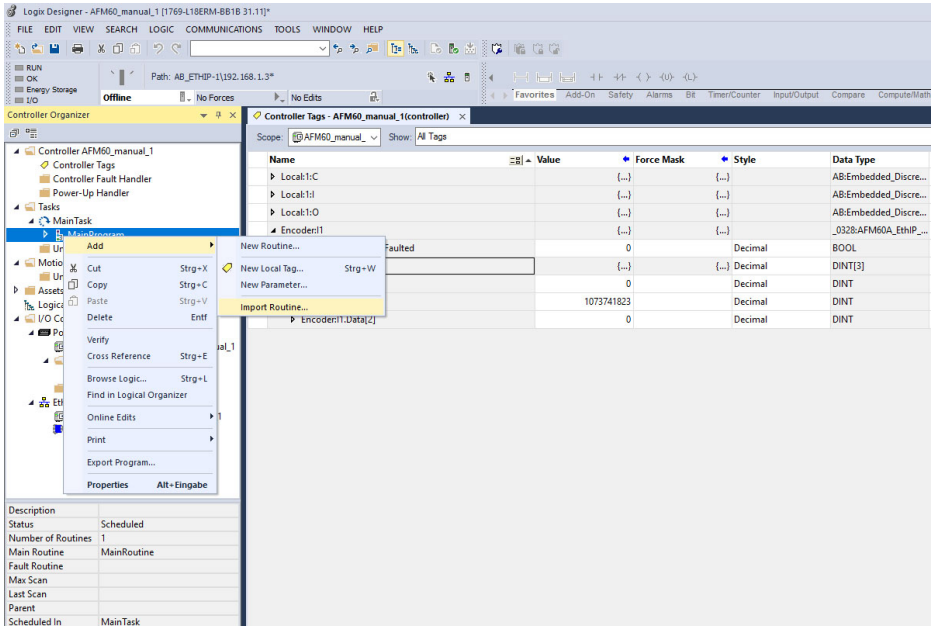


Figure 39: Selection of the Import Routine command...

Depending on whether instance 101WS or 103WS or the instance 102WS of the assembly object is used (see table 18, page 23), the appropriate ladder routine must be selected.

2. Select the **Sick-AFx_A102WS_FB_Enc1_GetSet.L5X** file via **Add** and click on **Import Routine**.

✓ The **Import Configuration** dialog opens.

NOTE

▶ Do not click on **OK** until all configuration steps for import have been completed. If you click OK by mistake, you have to restart the import (see figure 39, page 56).

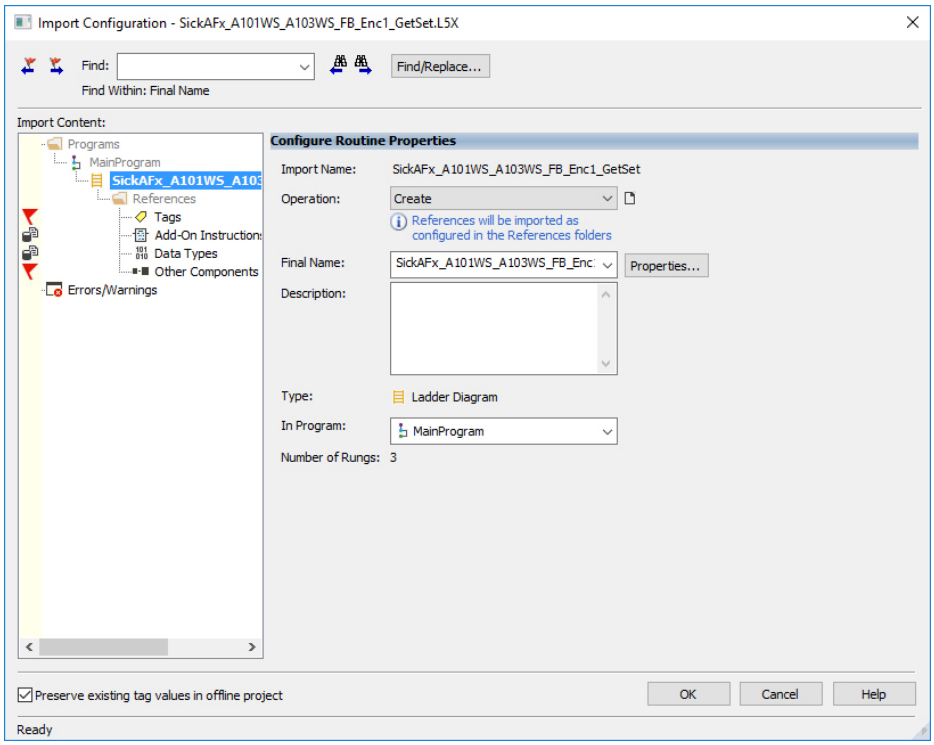


Figure 40: Import Configuration dialog

3. If necessary, change the name of the routine in the **Final Name** field.
   If several encoders are integrated into the project, then a unique final name must be assigned to the routine for each encoder.
4. Select the **Other Components** item.
5. In the **Final Name** column, select the drop-down list.
6. Select the encoder module for which the ladder routine is to be imported.

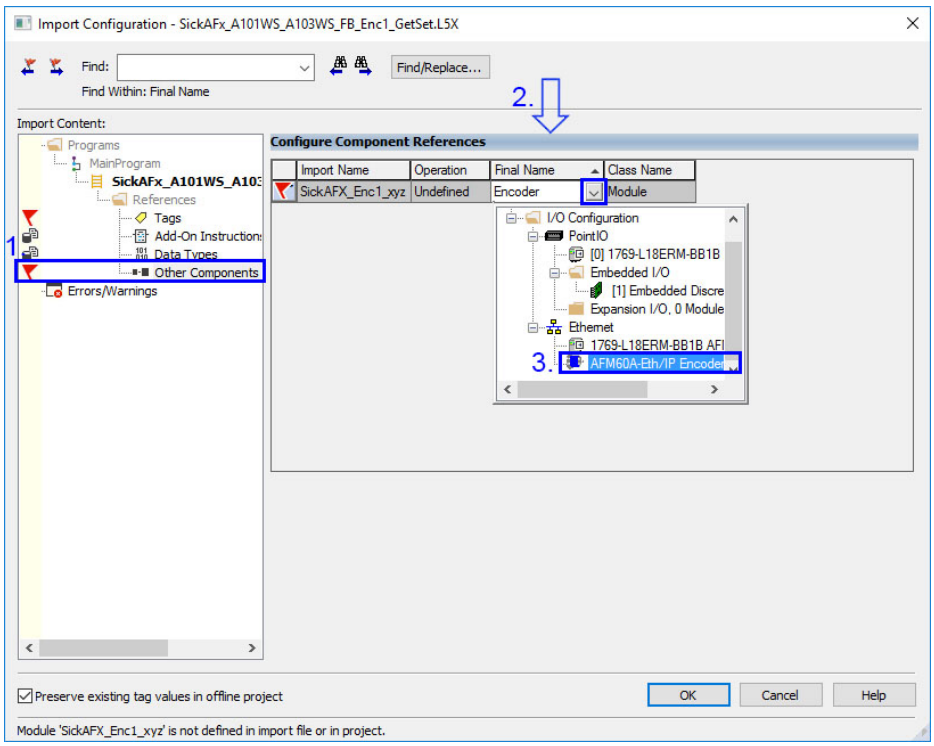


Figure 41: Encoder selection

7. In the **Operation** column, select the **Use Existing** option.

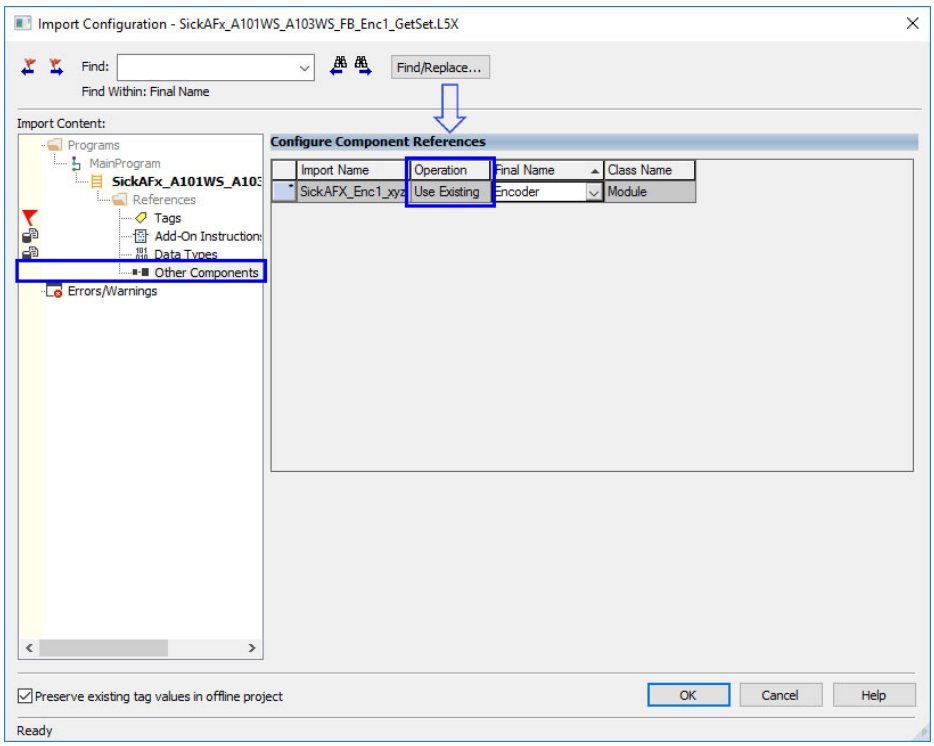


Figure 42: Selection of the operation for the component

8. Go to **Import Content** and select **Tags** .
9. In the **Final Name** column, select the drop-down list.
10. Select the encoder module whose tags are to be adjusted.

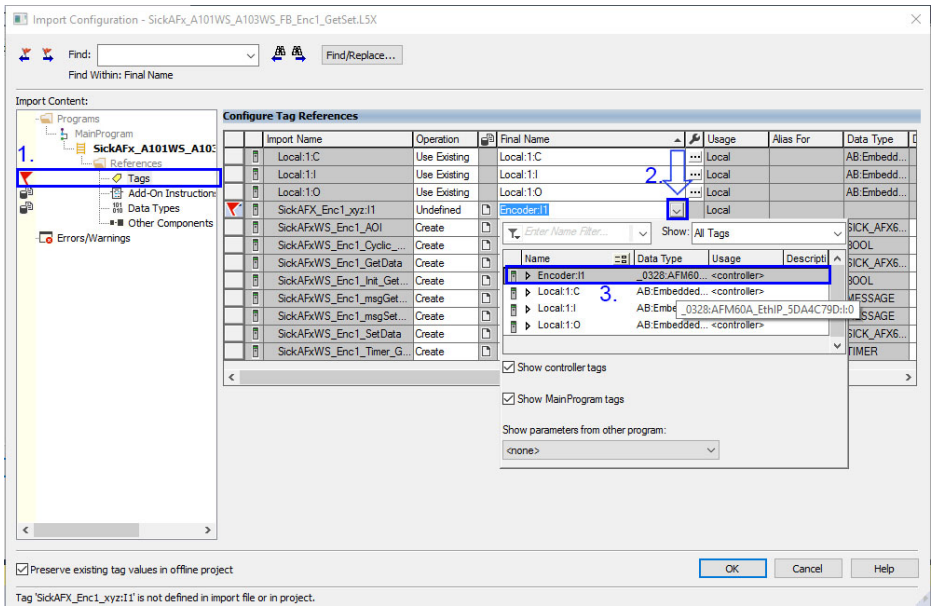


Figure 43: Selection of the tag of the instance used

11. In the **Operation** column, select the **Use Existing** option.

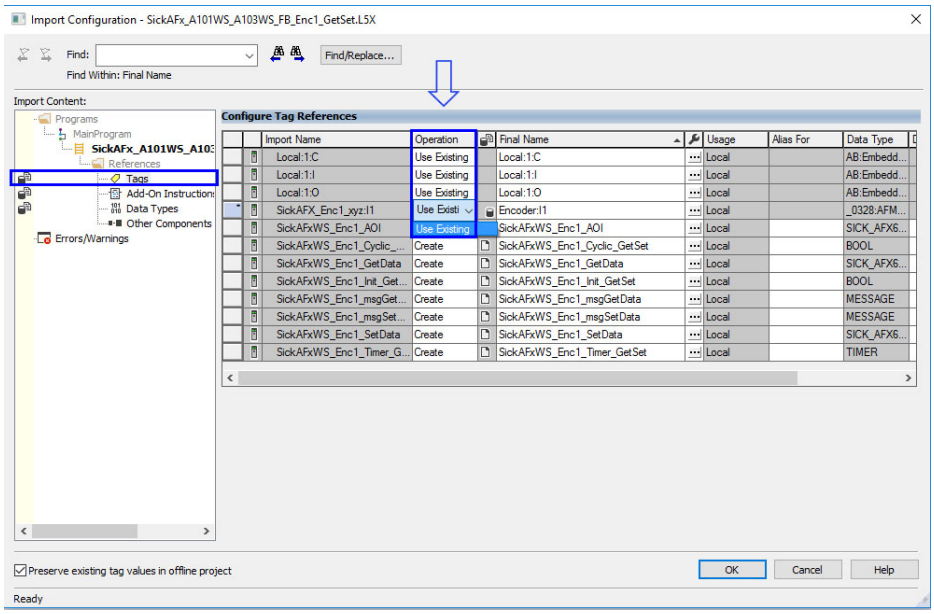


Figure 44: Selection of the operation for the tag references

12. If necessary, change the names of **Tags** in the **Final Name** column.
If multiple encoders are used in a project, then each final name may only be assigned once. Then the name is to be changed, for example, from ... Enc1... to ...Enc2....

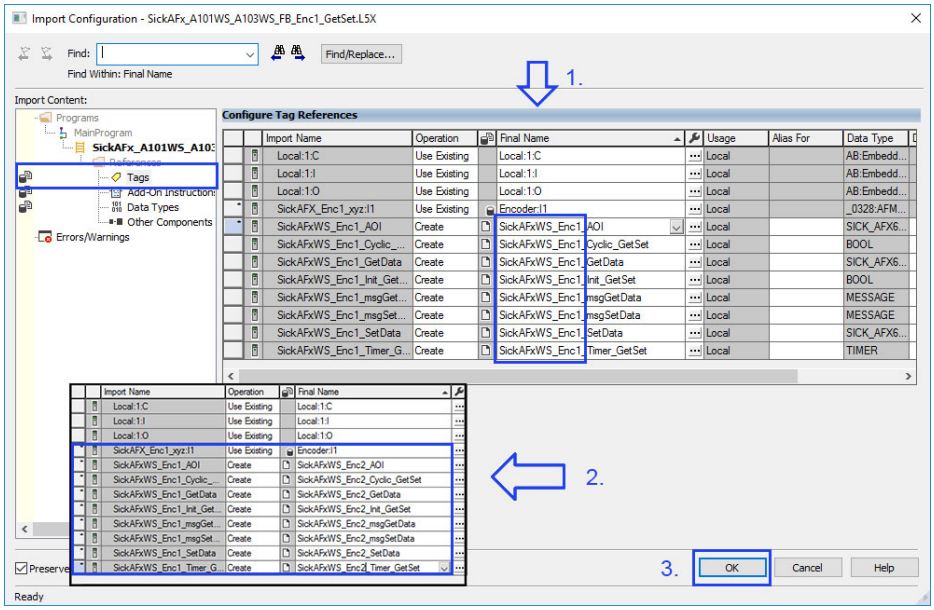


Figure 45: Changing the tag names

13. Click **OK** .

✓ The ladder routine is imported.

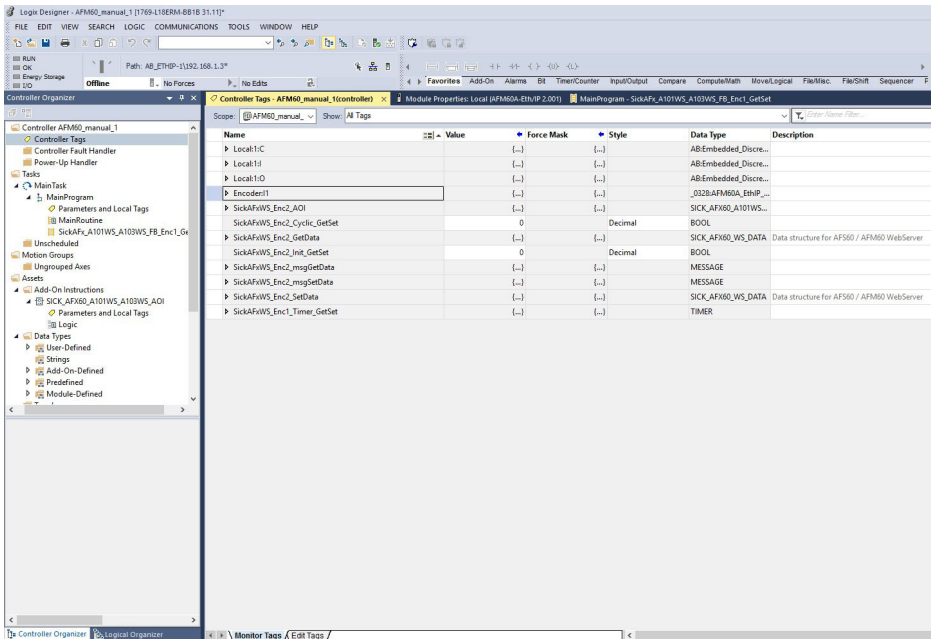


Figure 46: Project structure after import

5.5.2 Integration as SubRoutine in MainRoutine

The ladder routine must be integrated as the **SubRoutine** into the **MainRoutine** of your project.

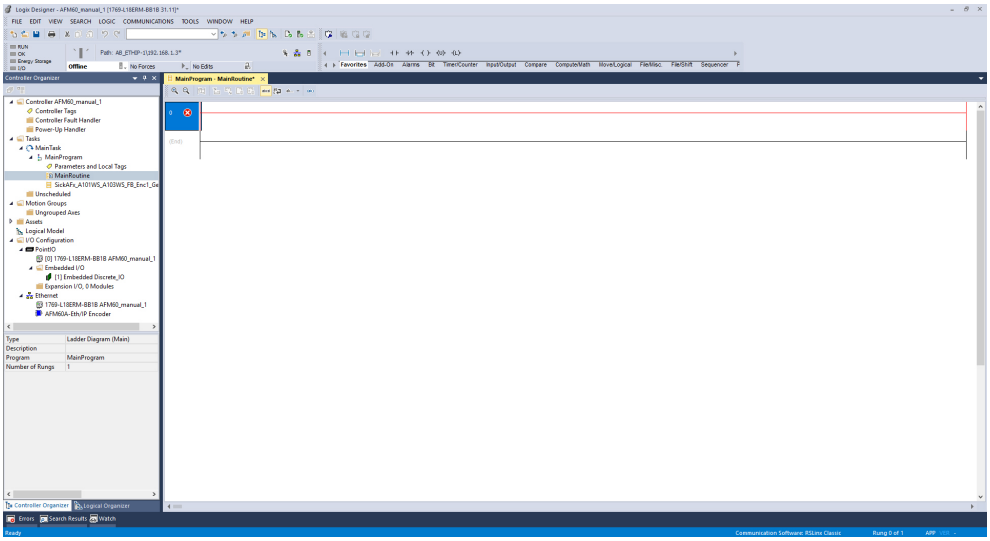


Figure 47: MainRoutine without SubRoutine

▶ As shown in the example, integrate the SickAFx ladder routine as the SubRoutine with the **JSR** (Jump To Subroutine) command.

