OPERATING INSTRUCTIONS

AFS60 EtherNet/IP
AFM60 EtherNet/IP

Absolute Encoder

SICK
Sensor Intelligence.

Described product

AFS60/AFM60 EtherNet/IP

Manufacturer

SICK STEGMANN GmbH
Dürrheimer Str. 36
78166 Donaueschingen

Germany

Legal information

Original document

This document is an original document of SICK STEGMANN GmbH.

Contents

1 About this document

Please read this chapter carefully before working with this documentation and the AFS60/AFM60 EtherNet/IP Absolute Encoder.

1.1 Function of this document

These operating instructions are designed to address the technical personnel of the machine manufacturer or the machine operator in regards to correct configuration, electrical installation, commissioning, operation and maintenance of the AFS60/AFM60 EtherNet/IP Absolute Encoder.

1.2 Target group

The operating instructions are addressed at the planners, developers and operators of systems in which one or more AFS60/AFM60 EtherNet/IP Absolute Encoders are to be integrated. They also address people who initialize the use of the AFS60/AFM60 EtherNet/IP or who are in charge of servicing and maintaining the device.

These instructions are written for trained persons who are responsible for the installation, mounting and operation of the AFS60/AFM60 EtherNet/IP in an industrial environment.

1.3 Information depth

These operating instructions contain information on the AFS60/AFM60 EtherNet/IP Absolute Encoder on the following subjects:

- product features
- electrical installation
- commissioning and configuration
- fault diagnosis and troubleshooting
- conformity

These operating instructions do not contain any information on the mounting of the AFS60/AFM60 EtherNet/IP. You will find this information in the mounting instructions included with the device.

They also do not contain any information on technical specifications, dimensional drawings, ordering information or accessories. You will find this information in the data sheet for the AFS60/AFM60 EtherNet/IP.

Planning and using measurement systems such as the AFS60/AFM60 EtherNet/IP also requires specific technical skills beyond the information in the operating instructions and mounting instructions. The information required to acquire these specific skills is not contained in this document.

When operating the AFS60/AFM60 EtherNet/IP, the national, local and statutory rules and regulations must be observed.

Further information

- www.odva.org

1.4 Scope

NOTE

These operating instructions apply to the AFS60/AFM60 EtherNet/IP Absolute Encoder with the following type codes:

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

1.5 Abbreviations used

CIP	Common Industrial Protocol
CMR	Counts per Measuring Range
CNR_D	Customized Number of Revolutions, Divisor = divisor of the customized number of revolutions
CNR_N	Customized Number of Revolutions, Nominator = dominator 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 provide an application-specific circuit
I/O	Input and Output Data (from the point of view of the master)
IP in EtherNet/IP	Industrial Protocol
IP in TCP/IP	Internet 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

1.6 Symbols used

NOTE

Refer to notes for special features of the device.

●, ☀, ○ LED symbols describe the state of a diagnostics LED. Examples:

● The LED is illuminated constantly.

☀ The LED is flashing.

○ The LED is off.

▶ **Take action ...** Instructions for taking action are shown by an arrow. Read carefully and follow the instructions for action.

⚠ **WARNING**
Warning!

A warning indicates an actual or potential risk or health hazard. They are designed to help you to prevent accidents.

Read carefully and follow the warning notices.

2 On safety

This chapter deals with your own safety and the safety of the equipment operators.

▶ Please read this chapter carefully before working with the AFS60/AFM60 EtherNet/IP or with the machine or system in which the AFS60/AFM60 EtherNet/IP is used.

2.1 Authorised personnel

The AFS60/AFM60 EtherNet/IP Absolute Encoder must only be installed, commissioned and serviced by authorized personnel.

NOTE

Repairs to the AFS60/AFM60 EtherNet/IP are only allowed to be undertaken by trained and authorized service personnel from SICK STEGMANN GmbH.