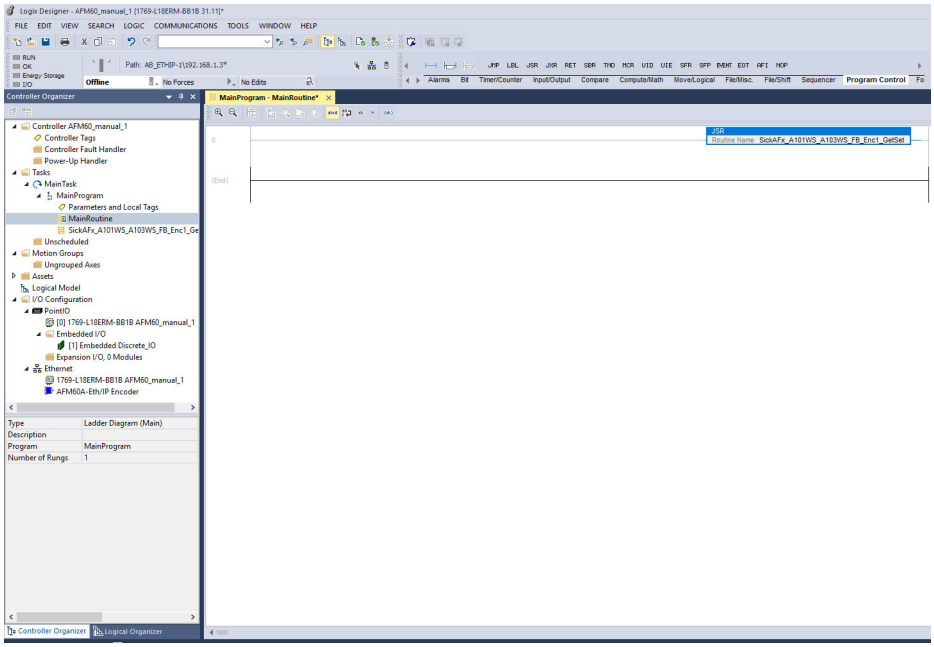


Figure 48: MainRoutine with SubRoutine

5.5.3 Using the SubRoutine

1. Switch the controller to online mode.

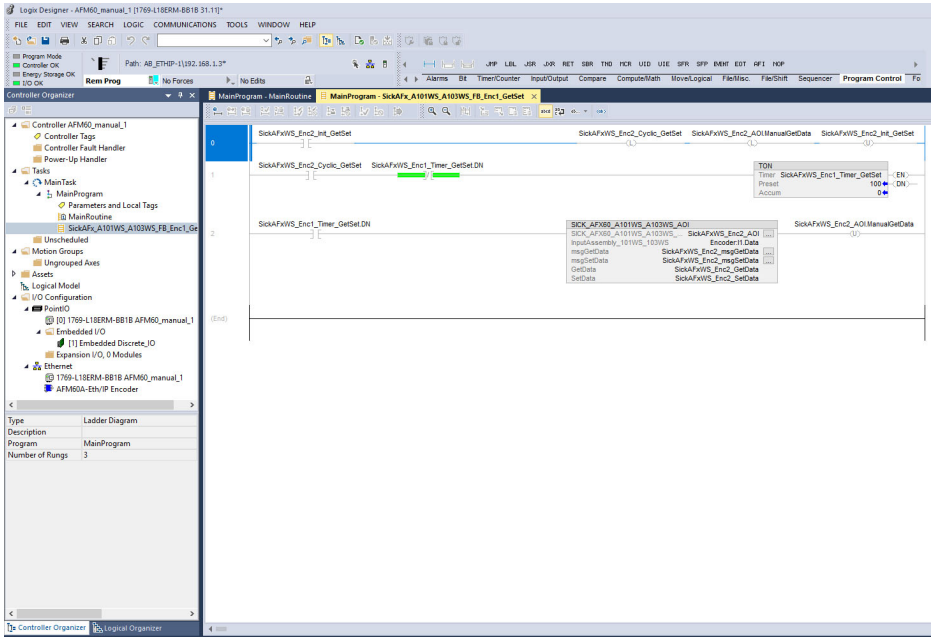


Figure 49: Imported SickAFx ladder routine in online mode

2. In the MainProgram, switch to **SickAFx_A101WS_A103WS_FB_Enc1_GetSet** .

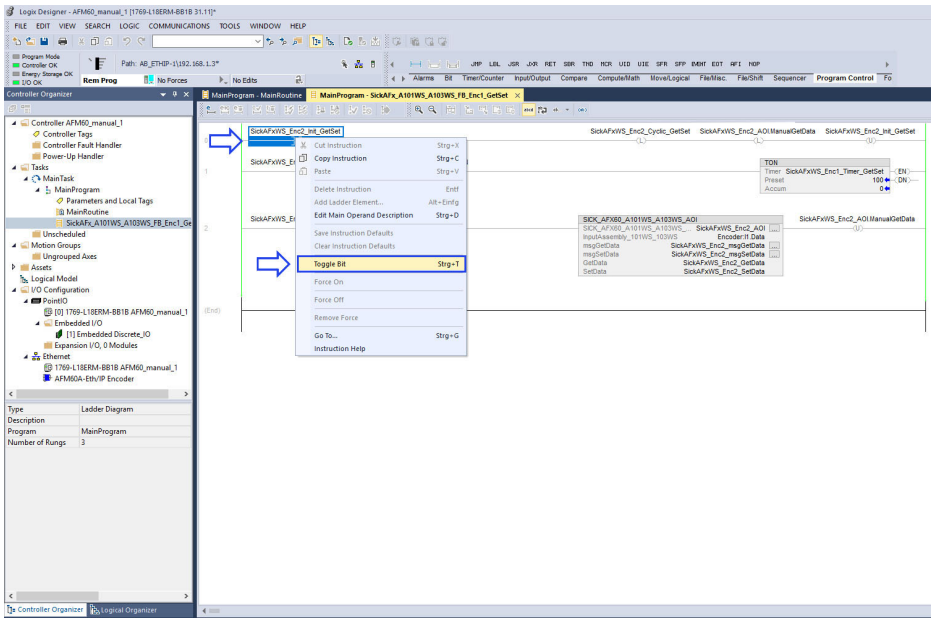


Figure 50: Initialization and start of the SubRoutine

3. In the context menu of **SickAFxWS_Enc1_Init_GetSet**, activate the **Toggle Bit** command.

✓ This completes the integration and parameterization of the encoder can be performed both on the controller side and via the web server.

5.5.4 Reading out and changing parameters of the encoder

Under **Controller Tags**, the parameters of the encoder can be read out in the **SickAFxWS_Enc1_GetData** node.

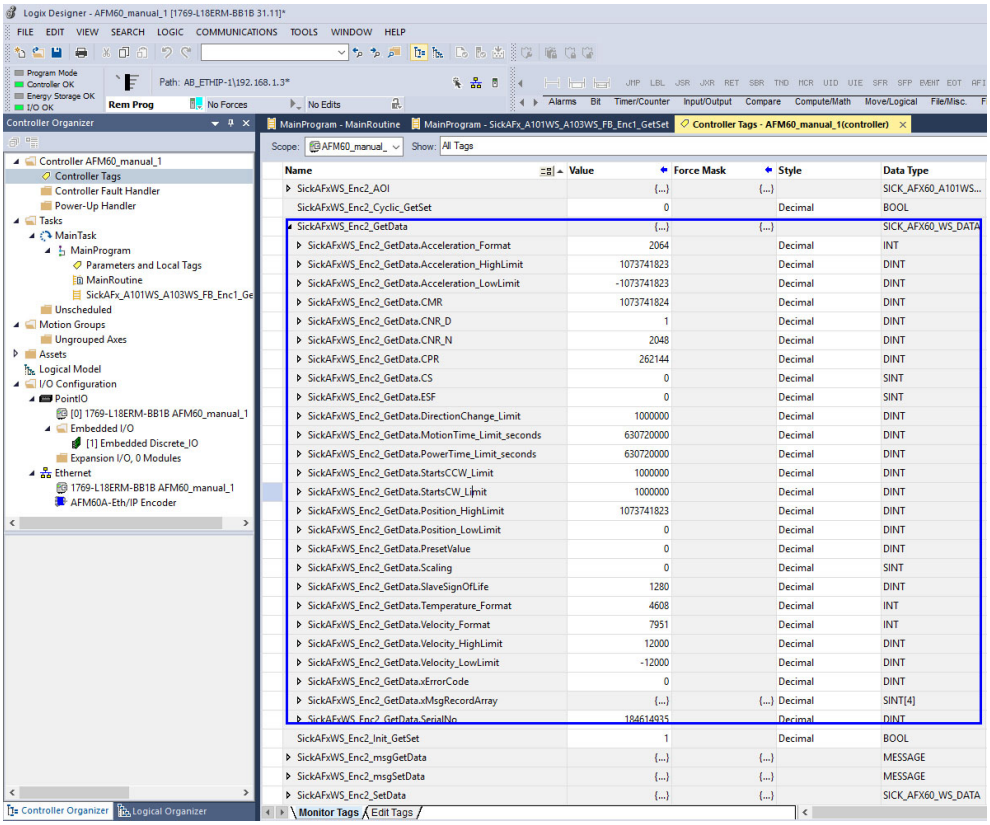


Figure 51: Reading out the parameters under GetData

The parameters changed in the web server are displayed in the controller.

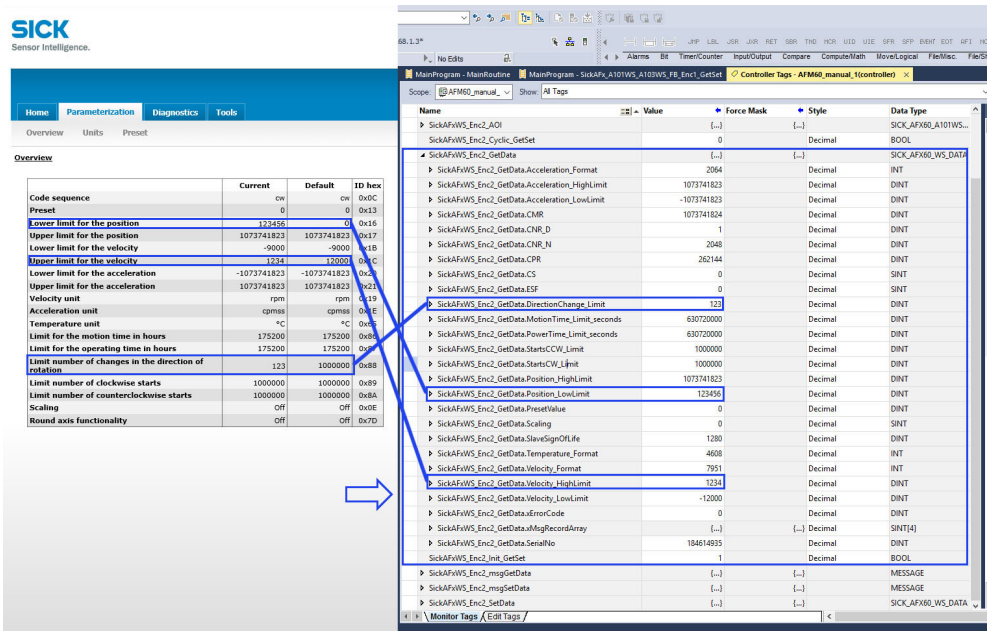


Figure 52: Example of changing data in the web server and reading out the parameters in the controller

NOTE

If a parameter is changed via the web server, bit 15 in the Fault header is automatically set as a warning (see table 33, page 108).

Under **Controller Tags**, the parameters of the encoder can be changed in the **Sick-AFxWS_Enc1_SetData** node.

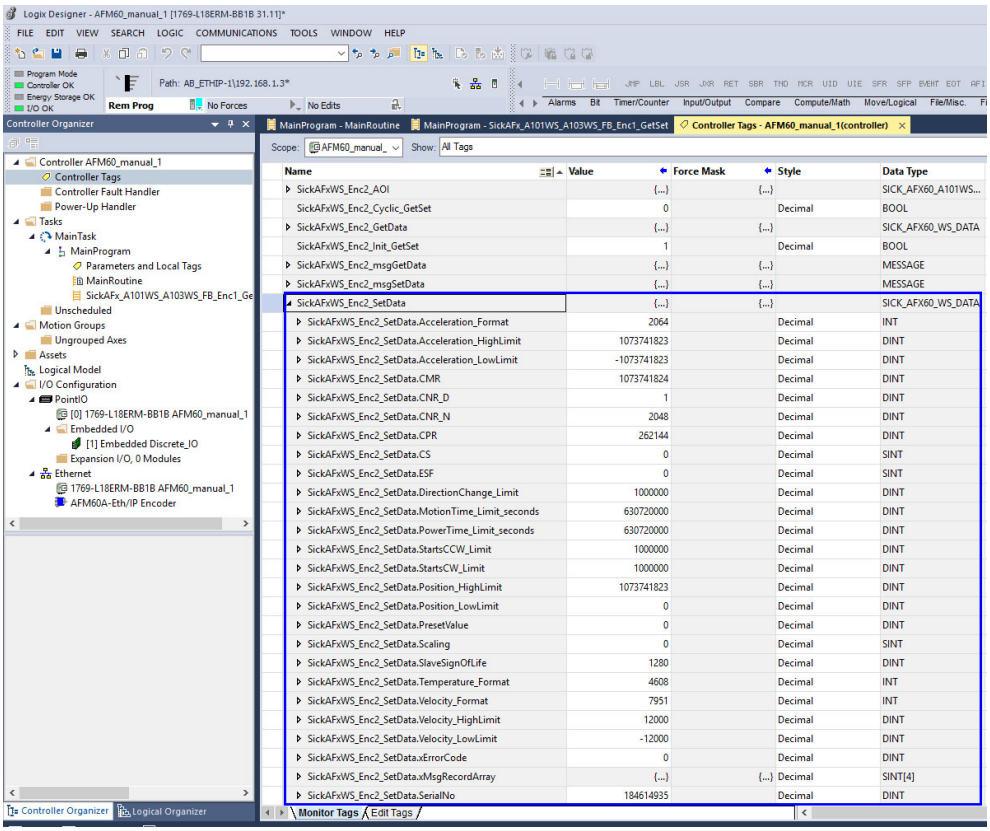


Figure 53: Changing parameters under SetData

Parameters that are changed in the controller are displayed in the web server on the Parameterization page.

NOTE

The web browser must be updated to display the changed data.

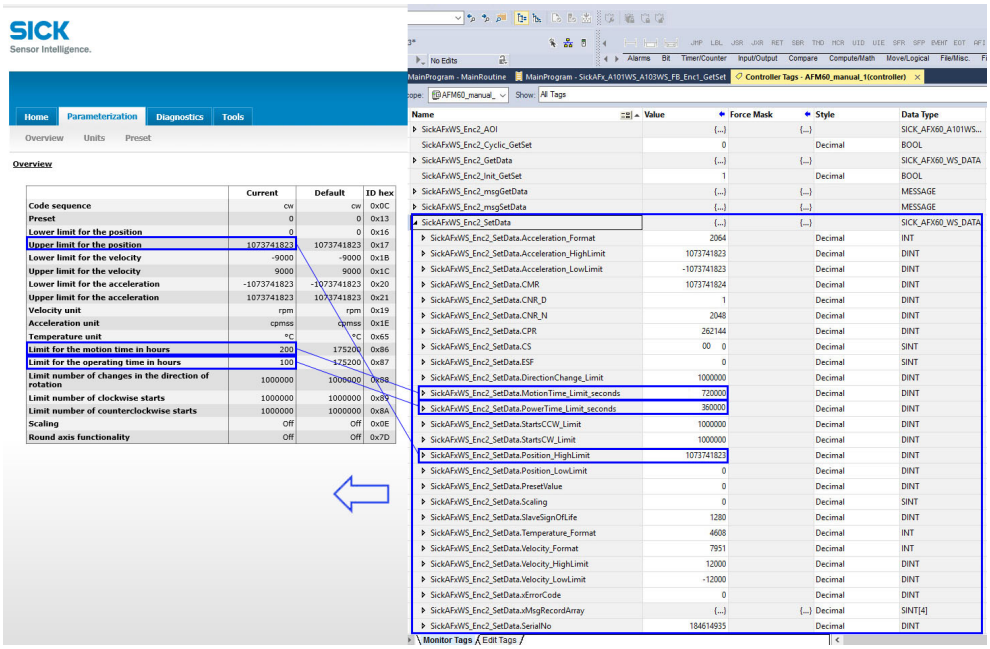


Figure 54: Example of changing data in the controller and reading it out in the web server

DANGER

Before changing the preset value, check whether there is any danger from the machine or system in which the encoder is integrated!

As soon as the value has been entered and the entry has been confirmed with the [Enter] key, it is accepted as the position value (see figure 112, page 98).

5.6 Function block

A function block can be used for communication between an Allen-Bradley controller and the absolute encoder.

5.6.1 Requirements

- Function block and complete documentation downloaded from SICK homepage: “EthernetIP function block - EtherNet/IP function block for encoder-specific additional functions for RSLogix5000 including operating instructions”.
- The encoder must be integrated into the controller using an EDS file or as a generic module.

5.6.2 Import and wiring

In order to be able to use the function block in the RSLogix 5000 software, import the component into a project as an add-on instruction (file name: SICK_AFx60_Vxxx.L5X).

The function block must then be called up and wired. Only with valid wiring it is possible to read parameters from or write parameters to the encoder.

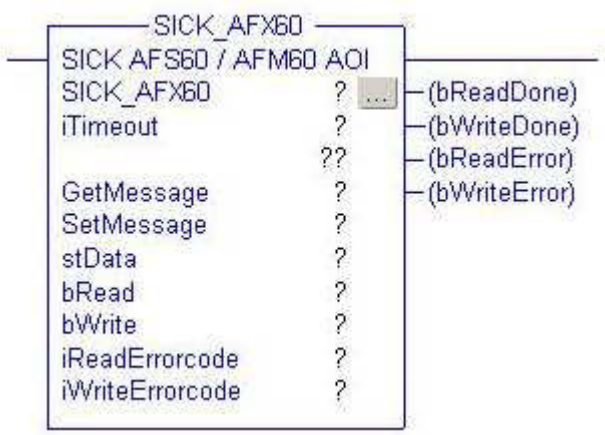


Figure 55: Function block in the Rockwell control

A detailed description of the wiring can be found in the “AFS60/AFM60 EtherNet/IP Add-On Instruction” operating instructions. These operating instructions are supplied with the function block as a PDF.