The following qualifications are necessary for the various tasks:

Activity	Qualification
Mounting	• Basic technical training • Knowledge of the current safety regulations in the workplace
Electrical installation and replacement	• Practical electrical training • Knowledge of current electrical safety regulations • Knowledge on the use and operation of devices in the related application (e.g. industrial robots, storage and conveyor technology)
Commissioning, operation and configuration	• Knowledge on the current safety regulations and the use and operation of devices in the related application • Knowledge of automation systems (e.g. Rockwell ControlLogix Controller) • Knowledge of EtherNet/IP • Knowledge of the usage of automation software (e.g. Rockwell RSLogix)

Table 1: Authorised personnel

2.2 Correct use

The AFS60/AFM60 EtherNet/IP Absolute Encoder is a measuring device that is manufactured in accordance with recognized industrial regulations and meets the quality requirements as per ISO 9001:2008 as well as those of an environment management system as per ISO 14001:2009.

An encoder is a device for mounting that cannot be used independent of its foreseen function. For this reason an encoder is not equipped with immediate safe devices.

Considerations for the safety of personnel and systems must be provided by the constructor of the system as per statutory regulations.

Due to its design, the AFS60/AFM60 EtherNet/IP can only be operated within an EtherNet/IP network. It is necessary to comply with the EtherNet/IP specifications and guidelines for setting up an EtherNet/IP network.

In case of any other usage or modifications to the AFS60/AFM60 EtherNet/IP, e.g. opening the housing during mounting and electrical installation, or in case of modifications to the SICK software, any claims against SICK STEGMANN GmbH under warranty will be rendered void.

2.3 General safety notes and protective measures

⚠ **WARNING**
Please observe the following procedures in order to ensure the correct and safe use of the AFS60/AFM60 EtherNet/IP!

The encoder is to be installed and maintained by trained and qualified personnel with knowledge of electronics, precision mechanics and control system programming. It is necessary to comply with the related standards covering the technical safety stipulations.

The safety regulations are to be met by all persons who are installing, operating or maintaining the devices:

- The operating instructions must always be available and must always be followed.
- Unqualified personnel are not allowed to be present in the vicinity of the system during installation and maintenance.
- The system is to be installed in accordance with all applicable safety regulations and the mounting instructions.
- All work safety regulations of the applicable countries are to be followed during installation.
- Failure to follow all applicable health and safety regulations may result in injury or damage to the system.
- The current and voltage sources in the encoder are designed in accordance with all applicable technical regulations.

2.4 Environmental protection

Please note the following information on disposal.

Assembly	Material	Disposal
Packaging	Cardboard	Waste paper
Shaft	Stainless steel	Scrap metal
Flange	Aluminium	Scrap metal
Housing	Aluminium die cast	Scrap metal
Electronic assemblies	Various	Electronic waste

Table 2: Disposal of the assemblies

3 Product description

This chapter provides information on the special features and properties of the AFS60/AFM60 EtherNet/IP. Absolute EncoderIt describes the construction and the operating principle of the device.

- ▶ Please read this chapter before mounting, installing and commissioning the device.

NOTE

SICK uses standard IP technology in its products. The focus is on the availability of the products and services. SICK always assumes that the integrity and confidentiality of data and the rights related to the usage of the aforementioned products will be addressed by the customer. In any case suitable security measures, e.g. network separation, firewalls, anti-virus protection, patch management etc. are always to be implemented by the customer to suit the situation.

3.1 Special features

Properties	Singleturn encoder	Multiturn encoder
Absolute Encoder in 60 mm design	■	■
Robust nickel coded disk for harsh environments	■	■
High precision and reliability	■	■
Large ball bearing spacing of 30 mm	■	■
High level of resistance to vibration	■	■
Optimal rotational accuracy	■	■
Compact design	■	■
Face mount flange, servo flange and blind hollow shaft	■	■
18 bit singleturn resolution (1 to 262,144 steps)	■	■
30 bit total resolution		■
12 bit multiturn resolution (1 to 4,096 revolutions)		■
Round axis functionality		■
EtherNet/IP interface (according to IEC 61784-1)	■	■
Supports the encoder profile 22h defined in the CIP (Common Industrial Protocol)	■	■
Device Level Ring (DLR)	■	■

Table 3: Special features of the encoder variants

3.2 Operating principle of the encoder

The AFS60/AFM60 EtherNet/IP acquires the position and velocity of rotating axes and outputs the position in the form of a unique digital numeric value. Optical acquisition of the rotary position value is from an internal coded disk.

The AFS60 EtherNet/IP is a singleturn encoder

Singleturn encoders are used if the absolute position of the shaft for one revolution is required.