5.7 Integration of the encoder as generic module

1. Right-click the **Ethernet** symbol and select the **New Module...** command.

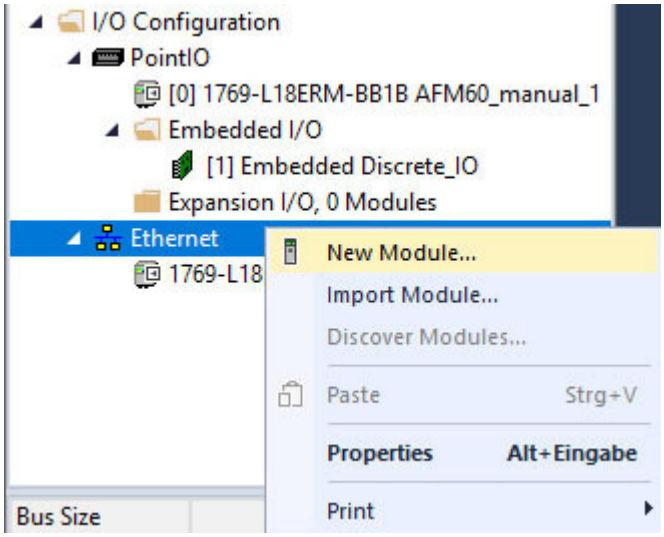


Figure 56: Integrating encoder

✓ The **Select Module** dialog opens.
2. Search for “generic”.
3. Select the marked **ETHERNET-MODULE (Generic Ethernet Module)** module.

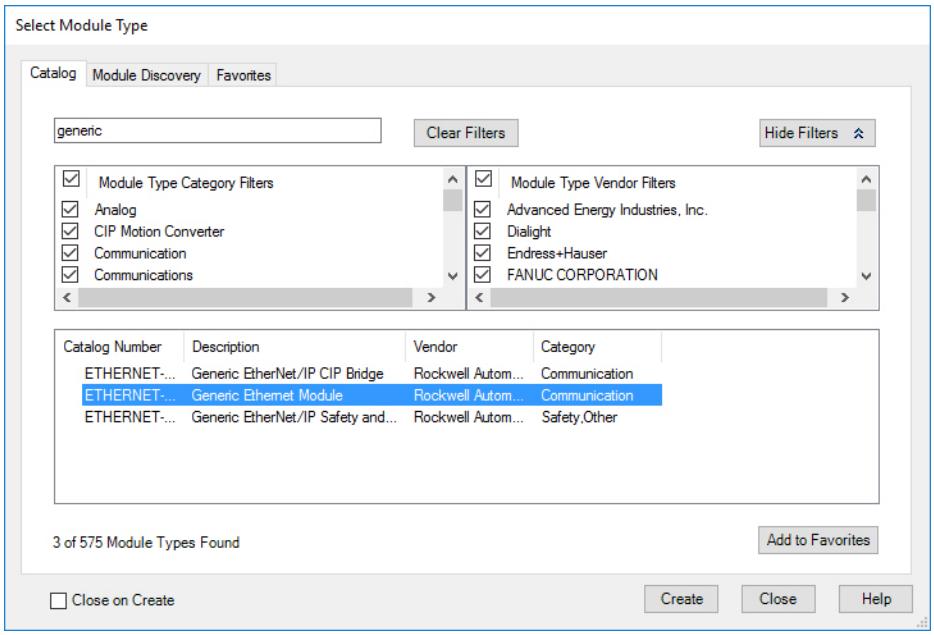


Figure 57: Select module

4. Click **Create** .

✓ The **Module Properties** dialog opens.

5.7.1 Module settings

1. In the **Module Properties [Modulname]** dialog, enter the **IP address** assigned for the encoder (see "IP address of the encoder", page 46).
2. Enter the settings for **Input**, **Output** and **Configuration**.

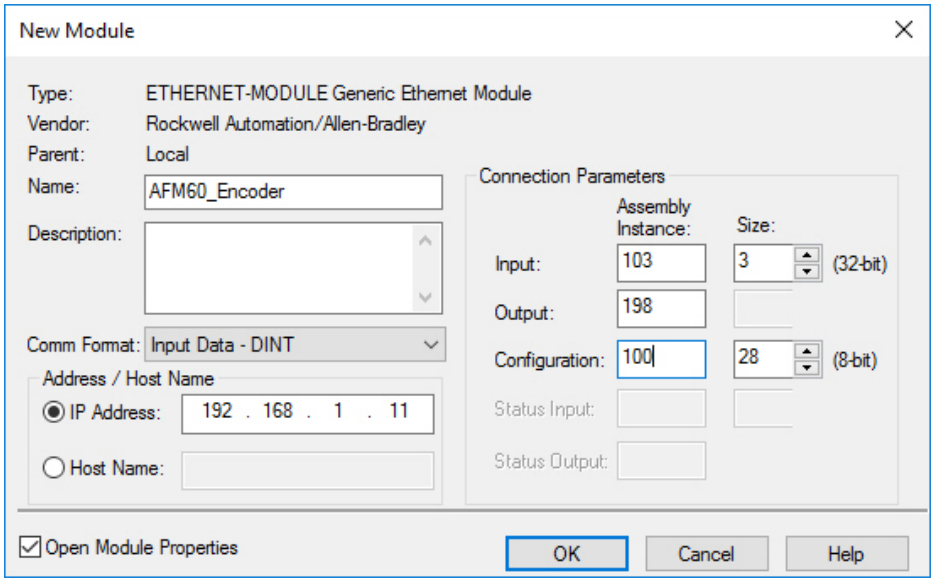


Figure 58: Entering module properties

Example:

- **Name:** AFM60_Encoder (name is freely selectable)
- **Comm Format:** Input data - DINT
- **IP Address:** 192.168.1.123
- **Input:** Assembly instance: 103; size: 3
  This selects instance 103 of the assembly object (see table 18, page 23).
  The size is 3 × 32 bits (= 12 bytes).

- **Output**: Assembly instance: 198
  Since the absolute encoder does not process an output assembly, the Output parameter is set to 198 (Input Only).
- **Configuration**: Assembly instance: 100; Size: 28
  This selects instance 100 of the assembly object (see table 18, page 23).
  The size is 28 × 8 bits (= 28 bytes).

NOTE
Instance 100 of the assembly object represents the configuration assembly. If this is called up, it must not be empty. Accordingly, the configuration assembly must be filled with valid data (see table 20, page 26) beforehand. Otherwise, the controller may output an error (see "Error messages of the Allen Bradley control system", page 110).

3. Click **OK** .

Example data for a configuration assembly

The data of the configuration assembly is transmitted (see table 20, page 26) in the previously configured 28 bytes of instance 100.

These can be viewed at **Controller Tags** in the following column: **Name** at **AFM60_Encoder:C → AFM60_Encoder:C.Data**.

NOTE
The low byte is displayed before the high byte.

Name	Value	Force Mask	Style
AFM60_Encoder:C.Data	{...}	{...}	Hex
AFM60_Encoder:C.Data[0]	16#00		Hex
AFM60_Encoder:C.Data[1]	16#00		Hex
AFM60_Encoder:C.Data[2]	16#00		Hex
AFM60_Encoder:C.Data[3]	16#00		Hex
AFM60_Encoder:C.Data[4]	16#00		Hex
AFM60_Encoder:C.Data[5]	16#10		Hex
AFM60_Encoder:C.Data[6]	16#00		Hex
AFM60_Encoder:C.Data[7]	16#00		Hex
AFM60_Encoder:C.Data[8]	16#00		Hex
AFM60_Encoder:C.Data[9]	16#80		Hex
AFM60_Encoder:C.Data[10]	16#00		Hex
AFM60_Encoder:C.Data[11]	16#00		Hex
AFM60_Encoder:C.Data[12]	16#00		Hex
AFM60_Encoder:C.Data[13]	16#01		Hex
AFM60_Encoder:C.Data[14]	16#00		Hex
AFM60_Encoder:C.Data[15]	16#00		Hex
AFM60_Encoder:C.Data[16]	16#00		Hex
AFM60_Encoder:C.Data[17]	16#00		Hex
AFM60_Encoder:C.Data[18]	16#00		Hex
AFM60_Encoder:C.Data[19]	16#00		Hex
AFM60_Encoder:C.Data[20]	16#00		Hex
AFM60_Encoder:C.Data[21]	16#00		Hex
AFM60_Encoder:C.Data[22]	16#00		Hex
AFM60_Encoder:C.Data[23]	16#00		Hex
AFM60_Encoder:C.Data[24]	16#0f		Hex
AFM60_Encoder:C.Data[25]	16#1f		Hex
AFM60_Encoder:C.Data[26]	16#00		Hex
AFM60_Encoder:C.Data[27]	16#00		Hex
AFM60_Encoder:C.Data[28]	16#00		Hex
AFM60_Encoder:C.Data[29]	16#00		Hex
AFM60_Encoder:C.Data[30]	16#00		Hex
AFM60_Encoder:C.Data[31]	16#00		Hex

Figure 59: Example data for a configuration assembly

- Steps per revolution CPR = 4,096 = 1000h
  C.Data[4] 00h and **C.Data[5] 10h**
- Total resolution CMR = 32,768 = 8000h
  C.Data[8] 00h and **C.Data[9] 80h**
- Direction of rotation cw = 0
  C.Data[12] 00h
- Scaling on = 1h
  C.Data[13] 01h
- Speed format = 1F0Fh
  C.Data[24] 0Fh and **C.Data[25] 1Fh**

5.7.2 Downloading the configuration to the control

1. Load the configuration for the controller.

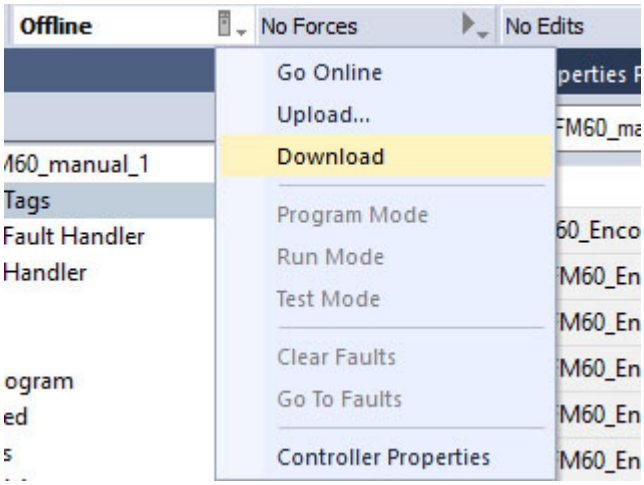


Figure 60: Loading configuration

✓ The status indicators for **Run Mode**, **Controller OK** and **I/O OK** turn green.

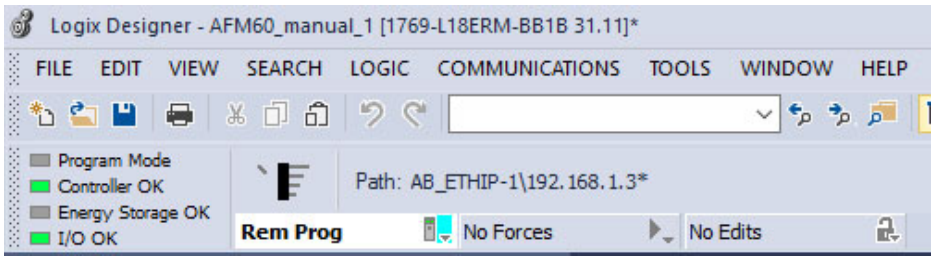


Figure 61: Communication status

5.7.3 Checking communication

The data received by the controller from the encoder can be displayed in order to check that communication between the controller and the encoder is working correctly.

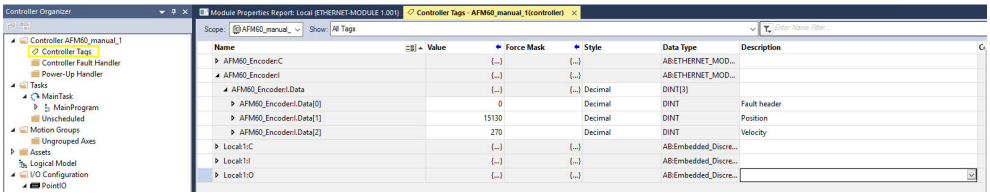


Figure 62: Checking communication

1. In **Controller Organizer**, open the **Controller AFM60_manual_1** → **Controller Tags** folder.
2. Under **Name**, open the **AFM60_Encoder:I** → **AFM60_Encoder:I.Data** item in the **Controller Tags** column.

Displayed data in the example:

- **AFM60_Encoder:I.Data[0]**: Fault header: 0
- **AFM60_Encoder:I.Data[1]**: Position: 15130
- **AFM60_Encoder:I.Data[2]**: Speed: 270 turns/min

5.8 Programming examples

The following examples show the configurations of two programs that read (temperature) or write (preset) acyclic data. For this purpose, the programs are written in the form of ladder logic using the RSLogix 5000 software from Rockwell Automation.

NOTE
During programming, the controller must be in offline mode.

Figure 63: Controller in offline mode

1. Define and declare the variables for the program.
2. Insert the blocks of the program into the ladder logic and assign the variables accordingly.
3. Start the download of the program for the controller.
4. Finally, test the program.

5.8.1 Reading out temperature

In the first example, the temperature of the encoder is to be read out using parameter 64h, Temperature Value.

Defining and declaring variables

First variables TEMP_Trigger, TEMP_OneShot, TEMP_Value and TEMP_Message have to be defined and declared for the program.

First, variable TEMP_Trigger is created, which controls the readout process.

1. Right-click in **Controller Organizer** on **Controller Tags** and select **New Tag**.

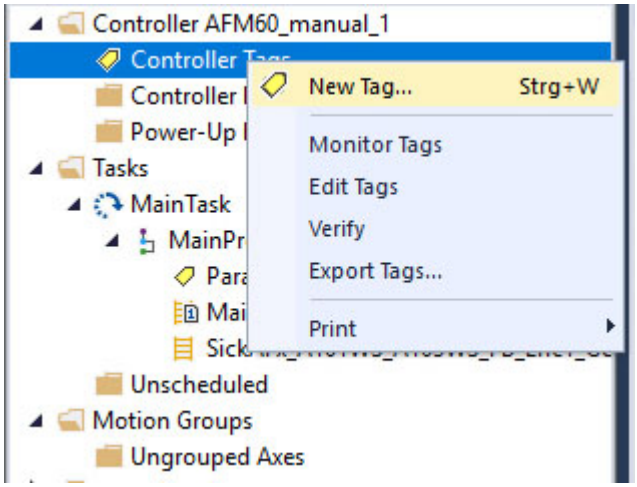


Figure 64: Create a new variable

✓ The **New Tag** dialog opens.

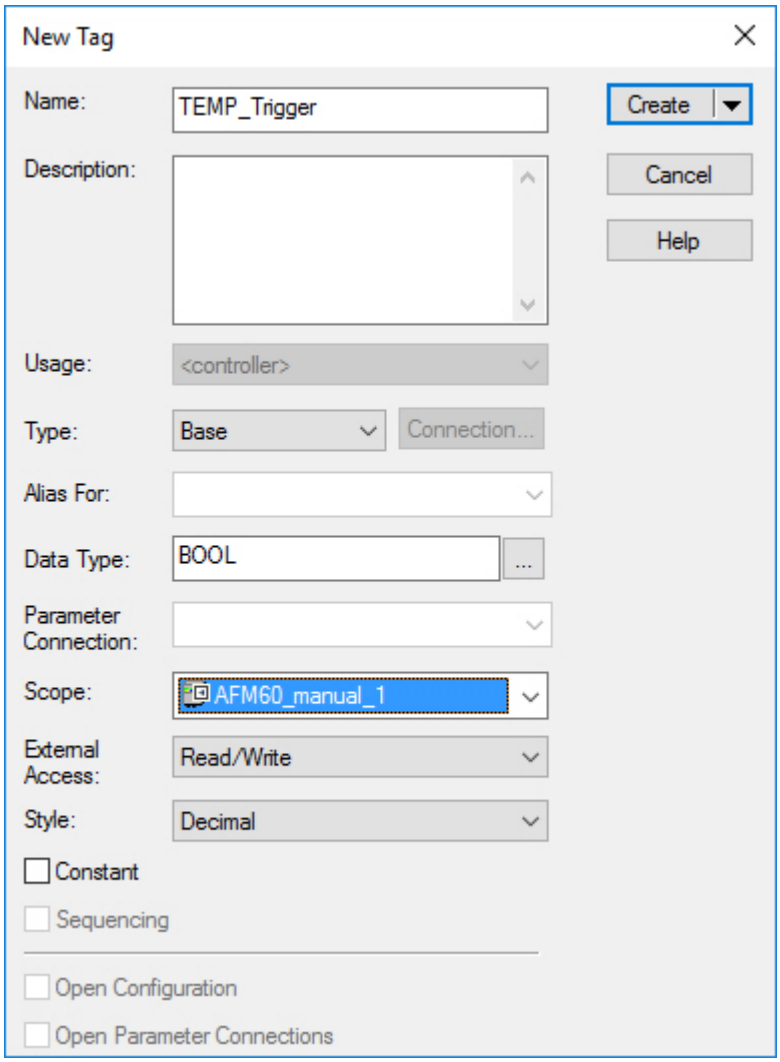


Figure 65: Definition of variable TEMP_Trigger

2. Enter TEMP_Trigger in the **Name** field, select the BOOL data type in the **Data Type** field and click on **OK**.
   To trigger the process only once, another element, in this case edge-sensitive, must be defined and declared. This causes the process to be triggered only when an edge change of variable TEMP_Trigger from 0 to 1 occurs.
3. Select **New Tag** again.