The AFM60 EtherNet/IP is a multiturn encoder

Multiturn encoders are used if more than one shaft revolution must be acquired absolutely.

3.2.1 Scaleable resolution

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

The steps per revolution can be scaled from 1 ... 262,144 as an integer. The total resolution of the AFM60 EtherNet/IP must be 2^n times the steps per revolution. This restriction is not relevant if the round axis functionality is activated.

3.2.2 Preset function

The position value for an encoder can be set with the aid of a preset value. I.e. the encoder can be set to any position within the measuring range. In this way, e.g., the encoder's zero position can be adjusted to the machine's zero point.

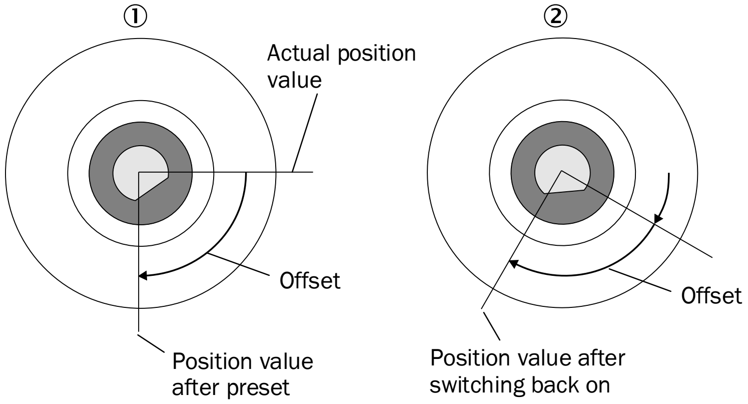

①
②
Actual position value
Offset
Position value after preset
Offset
Position value after switching back on

Figure 1: Setting a preset value

① = Setting a preset value

② = On switching back on

On switching off the encoder, the offset, the delta between the real position value and the value defined by the preset, is saved. On switching back on the new preset value is formed from the new real position value and the offset. Even if the position of encoder changes while it is switched off, this procedure ensures the correct position value is still output.

3.2.3 Round axis functionality

The encoder supports the function for round axes. During this process, the steps per revolution are set as a fraction (see section 3.6.10 on page 42). As a result, the total resolution does not have to be configured to 2^n times the steps per revolution and can also be a decimal number (e.g. 12.5).

NOTE

The output position value is adjusted with the zero point correction, the code sequence set and the gearbox parameters entered.

Example with transmission ratio

A rotary table for a filling system is to be controlled. The steps per revolution are pre-defined by the number of filling stations. There are nine filling stations. For the precise measurement of the distance between two filling stations, 1000 steps are required.

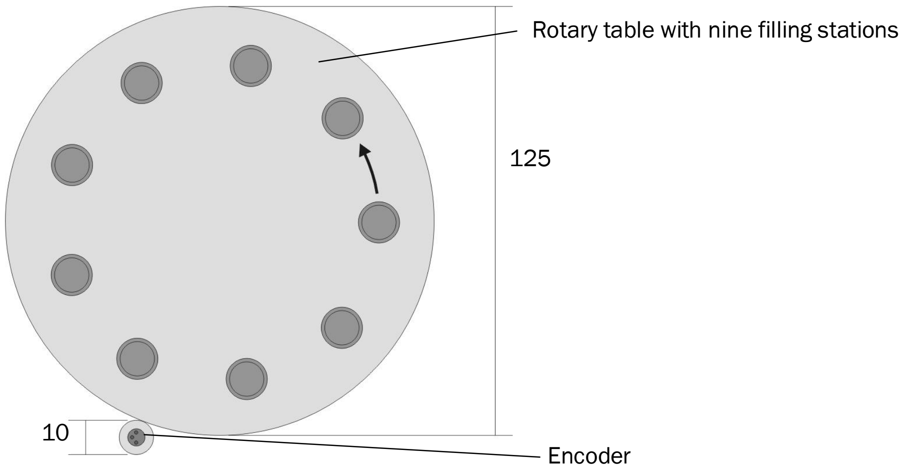


Figure 2: Example position measurement on a rotary table with transmission ratio

The number of revolutions is pre-defined by the transmission ratio = 12.5 of the rotary table gearing.

The total resolution is then 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, as the total resolution is not 2^n times the steps per revolution.