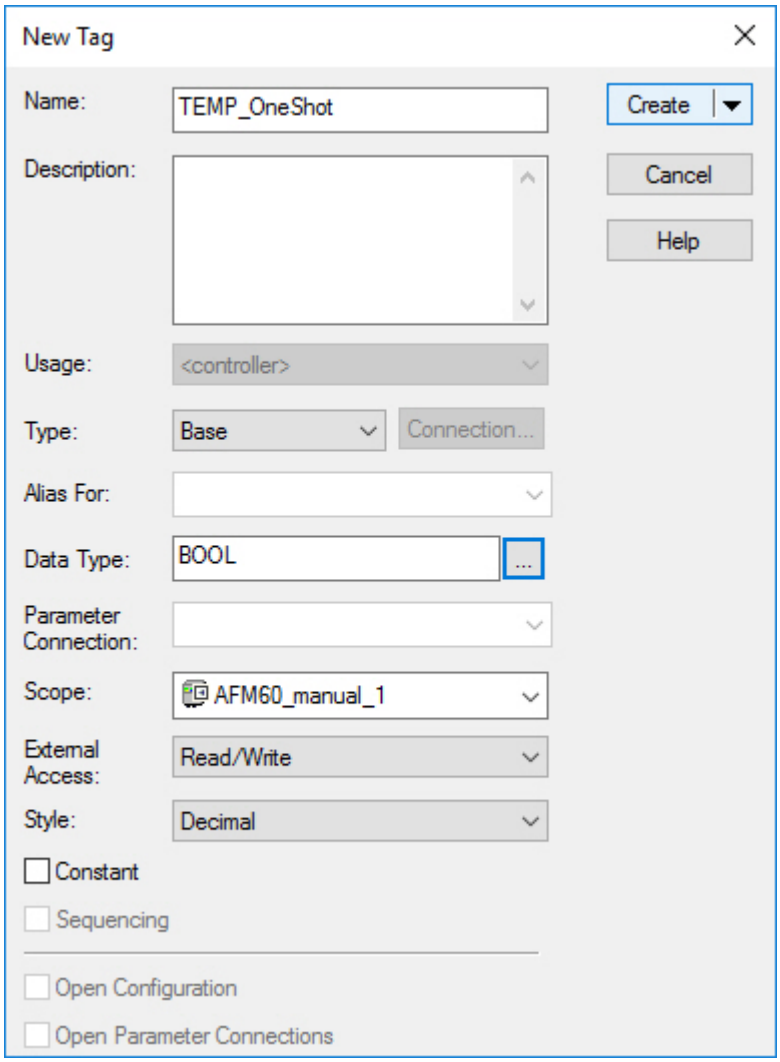


Figure 66: Definition of variable TEMP_OneShot

4. In the **New Tag** dialog, enter TEMP_OneShot in the **Name** field, select the BOOL data type in the **Data Type** field and click on **OK**.
   Another variable must be created, which will later contain the temperature value (see table 24, page 29, attribute ID 64h, temperature value).
5. Select **New Tag** again.

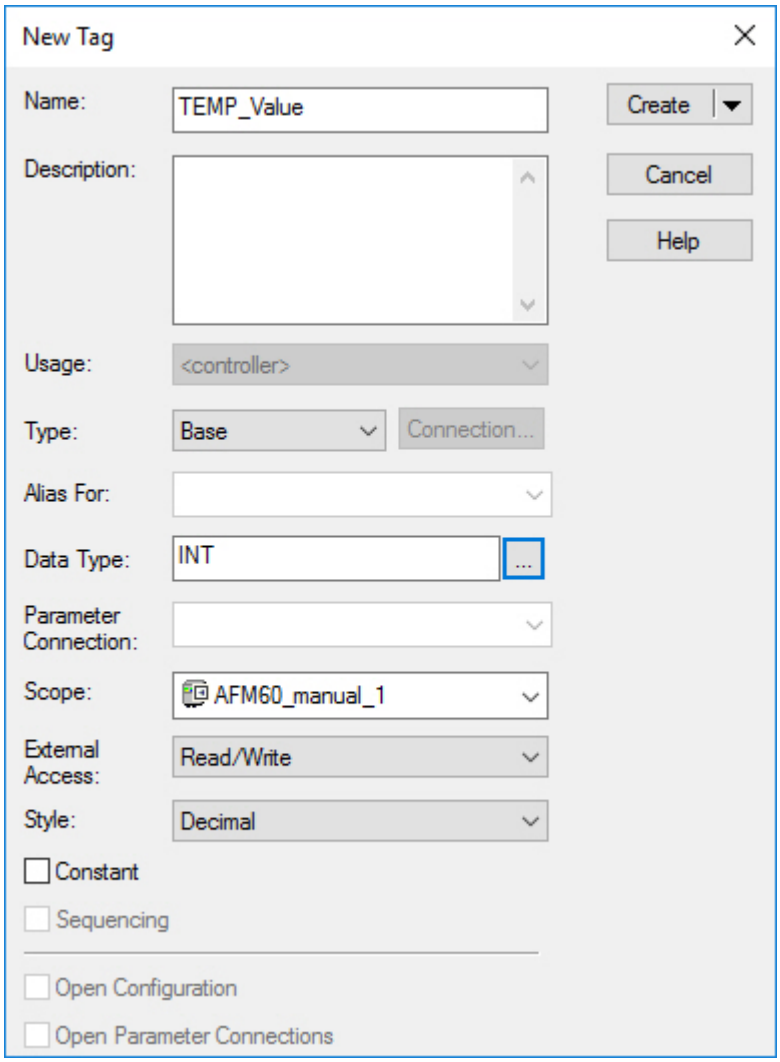


Figure 67: Definition of variable TEMP_Value

6. In the **New Tag** dialog, enter TEMP_Value in the **Name** field, select the INT data type in the **Data Type** field and click on **OK**.
   Finally, a variable must be defined and declared that obtains the temperature value from the controller.
7. Select **New Tag** again.

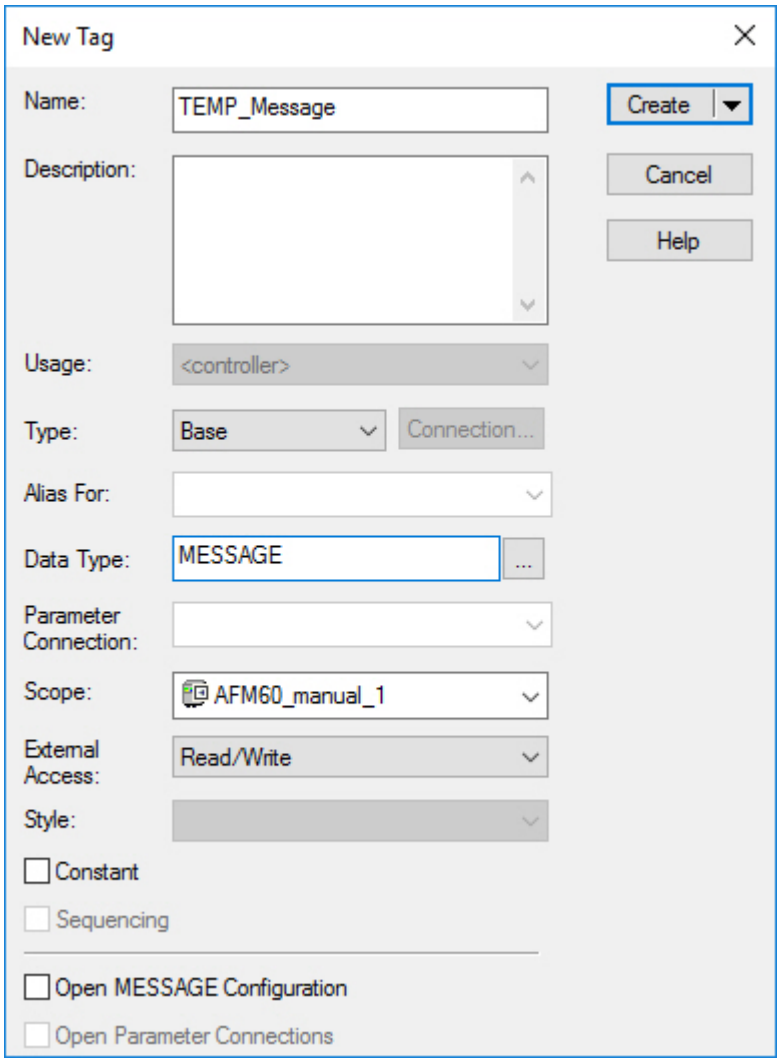
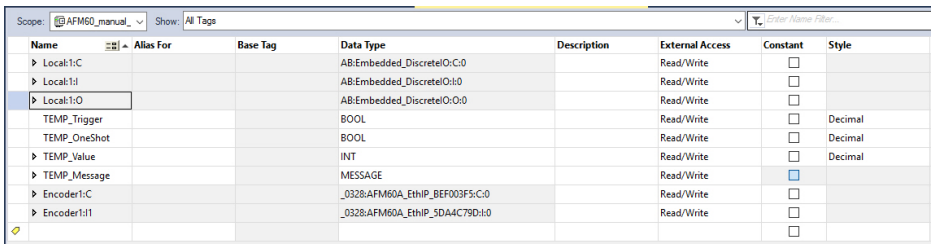


Figure 68: Definition of variable TEMP_Message

8. In the **New Tag** dialog, enter TEMP_Message in the **Name** field, select the MESSAGE data type in the **Data Type** field and click on **OK**.

✓ The following figure shows the resulting variable structure for acyclic reading of the temperature:



Name	Alias For	Base Tag	Data Type	Description	External Access	Constant	Style
Local:1:C			AB:Embedded_DiscreteIO:C:0		Read/Write	☐	
Local:1:I			AB:Embedded_DiscreteIO:I:0		Read/Write	☐	
Local:1:O			AB:Embedded_DiscreteIO:O:0		Read/Write	☐	
TEMP_Trigger			BOOL		Read/Write	☐	Decimal
TEMP_OneShot			BOOL		Read/Write	☐	Decimal
TEMP_Value			INT		Read/Write	☐	Decimal
TEMP_Message			MESSAGE		Read/Write	☐	
Encoder1:C			_0328:AFM60A_EthIP_BEF003F5:C:0		Read/Write	☐	
Encoder1:I			_0328:AFM60A_EthIP_5DA4C79D:I:0		Read/Write	☐	

Figure 69: Variable structure for reading out the temperature

Defining process flow

After the variables have been defined and declared, the program blocks must be inserted into the ladder logic and the variables assigned accordingly.

1. Open the **MainRoutine** window under **Tasks** → **Main Task** → **MainProgram** .

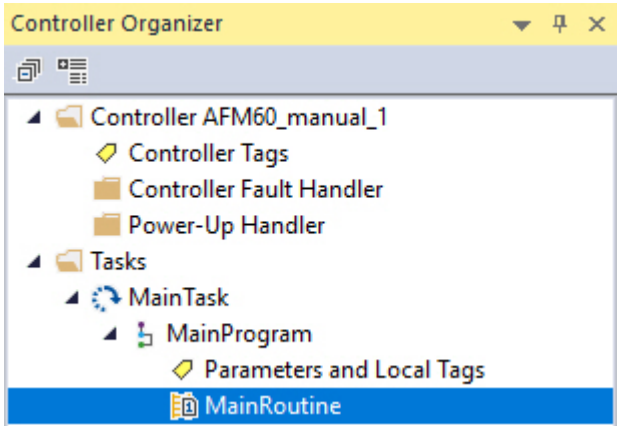


Figure 70: Opening MainRoutine

The first block to be inserted is an input that is to trigger the “Read temperature” process.

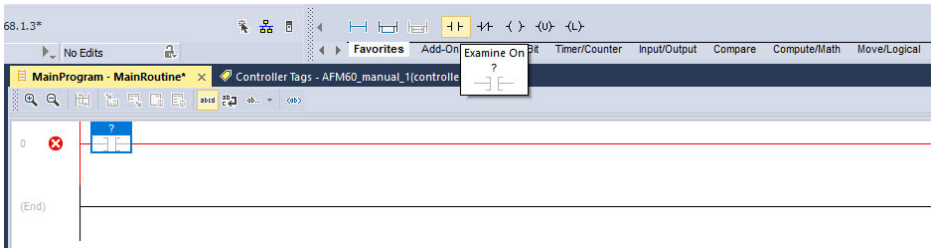


Figure 71: Inserting ExamineOn module

2. From the **Favorites** tab, select the **ExamineOn** block and insert it into the **MainRoutine** . The corresponding variable must be assigned to this input, in our example variable TEMP_Trigger.

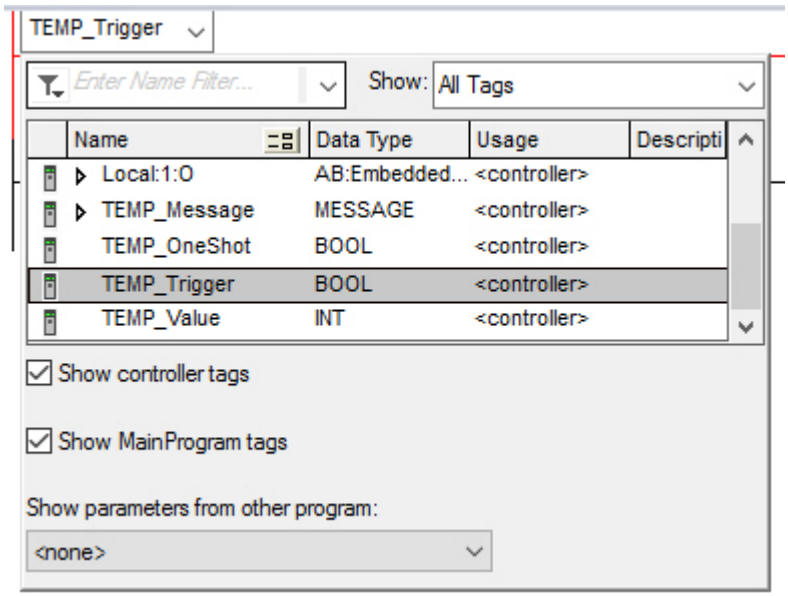


Figure 72: Assignment of variable TEMP_Trigger to ExamineOn

3. Click on the **question mark** .
✓ A drop-down menu will open.
4. Select variable TEMP_Trigger.
For the edge sensitivity of the process flow, the ONS block must be inserted.

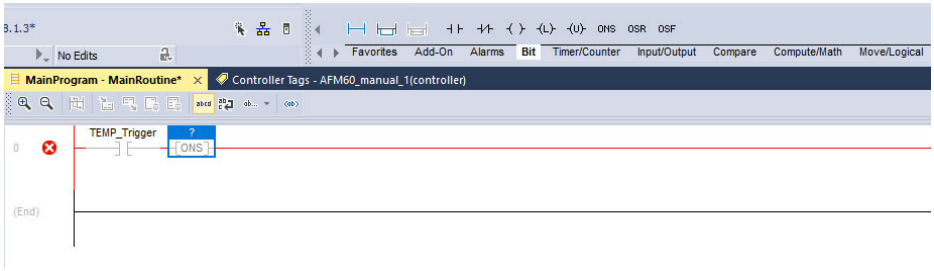


Figure 73: Inserting ONS block

5. From the **Bit** tab, select the **ONS** block and insert it into the **MainRoutine** .
   A variable must also be assigned to this block.

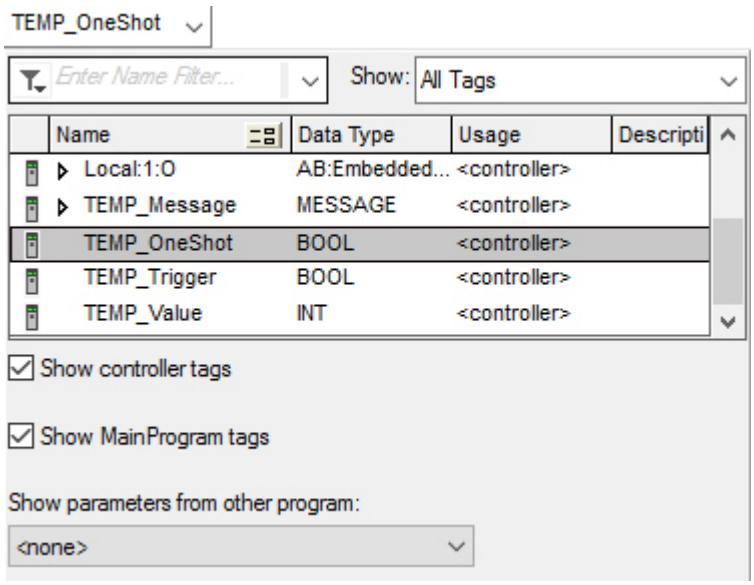


Figure 74: Assignment of variables TEMP_OneShot to ONS

6. Click on the **question mark** .

✓ A drop-down menu will open.

7. Select variable TEMP_OneShot.
   In the next step, the message must be configured to read the temperature value from the encoder.

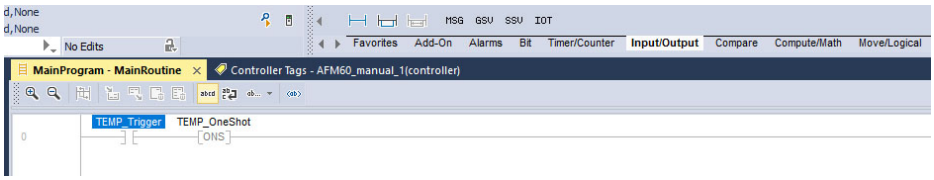


Figure 75: Inserting MSG block

8. From the **Input/Output** tab, select the **MSG** block and insert it into the **MainRoutine** .

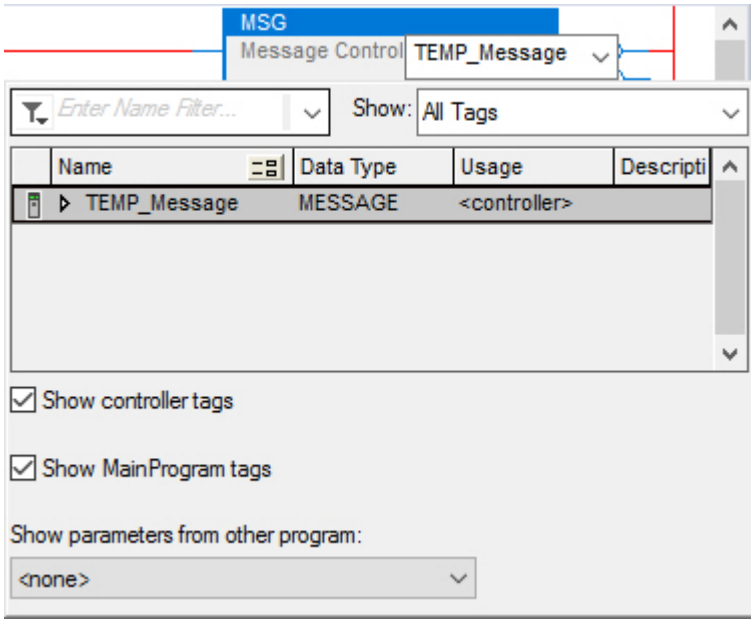


Figure 76: Assignment of variables TEMP_Message to MSG

9. In the **Message Control** field, select variable TEMP_Message.
   The MSG block must then be configured.

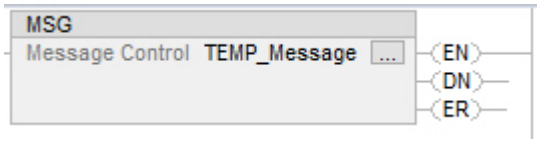


Figure 77: Opening configuration dialog of the MSG block

10. Click on the button with the three dots.

✓ The **Message Configuration** dialog opens.

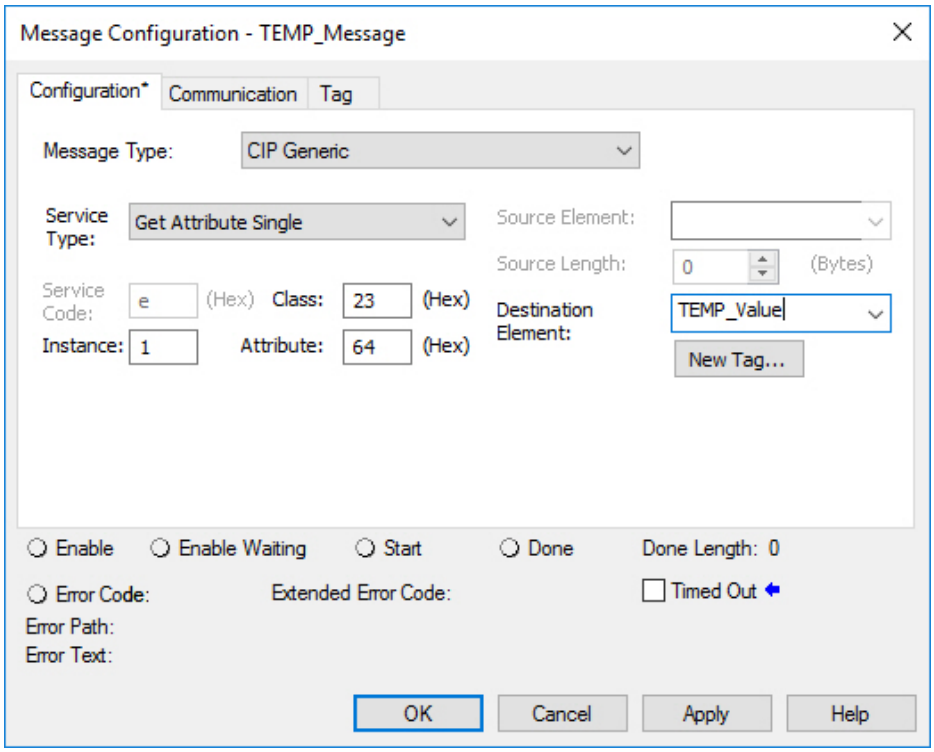


Figure 78: Configuration dialog of the MSG block

11. Configure the following parameters in the Configuration tab:
    - **Service Type:** Get Attribute Single (see table 21, page 28)
    - **Instance:** 1 (as only one device is connected to the controller)
    - **Class:** 23(h) (position sensor object, see table 8, page 19)
    - **Attribute:** 64(h) (Temperature Value, see table 24, page 29)
    - **Destination:** TEMP_Value

NOTE
TEMP_Value is the fourth variable created. The value of the temperature is written into this when the example program is executed.

12. Open the Communication tab.

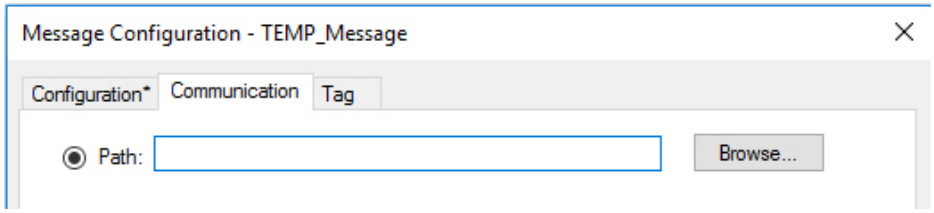


Figure 79: Communication tab

13. Next to the **Path** field, click the **Browse...** button.
✓ The **Message Path Browser** dialog opens.
14. Select the connected encoder.

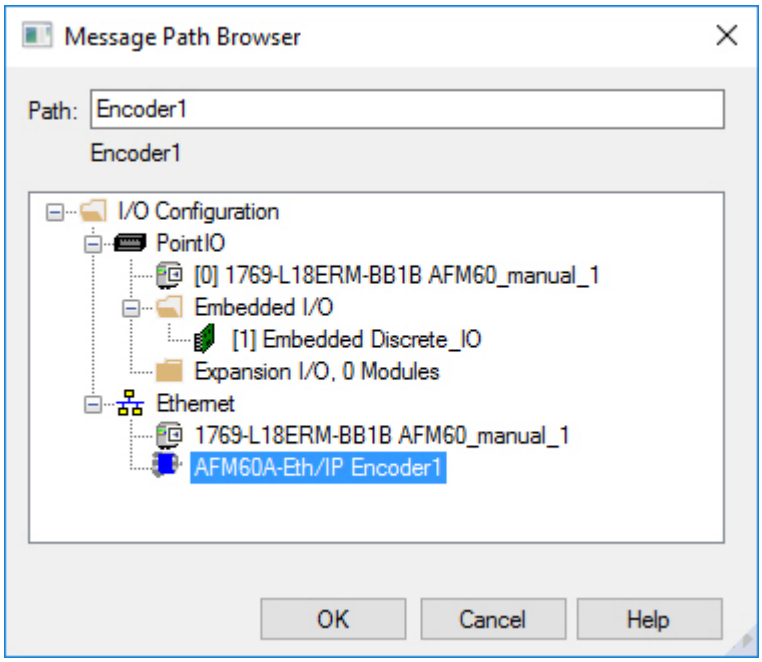


Figure 80: Selecting encoder

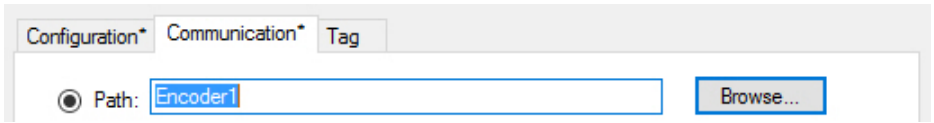


Figure 81: Selected encoder

✓ The encoder is transferred to the **Path** field.

15. End the **Message Path Browser** dialog with **OK** .

Transmitting program to controller

The program is then transmitted to the controller.

1. In the **Offline** menu, select the **Download** command.

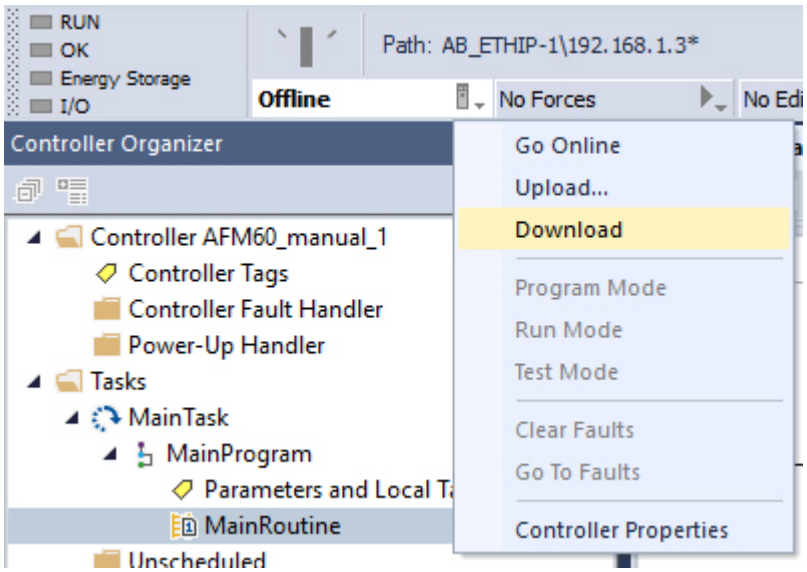


Figure 82: Transmitting the program to the controller

2. Confirm the next message.

Testing program

If, in the Controller Organizer, variable TEMP_Trigger is changed from 0 to 1, then in variable TEMP_Value, the temperature value is displayed (here: 39.00 °C).

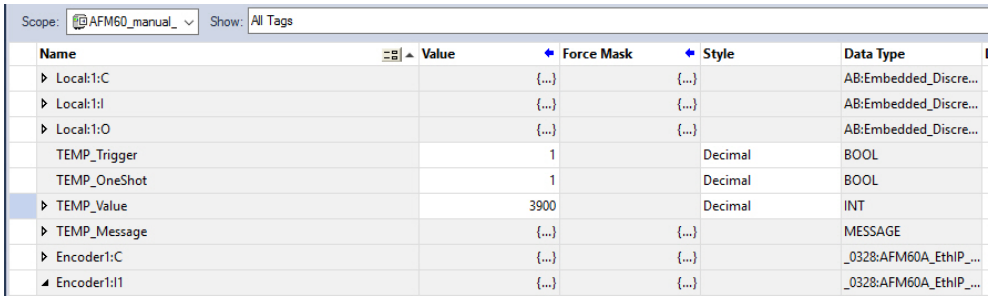


Figure 83: Display of the temperature value in TEMP_Value

5.8.2 Setting preset value

In the following example, a preset value is to be set.

Defining and declaring variables

First variables PRESET_Trigger, PRESET_OneShot, PRESET_Value and PRESET_Message must be defined and declared for the program.

First, variable PRESET_Trigger is created to trigger the process.

1. Right-click in **Controller Organizer** on **Controller Tags** and select **New Tag**.

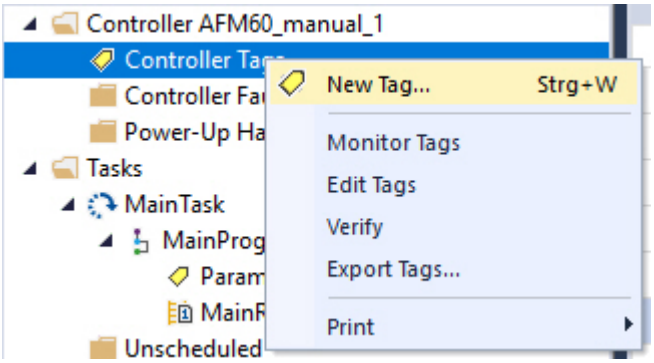


Figure 84: Create a new variable

✓ The **New Tag** dialog opens.

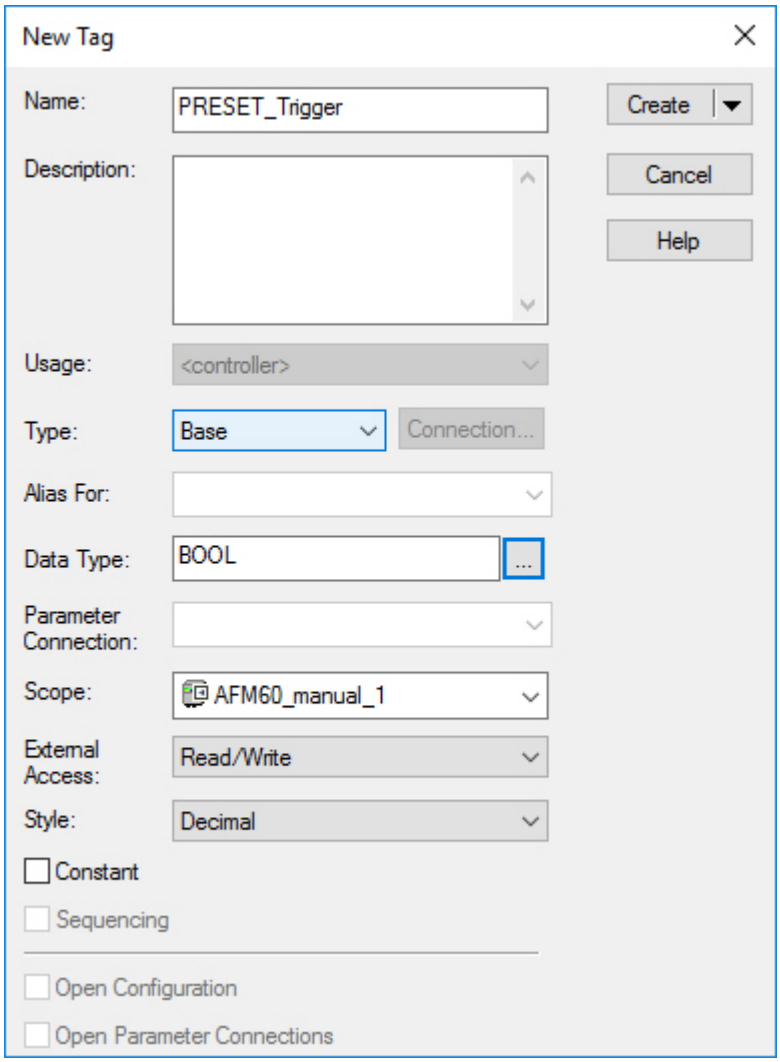


Figure 85: Definition of variable PRESET_Trigger

2. Enter PRESET_Trigger in the **Name** field, select the BOOL data type in the **Data Type** field and click on **OK**.
   To trigger the process only once, another element, in this case edge-sensitive, must be defined and declared. This causes the process to be triggered only when an edge change of variable PRESET_Trigger from 0 to 1 occurs.
3. Select **New Tag** again.

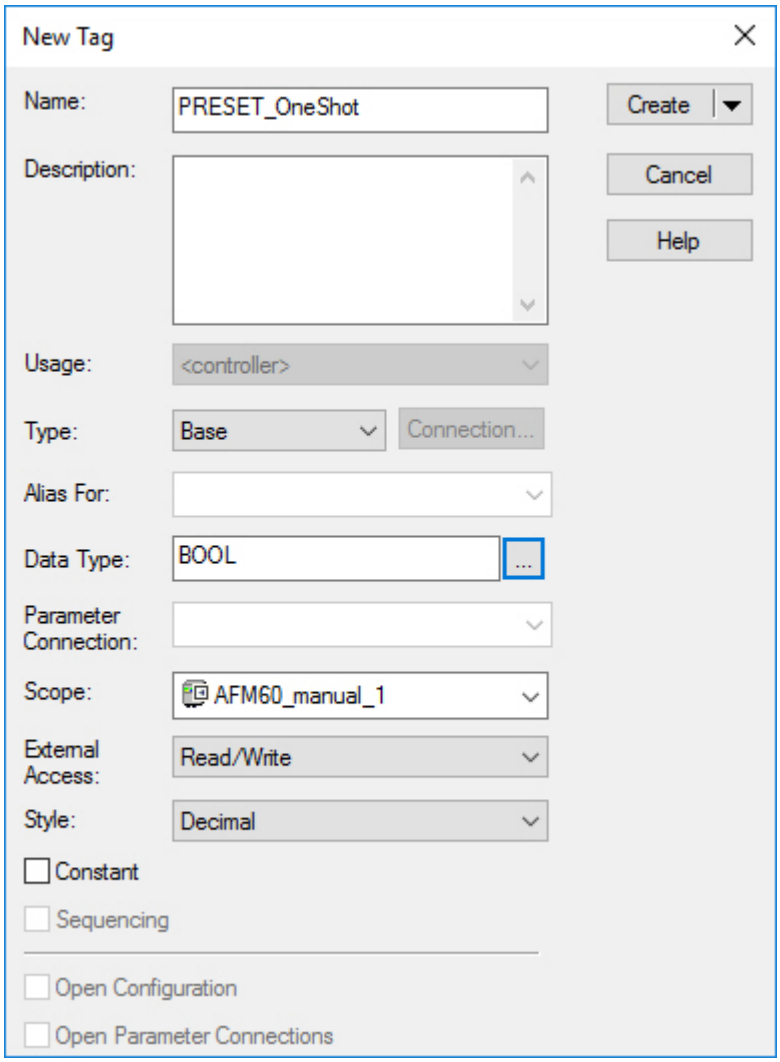


Figure 86: Definition of variable PRESET_OneShot

4. In the **New Tag** dialog, enter PRESET_OneShot in the **Name** field, select the BOOL data type in the **Data Type** field and click on **OK**.
   Another variable must be created which will later contain the preset value (see table 24, page 29, attribute ID 13h, preset value).
5. Select **New Tag** again.

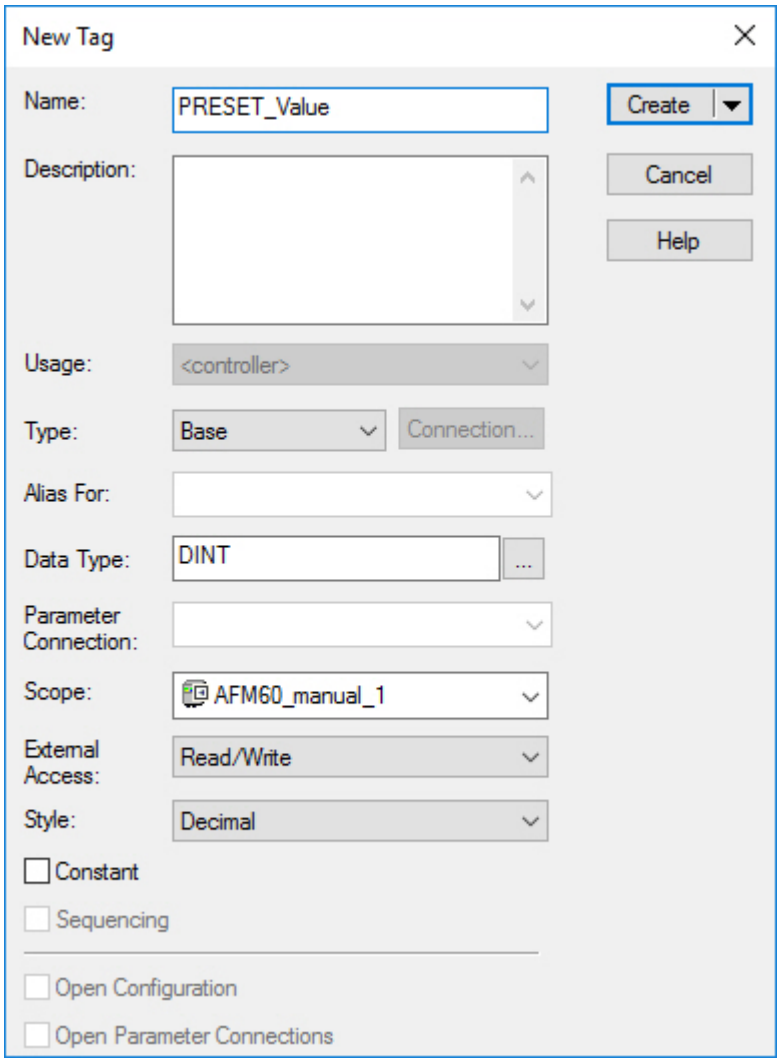


Figure 87: Definition of variable PRESET_Value

6. In the **New Tag** dialog, enter PRESET_Value in the **Name** field, select the DINT data type in the **Data Type** field and click on **OK**.
   Finally, a variable must be defined and declared that obtains the preset value from the controller.
7. Select **New Tag** again.

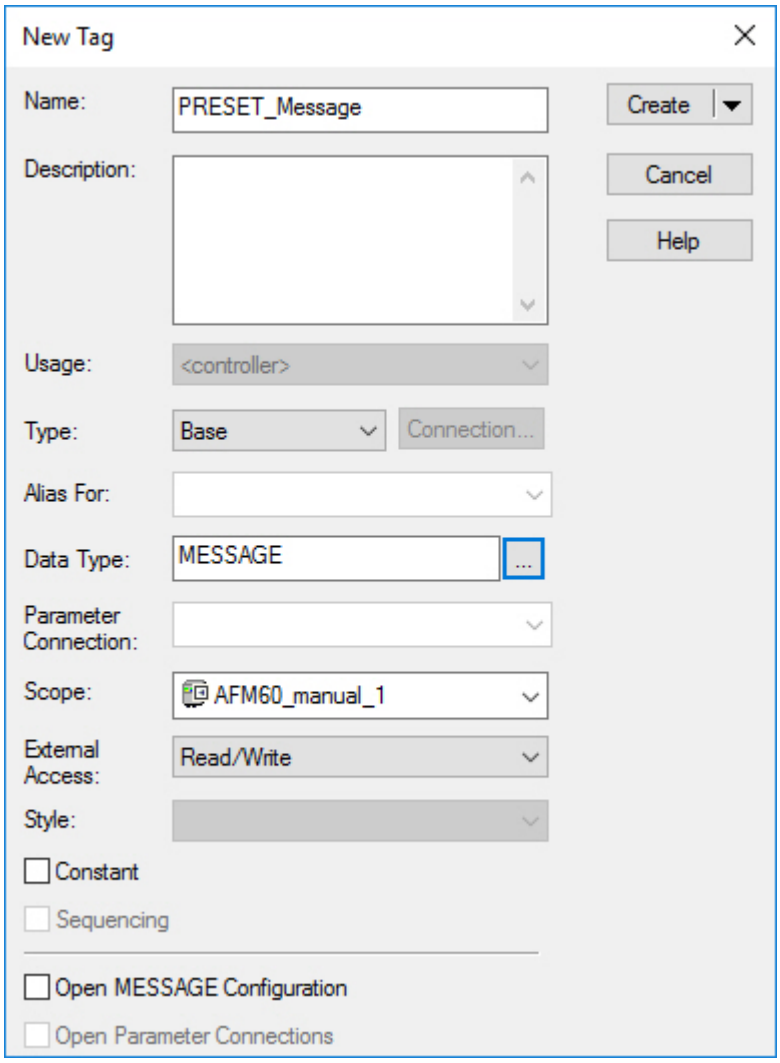


Figure 88: Definition of variable PRESET_Message

8. In the **New Tag** dialog, enter PRESET_Message in the **Name** field, select the MESSAGE data type in the **Data Type** field and click on **OK**.
   The following figure shows the resulting variable structure for setting a preset value:

Name	Alias For	Base Tag	Data Type	Description	External Access	Constant	Style
Local:1:C			AB:Embedded_DiscreteIO:C:0		Read/Write	☐	
Local:1:I			AB:Embedded_DiscreteIO:I:0		Read/Write	☐	
Local:1:O			AB:Embedded_DiscreteIO:O:0		Read/Write	☐	
Encoder1:C			_0328:AFM60A_EthIP_BEF003F5:C:0		Read/Write	☐	
Encoder1:I1			_0328:AFM60A_EthIP_5DA4C79D:I:0		Read/Write	☐	
PRESET_Trigger			BOOL		Read/Write	☐	Decimal
PRESET_OneShot			BOOL		Read/Write	☐	Decimal
PRESET_Value			DINT		Read/Write	☐	Decimal
PRESET_Message			MESSAGE		Read/Write	☐	
						☐	

Figure 89: Variable structure for setting a preset value

Defining process flow

After the variables have been defined and declared, the program blocks must be inserted into the ladder logic and the variables assigned accordingly.

1. Open the **MainRoutine** window under **Tasks** → **Main Task** → **MainProgram** .

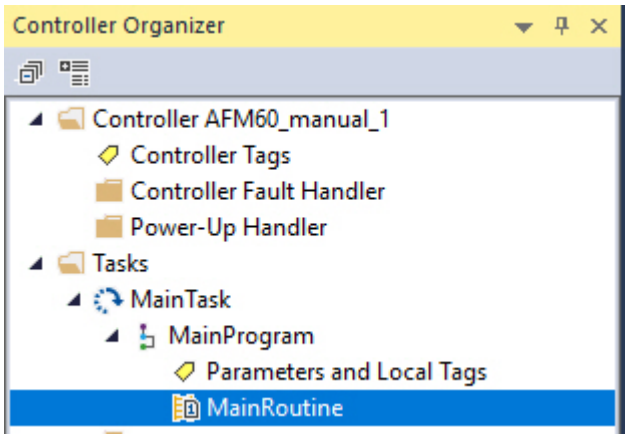


Figure 90: Opening MainRoutine

If the process flow for writing a preset value is to run parallel to the previous example, then a new string must be inserted.

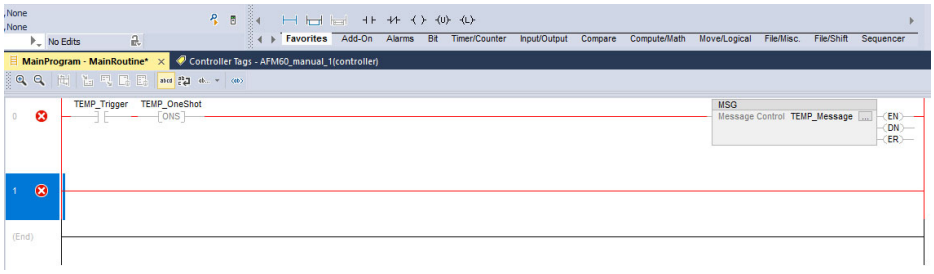


Figure 91: Inserting Rung block

2. From the **Favorites** tab, select the **Rung** block and insert it into the **MainRoutine** . The first block to be inserted is an input that is to trigger the “Set preset value" process.

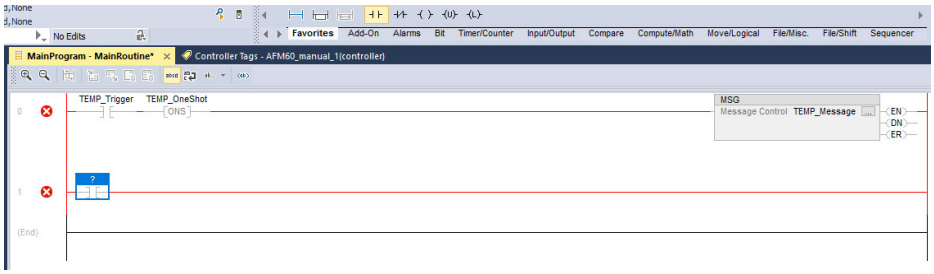


Figure 92: Inserting ExamineOn module

3. From the **Favorites** tab, select the **ExamineOn** block and insert it into the **MainRoutine** . The corresponding variable must be assigned to this input, in our example variable PRESET_Trigger.

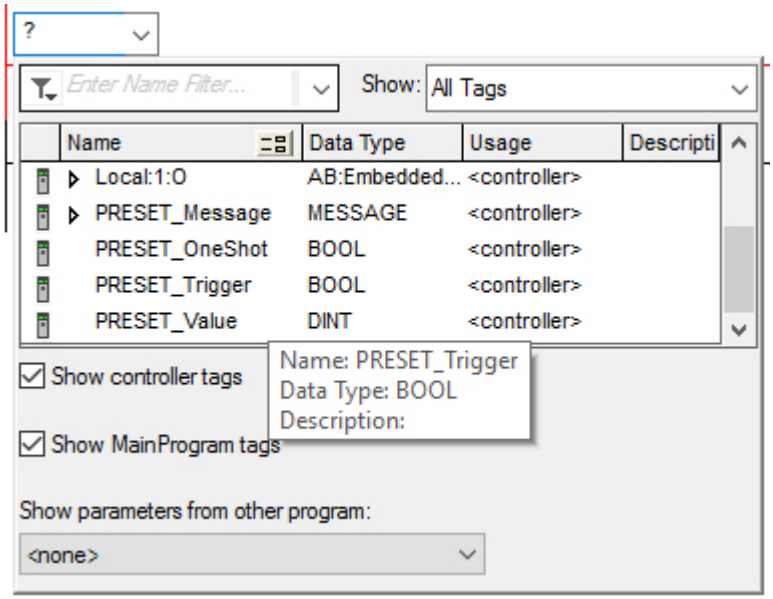


Figure 93: Assignment of variable PRESET_Trigger to ExamineOn

4. Click on the **Fragezeichen** .
✓ A drop-down menu will open.
5. Select variable PRESET_Trigger.
For the edge sensitivity of the process flow, the ONS block must be inserted.

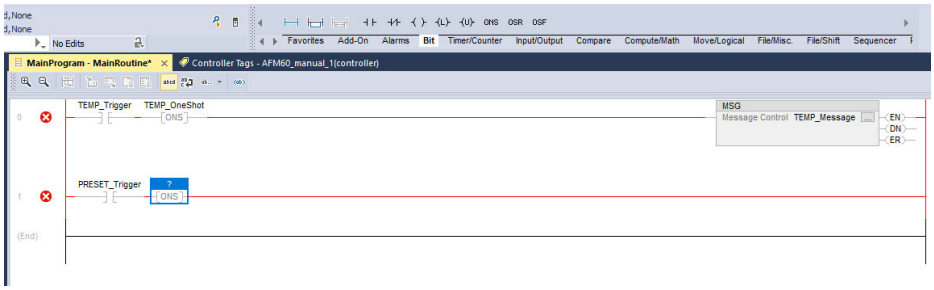


Figure 94: Inserting ONS block

6. From the **Bit** tab, select the **ONS** block and insert it into the **MainRoutine** .
A variable must also be assigned to this block.

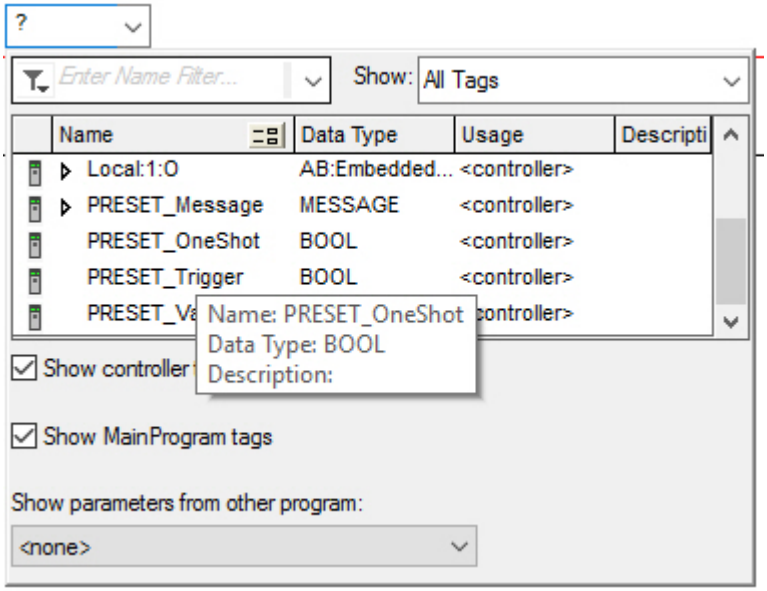


Figure 95: Assignment of variables PRESET_OneShot to ONS

7. Click on the **Fragezeichen** .

✓ A drop-down menu will open.

8. Select variable PRESET_OneShot.
   In the next step, the message must be configured to write the preset value into the encoder.

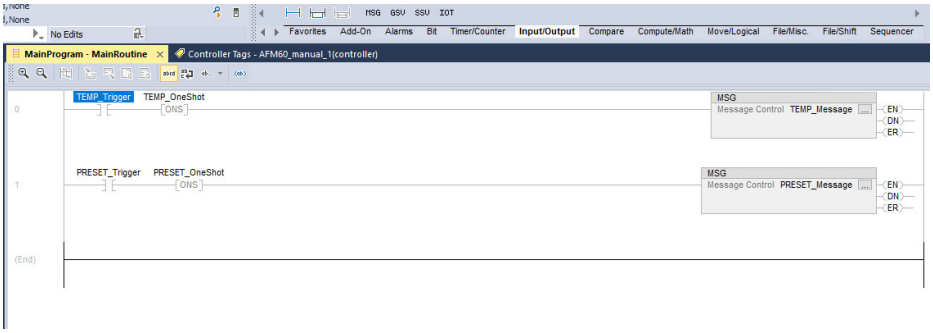


Figure 96: Inserting MSG block

9. From the **Input/Output** tab, select the **MSG** block and insert it into the **MainRoutine** .

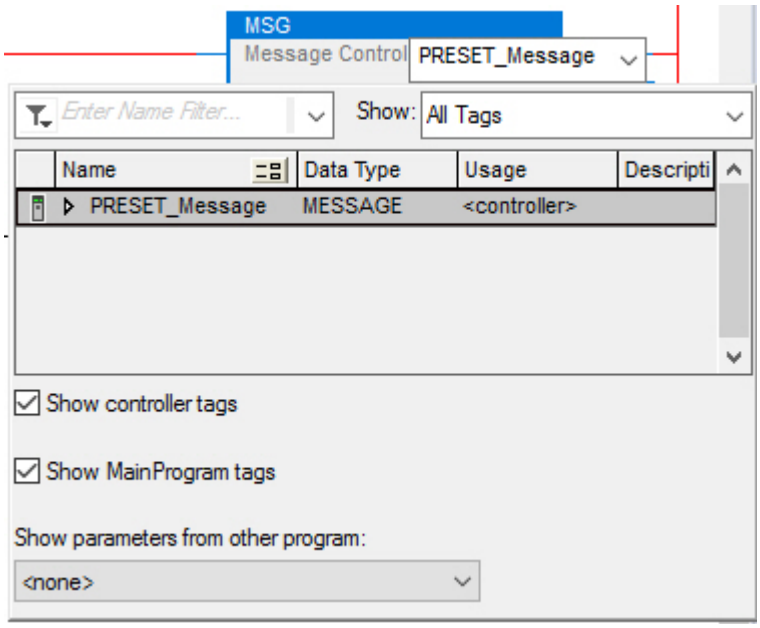


Figure 97: Assignment of variables PRESET_Message to MSG

10. In the **Message Control** field, select variable PRESET_Message.
The MSG block must then be configured.

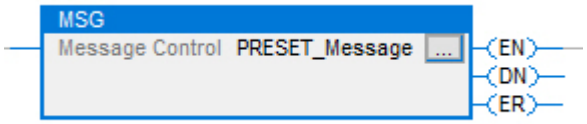


Figure 98: Opening configuration dialog of the MSG block

11. Click on the button with the three dots.

✓ The Message Configuration dialog opens.

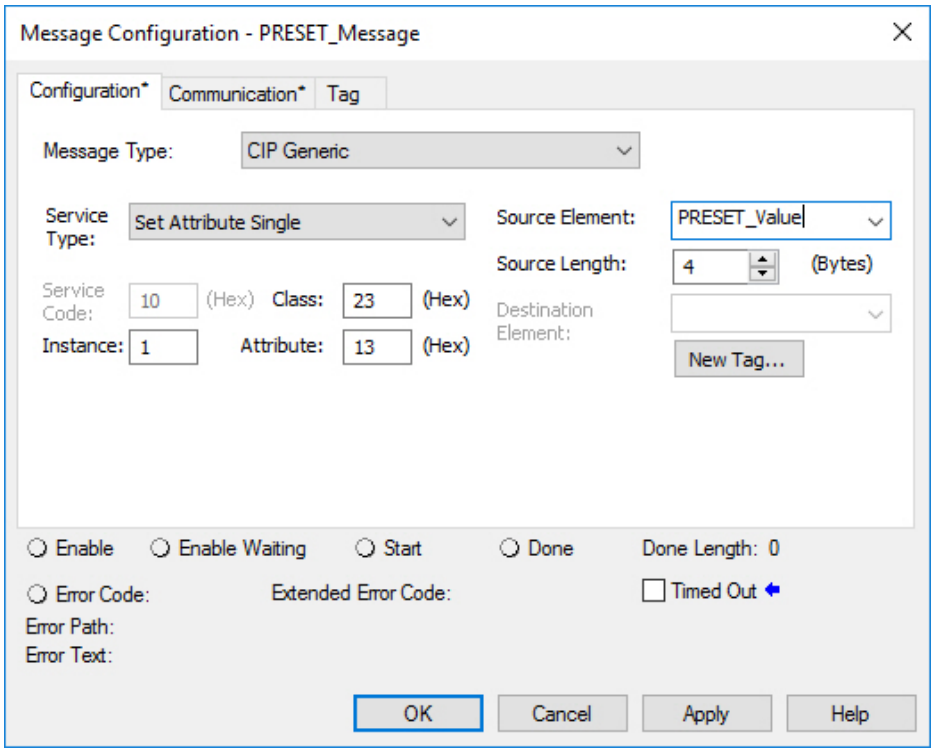


Figure 99: Configuration dialog of the MSG block

12. Configure the following parameters in the **Configuration** tab:
    - **Service Type:** Set Attributes Single (see table 21, page 28)
    - **Instance:** 1 (as only one device is connected to the controller)
    - **Class:** 23(h) (position sensor object, see table 8, page 19)
    - **Attribute:** 13(h) (Preset Value, see table 24, page 29)
    - **Source Element:** PRESET_Value
    - **Source Length:** 4

NOTE

PRESET_Value is the fourth variable created. The preset value is taken from this when the example program is executed and written to attribute 13h of the position sensor object.

13. Open the Communication tab.

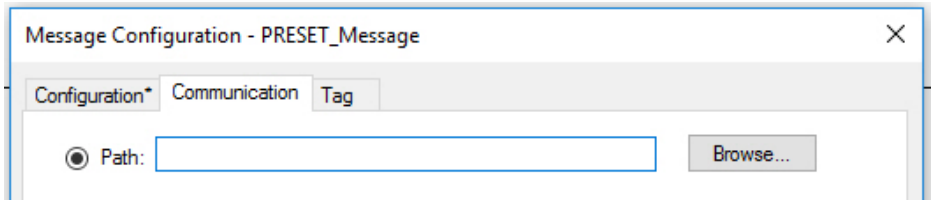


Figure 100: Communication tab

14. Next to the **Path** field, click the **Browse...** button.

✓ The **Message Path Browser** dialog opens.

15. Select the connected encoder.

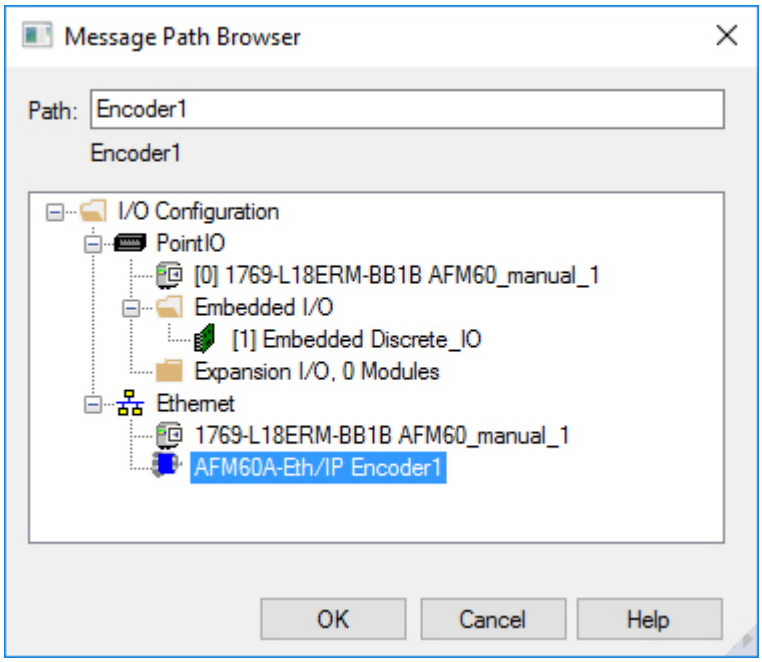


Figure 101: Selecting encoder

Figure 102: Selected encoder

✓ The encoder is transferred to the **Path** field.

16. End the **Message Path Browser** dialog with **OK** .

Transmitting program to controller

The program is then transmitted to the controller.

1. In the **Offline** menu, select the **Download** command.

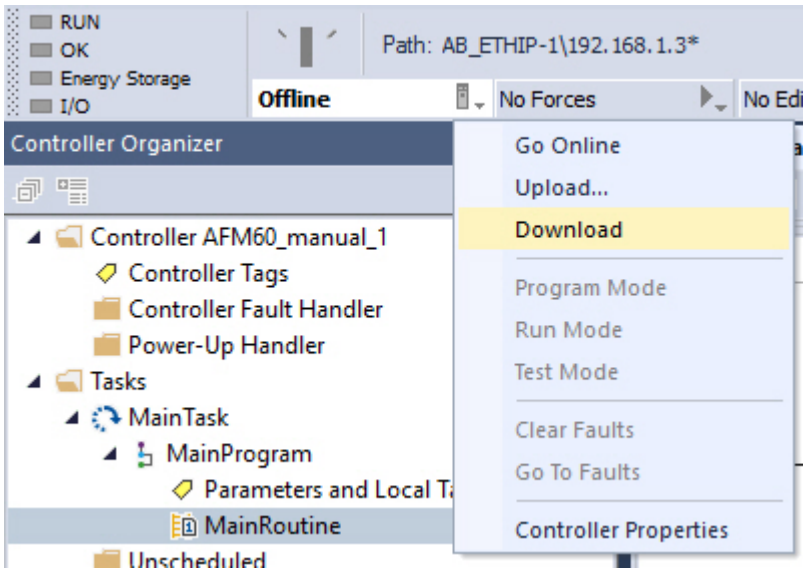


Figure 103: Transmitting the program to the controller

2. Confirm the next message.

Testing program

◢ Encoder1:I1	{...}	{...}		_0328:AFM60A_EthIP_...
Encoder1:I1.ConnectionFaulted	0		Decimal	BOOL
◢ Encoder1:I1.Data	{...}	{...}	Decimal	DINT[3]
▷ Encoder1:I1.Data[0]	0		Decimal	DINT
▷ Encoder1:I1.Data[1]	500		Decimal	DINT
▷ Encoder1:I1.Data[2]	0		Decimal	DINT
PRESET_Trigger	1		Decimal	BOOL
PRESET_OneShot	1		Decimal	BOOL
▷ PRESET_Value	500		Decimal	DINT
▷ PRESET_Message	{...}	{...}		MESSAGE

Figure 104: Display of the preset value in PRESET_Value

1. To test the example program, in variables **PRESET_Value**, enter a value in **Controller Organizer** (500 in the example).
2. Set variable **PRESET_Trigger** from 0 to 1.

✓ In position date **AFM60_EIP:I.Data[1]**, the value now jumps to 500.

6 Configuration using the integrated web server

A web server is integrated into the absolute encoder. With this web server, the state of the encoder can be observed, and the encoder can be parameterized and diagnosed.

NOTE

If parameters are changed with the web server, observe the corresponding notes (see "Integration and configuration options", page 34).

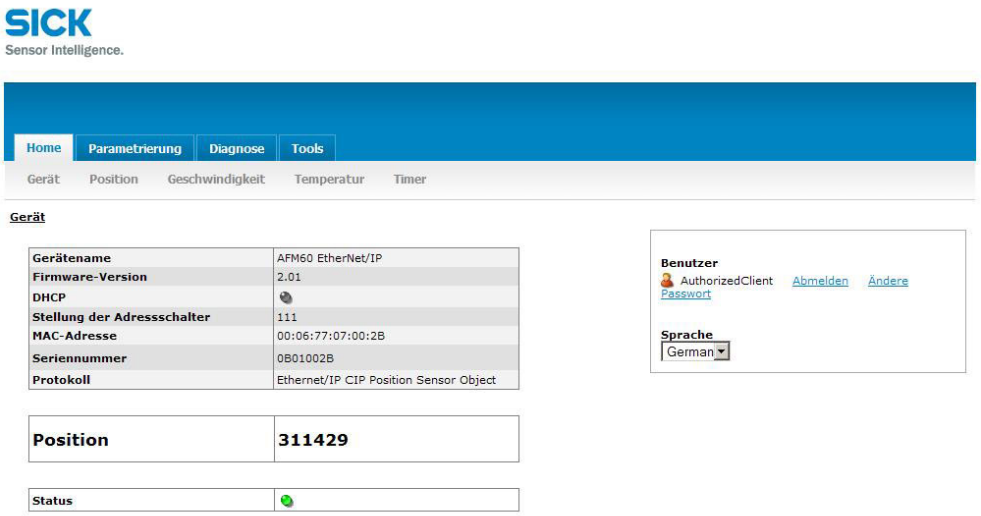


Figure 105: Web server interface

Requirements

- The encoder must be connected.
- The encoder must communicate with a browser-enabled device.
- The web server supports Internet Explorer V8.0 64 bit and higher, Google Chrome V38.0 and higher, Firefox V33.0.2 and higher.
- The IP address of the encoder must be known (see "IP address of the encoder", page 46).

Language

The web server starts in English.

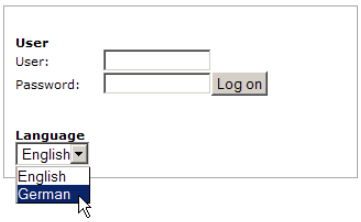


Figure 106: Select language

In the **Language** selection field, the language of the interface can be changed to German (Deutsch).

6.1 Home

NOTE
All displayed values are updated about once a second.

6.1.1 Device

This page lists the basic data about the encoder.

In addition, an LED symbol indicates the following status:

●	**Green**	Encoder is in Operational status (ready for operation, no alarms, warnings or errors occurred).
☀	**Green**	Incorrect scaling parameters present.
●	**Red**	The alarm flag is set.
☀	**Red**	The warning flag is set.

A detailed description of the alarms, warnings or errors that have occurred can be found on the web server **Diagnose** page (see "Diagnostics", page 101).

6.1.2 Position

This page shows the following parameters from the position sensor object (see table 24, page 29):

- Current position value (attribute ID 0Ah)
- Lower limit of the position (attribute ID 16h)
- Upper limit of the position (attribute ID 17h)

The limit values can be changed via the “AuthorizedClient” user (see "Limits", page 100).

6.1.3 Speed

This page shows the following parameters from the position sensor object (see table 24, page 29):

- Current speed (attribute ID 18h)
  The unit of speed is defined by attributes 19h and 20h.
- Lower speed limit (attribute ID 1Bh)
- Upper speed limit (attribute ID 1Ch)

The limit values can be changed via the “AuthorizedClient” user (see "Limits", page 100).

6.1.4 Temperature

This page shows the following parameters from the position sensor object (see table 24, page 29):

- Current temperature (attribute ID 64h)
  The temperature is displayed with ± 5 ° accuracy.
- Lower limit of the temperature (attribute ID 67h)
- Upper limit of the temperature (attribute ID 68h)

The limit values can be changed via the “AuthorizedClient” user (see "Limits", page 100).

6.1.5 Timer

This page shows the following parameters from the position sensor object (see table 24, page 29):

- Stored movement time in seconds (attribute ID 6Bh)
- Stored operating time in seconds (attribute ID 6Ch)

The limit values can be changed via the “AuthorizedClient” user (see "Limits", page 100).

6.2 Parameterization

The encoder can be parameterized with the help of this page. The parameterization sets the attributes of the position sensor object (see table 24, page 29). The parameterization options depend on who has logged in as a user.

After a new parameter has been entered, press the [Enter] key. The parameter is written into the volatile memory of the encoder.

NOTE
Only the last changed parameter is written into the volatile memory by pressing the [Enter] key. If several values are to be changed (e.g. the lower and upper speed limits), press the [Enter] key after each entry.

The following parameterization options are available without logging in:

- Overview
- Units
- Preset

The following parameterization options are available after logging in as the “AuthorizedClient” user:

- Scaling
- Round axis functionality
- Changing preset value
- Limits
- Reset

Log in

For parameterization, the following access data can be used for login:

- User: AuthorizedClient
- Password: enc123

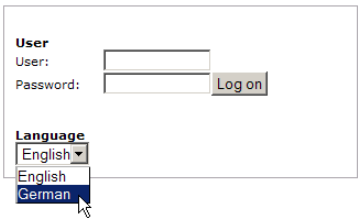


Figure 107: Log in

Changing the password

NOTE
For data security reasons and to avoid unauthorized access, changing the password at the first login is recommended.

1. Go to **Benutzer** and click on the **Ändere Passwort** link.

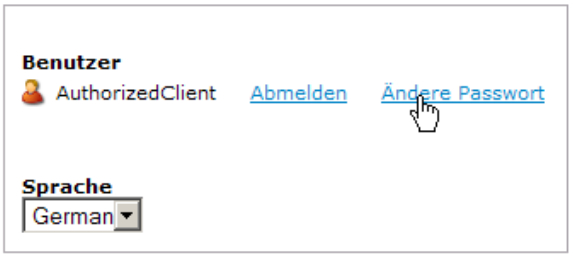


Figure 108: Changing the password

✓ The **Ändere Passwort** dialog opens.

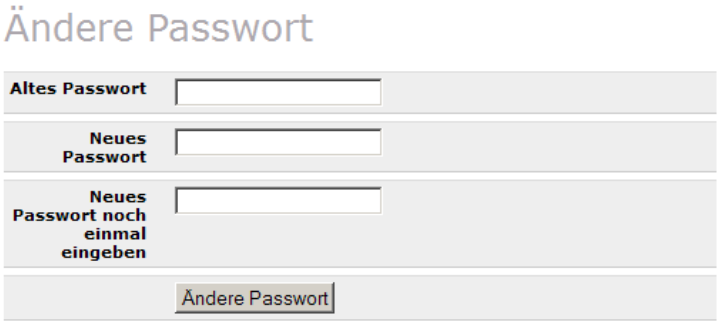


Figure 109: Dialog for changing the password

2. Enter the previously used password in the **Altes Passwort** field.
3. Enter a new password in the **Neues Passwort** field.
   Enter at least 1 character or a maximum of 16 characters (all Unicode characters are allowed).
4. In the **Neues Passwort noch einmal eingeben** field, enter the new password again.
5. Click **Ändere Passwort** .

✓ The new password will be applied.

NOTE
The password is transmitted unencrypted on the network for technical reasons. Appropriate measures must be taken to prevent the password from being read.

6.2.1 Overview

This page shows an excerpt of the attributes of the position sensor object (see table 24, page 29).

- The **Aktuell** column shows the currently configured parameters.
- The **Default** column shows the factory settings.
- The **ID hex** column shows the attribute IDs of the position sensor object.

6.2.2 Units

On this page, the units for direction, speed, acceleration and temperature can be parameterized from the position sensor object (see table 24, page 29).

- Code sequence (attribute ID 0Ch)
  - Clockwise
  - Counterclockwise
- Speed unit (attribute ID 19h)
  - counts/s
  - counts/ms

- turns/s
- turns/min
- turns/h

- Acceleration unit (attribute ID 1Eh)
  - counts/ms^2
  - counts/s^2
  - turns/s^2
  - rad/s^2
- Temperature unit (attribute ID 65h)
  - °C (Celsius)
  - °F (Fahrenheit)

6.2.3 Changing preset value

On this page, the preset value of the position sensor object can be parameterized (attribute ID 13h, see figure 112, page 98).

DANGER
Before changing the preset value, check whether there is any danger from the machine or system in which the encoder is integrated!

As soon as the value has been entered and the entry has been confirmed with the [Enter] key, the value is accepted as the position value.

The Preset function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Preset function should only be used when the encoder is at a standstill.

6.2.4 Triggering preset

This page shows the current position value of the encoder and the preset value (attribute ID 13h) from the position sensor object.

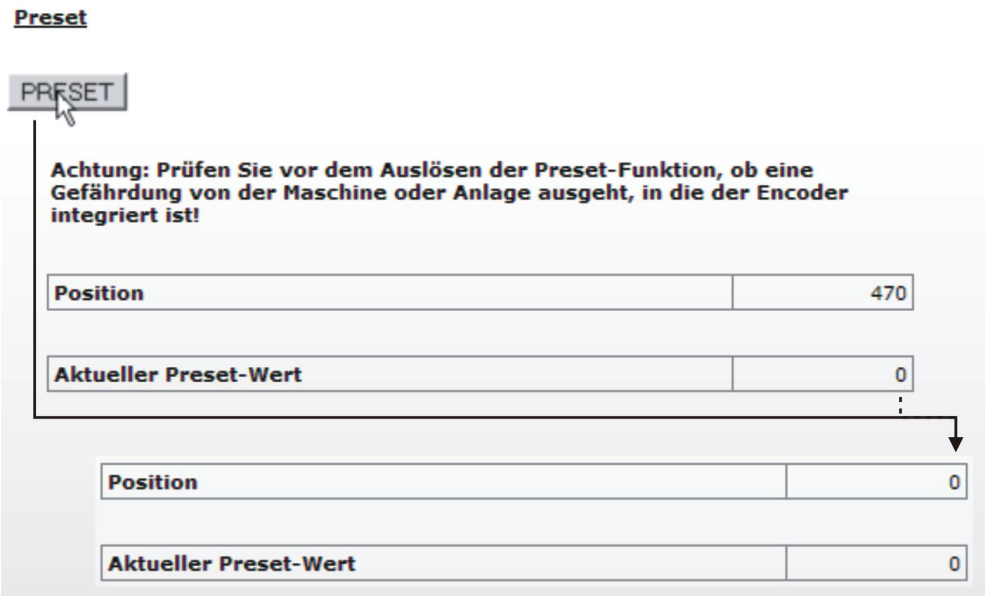


Figure 110: Triggering preset

1. Click **PRESET** .

✓ The position value is set to the preset value.

The preset value can be changed via the “AuthorizedClient” user (see figure 109, page 96).

6.2.5 Scaling

On this page, you can configure the scaling parameters of the position sensor object (see table 24, page 29).

- **Skalierung** (attribute ID 0Eh)
  - on
  - off

If the scaling is set to **on**, then the following parameters are displayed:

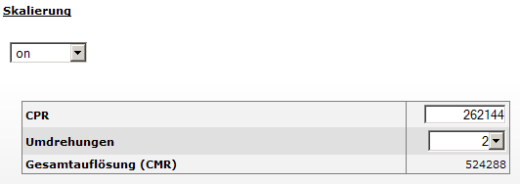


Figure 111: Figure 110: Scaling

- **CPR**, number of steps per revolution (attribute ID 10h)
- **Umdrehungen**, number of revolutions of the total resolution (This is not an attribute of the position sensor object).
  Only the following values can be selected: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1,024, 2,048 and 4,096.
- The **Gesamtauflösung (CMR)** field shows the value of attribute ID 11h “Total Measuring Range, Total Resolution” of the position sensor object (see table 24, page 29).

NOTE
If round axis functionality is activated, then no scaling can be set.

DANGER
Before using the Scaling function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Scaling function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Scaling function (steps per revolution or total resolution) should only be used when the encoder is at a standstill.

6.2.6 Round axis functionality

If round axis functionality is activated, then the (corresponding) numerators, denominators and the total resolution can be configured (see table 24, page 29).

- Round axis functionality (attribute ID 7Dh)
  - on
  - off

If round axis functionality is set to **on** , then the following parameters are displayed:

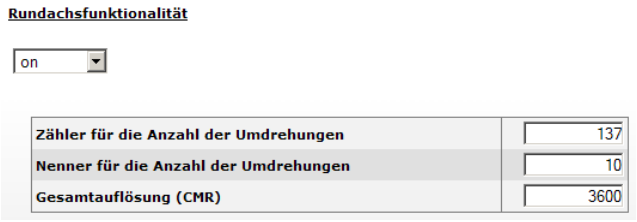


Figure 112: Round axis functionality

- **Zähler für die Anzahl der Umdrehungen** (attribute ID 7Eh)
- **Nenner für die Anzahl der Umdrehungen** (attribute ID 7Fh)
- **Gesamtauflösung** (CMR) (attribute ID 11h)

The requirements and restrictions for the parameters are described in chapter 3.7.10.

NOTE
If round axis functionality is activated, then the scaling is set to **on** on the Scaling page. However, no scaling parameters are offered.

DANGER
Before using the Round axis functionality function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Round axis functionality function can lead to an immediate change of the position value output by the encoder.

This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Round axis functionality function should only be used when the encoder is at a standstill.

6.2.7 Limits

On this page, the position, speed, acceleration and temperature limits can be parameterized:

- Lower limit of the position (attribute ID 16h)
- Upper limit of the position (attribute ID 17h)

NOTE
Area monitoring can be implemented by specifying an upper and lower limit of the position. It is not an electronic cam.

- Lower speed limit (attribute ID 1Bh)
- Upper speed limit (attribute ID 1Ch)
- Lower limit of acceleration (attribute ID 20h)
- Upper limit of acceleration (attribute ID 21h)

If these limits are exceeded, then the consequence is:

- The warning flag (attribute ID 31h) of the position sensor object is set (see table 24, page 29).
- On the **Gerät** page, the status LED flashes (see "Device", page 95).
- On the **Status** page, the warning text is displayed (see "Status", page 101).

In addition, other limits that are not included in the position sensor object can be set:

- Limit of movement time in hours [1)]
- Operating time limit in hours [1)]
- Limit of the number of changes in the direction of rotation
- Limit of the number of clockwise starts
- Limit of the number of counterclockwise starts

1) The movement time and the operating time are always calculated from the first commissioning of the encoder. When configuring the limit, note that the encoder may already have some movement time and operating time.

6.2.8 Reset

On this page, various class services of the position sensor object can be executed (see table 21, page 28).

Saving parameters to non-volatile memory

▶ Click –**S**– .
The function uses the **Save** class service (service code 16h) of the position sensor object.
The parameters are saved to the non-volatile memory, the encoder is restarted.

Resetting to factory settings

DANGER
Before using the Reset function, check whether there is any danger from the machine or system in which the encoder is integrated!

The Reset function leads to a reset of the parameters of the position sensor object to the factory settings, which can lead to an immediate change of the position value output by the encoder. This could cause an unexpected movement that could endanger people or damage the system or other objects.

NOTE
The Reset function should only be used when the encoder is at a standstill.

▶ Click –**D**– .
The function uses the **Reset** class service (service code 05h) of the position sensor object (Data = 01h).
The parameters are reset to the factory settings and the encoder is restarted.

Restarting

- Click –**R**– .
The encoder is restarted.

NOTE
After the restart, the language is reset to English and the user is logged out.

6.3 Diagnostics

The diagnostic pages display detailed information on possible alarms, warnings and errors.

6.3.1 Status

The page shows a description of the error when a warning or alarm occurred.

Figure 113: Diagnostic status

- **Aktueller Status**
  The last three messages since switch-on are displayed (after switching off and switching back on, the memory is empty).
- **Statusspeicher**
  The texts for warnings, alarms and errors from the Fault header are displayed (see table 33, page 108). If no warning, no alarm and no error has occurred yet, the displayed text is **Keine Einträge**.
- **Bewegungszeit des Encoders**
  Indicates whether the movement time is within the tolerated values (see "Limits", page 100).
- **Betriebszeit des Encoders**
  Indicates whether the operating time is within the tolerated values (see "Limits", page 100).

6.3.2 Speed

This page shows the following values for the speed from the position sensor object (see table 24, page 29):

- Speed unit (attribute ID 19h)
- Current speed (attribute ID 18h)
- Highest speed that the encoder has reached since commissioning (attribute ID 6Dh)
- Lower speed limit (attribute ID 1Bh)
- Upper speed limit (attribute ID 1Ch)

6.3.3 Temperature

This page shows the following values for the temperature from the position sensor object (see table 24, page 29):

- Temperature unit (attribute ID 65h)
- Current temperature (attribute ID 64h)
- Highest operating temperature reached (attribute ID 6Fh)
- Lowest operating temperature reached (attribute ID 70h)
- Lower limit of the temperature (attribute ID 67h)
- Upper limit of the temperature (attribute ID 68h)

6.3.4 Time

This page shows the following values for the movement and operating times of the encoder from the position sensor object (see table 24, page 29):

- Stored movement time in seconds (attribute ID 6Bh)
- Limit of movement time in hours (see "Limits", page 100)
- Stored operating time in seconds (attribute ID 6Ch)
- Limit of operating time in hours (see "Limits", page 100)

6.3.5 Cycles

This page shows the following values for the cycles of the encoder from the position sensor object (see table 24, page 29):

- Number of changes in the direction of rotation (attribute ID 75h)
- Number of clockwise starts (attribute ID 76h)
- Number of counterclockwise starts (attribute ID 77h)
- Limit of the number of changes of the direction of rotation (see "Limits", page 100)
- Limit of the number of clockwise starts (see "Limits", page 100)
- Limit of the number of counterclockwise starts (see "Limits", page 100)

6.3.6 Heartbeat

The absolute encoder supports the slave sign-of-life functionality (see "Slave sign of life", page 38).

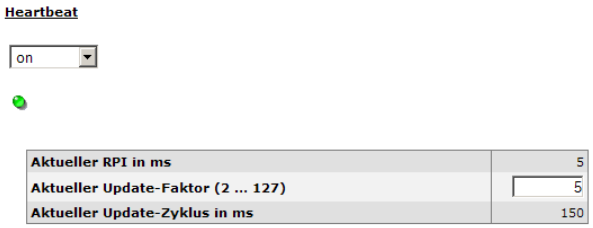


Figure 114: Heartbeat

If the heartbeat is set to **on** , then the following symbols and parameters are displayed:

An LED symbol indicates the heartbeat:

●	**Green**	Active
●	**Gray**	Not active

NOTE
Since the website is updated every second, the change between statuses cannot be displayed in real time.

The **Aktueller RPI in ms** column shows the RPI.

The update factor can be specified in the **Aktueller Update-Faktor (2 ... 127)** field.

The **Aktueller Update-Zyklus in ms** column shows the heartbeat.

6.4 Tools

6.4.1 EDS

The EDS files for integrating the encoder into the PLC are stored in the encoder.

▶ Click **Download EDS** to download the files as a RAR archive.
The RAR archive contains the EDS files for the singleturn and multiturn encoders and their icons.

6.4.2 Ladder routine

The ladder routine is used to map the configuration data between the controller and the web server (see "Configuration", page 34). The ladder routine is stored in the encoder.

Depending on whether the 101WS or 103WS instance or the 102WS instance of the assembly object is used (see table 18, page 23), the appropriate ladder routine must be downloaded.

▶ Select the ladder routine that matches the instance you are using. Click **Download Ladder-Routine ...** to download the file as a RAR archive.

6.4.3 Update

A firmware update via FTP can be performed as follows.

1. If there is a connection to the encoder's web server, then the web browser can be closed.
2. Start an FTP client and enter the IP address of the encoder.
3. Use the following login data:
   - User name = host
   - Password = enc123

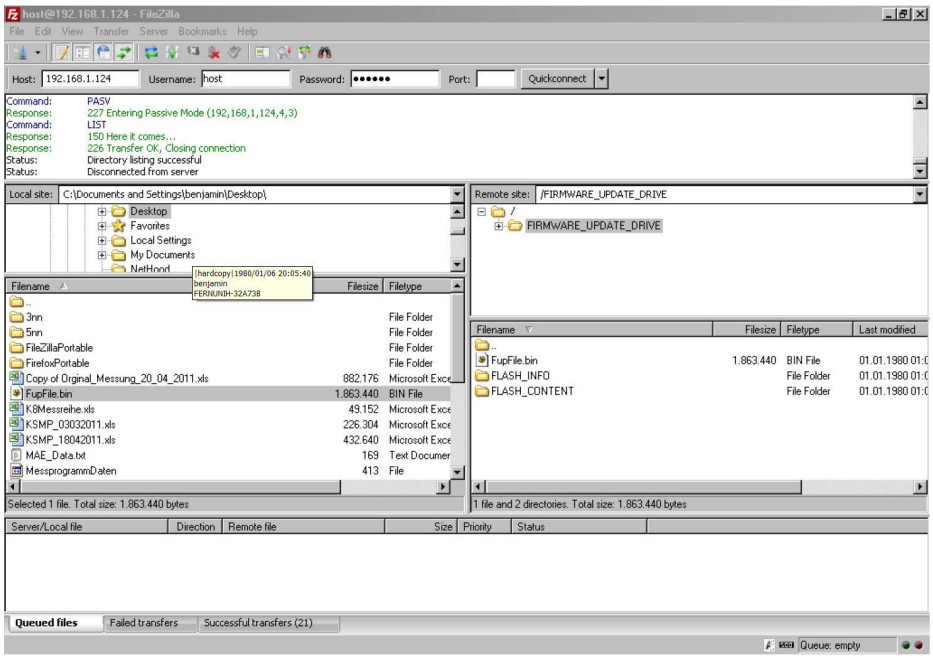


Figure 115: Example for the firmware update

4. Open the FIRMWARE_UPDATE_DRIVE folder.
5. Transmit the update file (*.bin) [1] to this folder.

The firmware update takes about 3 minutes.

- During the firmware update, the Encoder LED initially flashes red.
- The Encoder LED then lights up red.

Following the firmware update, the encoder performs a reboot.

- The Encoder LED then lights up green again.

NOTE

It must be ensured that the encoder is permanently supplied with voltage during the firmware update. In the event of a voltage interruption, the encoder is either reset to the status before the update or, in the worst case, is no longer responsive.

1) The file (*.bin) required for a firmware update can currently be requested from Sick Technical Support if required.

6.4.4 Address switch

This page shows the setting options of the address switches (see table 30, page 45).

6.4.5 Fault header information

The encoder has a Fault header in which alarms and warnings that have occurred are stored. The possible alarms and warnings are listed on the Fault header information page.

6.5 Test notes

DANGER

No commissioning without inspection by authorized personnel!

Before a system equipped with AFS60/AFM60 EtherNet/IP is put into operation for the first time, it must be checked and released by authorized personnel. Observe the following notes in chapter 2: see "Safety information", page 9.

7 Troubleshooting

7.1 Response to errors

DANGER
Cease operation if the cause of the malfunction has not been clearly identified!
The machine must be put out of operation if the error cannot be clearly assigned and safely rectified.

7.2 SICK-STEGMANN Support

If a fault cannot be rectified with the help of the information in this chapter, then contact the responsible SICK-STEGMANN subsidiary.

7.3 Diagnostics

7.3.1 Error and status indications of the LEDs

Mod, Net and Encoder status LEDs

The Mod LED shows the device status, the Net LED the status of the CIP connection and the Encoder LED the status of the internal measuring device of the absolute encoder.

Table 31: Meaning of the Mod, Net and Encoder status LEDs

Display		Description
Mod LED		
○	Off	No supply voltage
●	Green	Device in operation
☼	Green	Standby/device not configured, no IP address assigned
☼	Red	Warning, but device still ready for operation or Firmware update in progress
●	Red	Error, device not operational
☼	Red/green	Self-test when switching on
Net LED		
○	Off	No supply voltage or No IP address
☼	Green	No connection Device has IP address but no CIP connection
●	Green	Device has IP address and a CIP connection
☼	Red	Warning, connection time out reset by performing a reset or establishing a new connection
●	Red	Error IP address already assigned to other device
☼	Red/green	Self-test when switching on
Encoder LED		

Display		Description
○	Off	No supply voltage or No IP address
☼	Green	Warning Incorrect parameter
●	Green	Device in operation
☼	Red	Warning, but device still ready for operation or Firmware update in progress
●	Red	Error Encoder error or Restart after firmware update in progress
☼	Red/green	Self-test when switching on

Link 1 and 2 Ethernet link LEDs

The Link 1 and 2 Ethernet link LEDs indicate the physical connection status of the Ethernet interface.

Table 32: Meaning of the Link 1 and 2 LEDs

Display		Description
○	Off	No supply voltage or No Ethernet connection
●	Green	Ethernet connection established
●	Yellow	Interface port locked
☼	Green	Data transmission TxD/RxD
☼	Yellow	Data collisions

7.3.2 Self test via EtherNet/IP

A self-test is available to check the sensor system and the most important functions of the encoder.

NOTE
The self-test may only be performed when the encoder is at standstill.

The self-test can be triggered via the diagnostic bit of attribute ID 0Dh in the position sensor object (see table 24, page 29). If an error occurs, bit 27 of the Fault header is set (see table 33, page 108).

Following the self-test, the diagnostic bit of attribute 13 is automatically reset to 0.

7.3.3 Warnings, alarms and errors via EtherNet/IP

Within EtherNet/IP, warnings, alarms and errors can be retrieved via implicit messages as well as via explicit messages.

If connections are established via the I/O assembly, the Fault header can be read out via instances 101, 102 and 103 as well as instances 101WS, 102WS and 103WS (see table 18, page 23).

Using the position sensor object (see table 24, page 29), alarms and warnings of the encoder can be read out with the help of the attributes.

The following applies for errors, alarms and warnings:
Bit state = 0: No error, alarm or warning
Bit state = 1: Error, alarm or warning occurred

Fault header

Table 33: Fault header

Byte	Bit	Description	Alarm (A) / Warning (W)
0	0	Operating temperature of the microcontroller outside the permissible range	W
	1	Operating temperature of the encoder outside the permissible range	W
	2	Permissible internal LED current in the sensor system exceeded	W
	3	Supply voltage outside the permissible range	W
	4	Frequency error, maximum speed is exceeded	W
	5	The lower/upper limit of the speed configured with the attribute IDs 1Bh or 1Ch has been undercut/exceeded (see table 24, page 29).	W
	6	The lower/upper limit of the acceleration configured with the attribute IDs 20h or 21h has been undercut/exceeded (see table 24, page 29).	W
	7	The lower/upper limit of the position configured with the attribute IDs 16h or 17h has been undercut/exceeded (see table 24, page 29).	W
1	8	Position error (amplitude error of singleturn measurement)	A
	9	Position error (amplitude error of multiturn measurement)	A
	10	Position error (vector error $Sin^2 + Cos^2$ of singleturn measurement)	A
	11	Position error (vector error $Sin^2 + Cos^2$ of multiturn measurement)	A
	12 ... 14	Reserved	-
	15	The “start bit” has not yet been reset or a parameter has been changed via the web server. Bit 15 is set after each restart. It can be reset using the following command: Attribute ID 8Bh (see table 24, page 29).	W
2	16	Singleturn position error (error in sensor)	A
	17	Multiturn position error (synchronization MA single)	A
	18	Multiturn position error (synchronization quad single)	A
	19	Multiturn position error (internal interface)	A
	20	Multiturn position error (FRAM)	-
	21	Limit of the number of changes of the direction of rotation exceeded	W
	22	Limit of the number of clockwise starts exceeded	W
	23	Limit of the number of counterclockwise starts exceeded	W

Byte	Bit	Description	Alarm (A) / Warning (W)
3	24	Memory error (EEPROM checksum)	A
	25	Memory error (EEPROM IRQ)	A
	26	Error during commissioning (start-up)	A
	27	Error during self test	A
	28	Limit of the movement time of the encoder is exceeded	W
	29	So-called “sanity check flag”. The flag is set when the encoder has detected an incorrect speed or a position error. Is reset when the device is switched on again.	-
	30	Slave sign of life. Active if attribute ID 0Dh is set (see table 24, page 29). The bit changes its value in the configured update cycle.	-
	31	Limit of the operating time of the encoder is exceeded	W

Alarms

If, for example, the internal self-test determines that the position value was calculated incorrectly or an incorrect configuration value was transmitted to the encoder, then the alarm flag is set (attribute 46, see table 24, page 29).

DANGER
Alarms in the application must be evaluated!

In the event of a serious error, a correct position value may not be output. This could cause an unexpected movement that could endanger people or damage the system or other objects.

In addition, the Encoder LED permanently lights up red.

In attributes 44 and 45, the type of alarms is masked in a bit field.

Table 34: Alarms

Bit	Description
0	Position error
1	Error during self test
2 ... 11	Reserved
12	Checksum incorrect (manufacturer-specific)
4	Error at system start-up (manufacturer-specific)
14 ... 15	Reserved

Warnings

For example, if the limits for speed or temperature are undercut/exceeded, the warning flag is set (attribute ID 31h, see table 24, page 29).

In addition, the Encoder LED flashes red.

In attribute IDs 2Fh and 30h, the type of warnings is masked in a bit field.

NOTE
The position value continues to be calculated correctly, so the encoder is still ready for operation.

Table 35: Warnings

Bit	Description
0	Maximum speed exceeded.
1	Permissible internal LED current in the sensor system exceeded.
2 ... 5	Not supported.
6	The lower limit of the speed configured with attribute 1Bh has been undercut.
7	The upper limit of the speed configured with attribute 1Ch has been exceeded.
8	The lower limit of the acceleration configured with attribute 20h has been undercut.
9	The upper limit of the acceleration configured with attribute 21h has been exceeded.
10	The lower/upper limit of the position configured with attribute 16h or 17h has been undercut/exceeded.
11 ... 12	Reserved
13	The lower/upper limit of the temperature configured with attribute 67h or 68h has been undercut/exceeded.
14	The minimum/maximum supply voltage has been undercut/exceeded.

7.3.4 Error messages of the Allen Bradley control system

If the encoder is integrated in an Allen Bradley control system, certain error messages may occur whose message text cannot be clearly assigned.

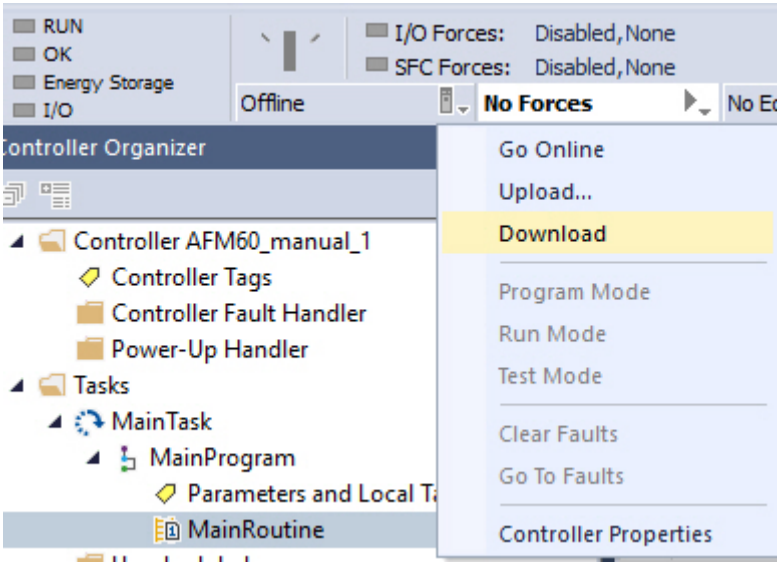


Figure 116: Example of an error message in RSLogix

The following error messages come from the RSLogix 5000 software.

Table 36: Error messages from the RSLogix 5000 software

Error code	Message	Possible cause
16#0108	Connection Request Error Connection Type (Multicast/Unicast) not supported.	▶ Check whether the configuration assembly (instance 100 of the assembly object) is activated. If so, check whether the configuration data in it is configured correctly and completely (see figure 60, page 70).

Error code	Message	Possible cause
16#0114	Electronic Keying Mismatched: Electronic keying product code and/or vendor ID mismatched.	▶ Check whether the wrong EDS file may have been selected (e.g. singleturn instead of multiturn or vice versa, see "Integration and configuration using an EDS file", page 52).
16#0127	Connection Request Error: Invalid output size.	▶ Check whether the correct communication format is being used for the control. The control's default value is "Data DINT". The encoder requires the communication format: "Input Data-DINT".
16#0204	Connection Request Error: Connection timed out.	▶ Check the supply voltage at the encoder. ▶ Check the Ethernet lines of the encoder for interruption. ▶ Check whether the IP address of the encoder matches the IP address stored in the control. Possible causes: o The address switches are not correctly engaged (see figure 19, page 44). o The encoder has lost the IP address assigned to it after a restart (see "Freezing the assigned IP address", page 49).

8 Annex

8.1 Conformity

The AFS/AFM60 EtherNet/IP absolute encoder has been manufactured in accordance with the following guidelines:

- Machinery Directive 2006/42 / EC
- EMC Directive 2014/30 / EU
- ROHS Directive 2011/65/EU

The complete EU declaration of conformity is available on the SICK homepage on the Internet: www.sick.com

8014213/1D2H/2021-08-31/en

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 229 25 44
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com

SICK
Sensor Intelligence.