The application problem can be solved using the round axis functionality. Here the steps per revolution are ignored. The total resolution as well as the nominator and divisor for the number of revolutions are configured.

9000 steps are configured as the total resolution.

For the nominator for the number of revolutions 125 is configured, 10 as the divisor ($^{125}/_{10}$ = 12.5).

After 12.5 revolutions (that is after one complete revolution of the rotary table) the encoder reaches the total resolution of 9000.

Example without transmission ratio

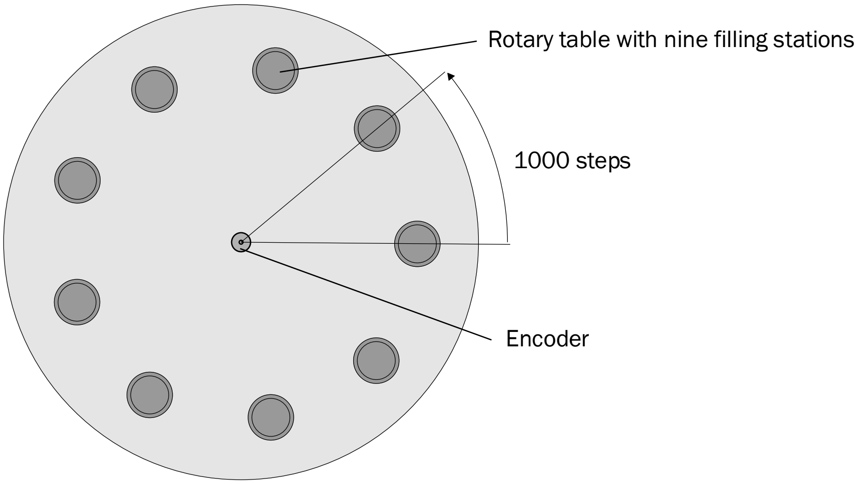


Figure 3: Example position measurement on a rotary table without transmission ratio

The encoder is mounted directly on the rotary table. The transmission ratio is 1:1.

The rotary table has 9 filling stations. The encoder must be configured such that it starts to count with 0 at one filling station and counts to 999 on moving to the next filling station position.

1000 steps are configured as the total resolution.

For the nominator for the number of revolutions 1 is configured, 9 as the divisor ($^1/_9$ revolutions = 1000).

After $^1/_9$ revolutions of the encoder shaft there are 1000 steps, then the encoder starts to count at 0 again.

3.3 Integration in EtherNet/IP

3.3.1 EtherNet/IP architecture

EtherNet/IP and therefore also the AFS60/AFM60 EtherNet/IP use Ethernet for the transmission technology.

The network components are generally integrated into a **star or line topology**.

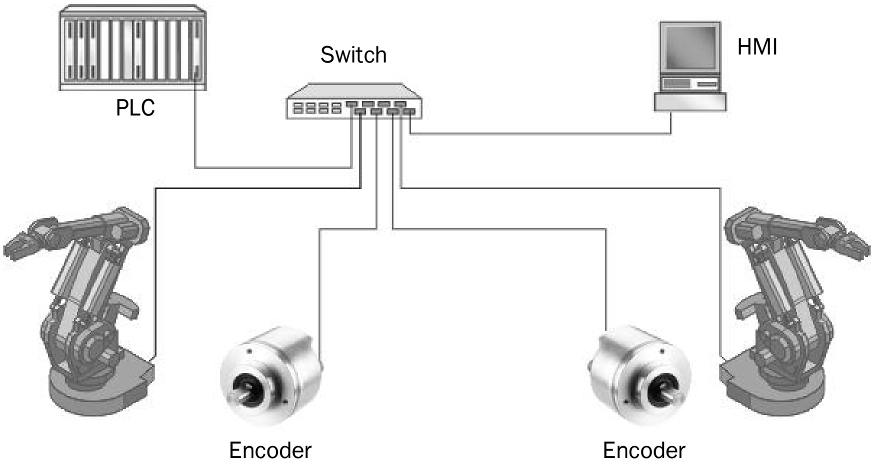


Figure 4: Example of an EtherNet/IP network in a star topology.

The system can also be integrated in a **Device Level Ring (DLR)** in order to achieve a higher reliability and less wiring effort.

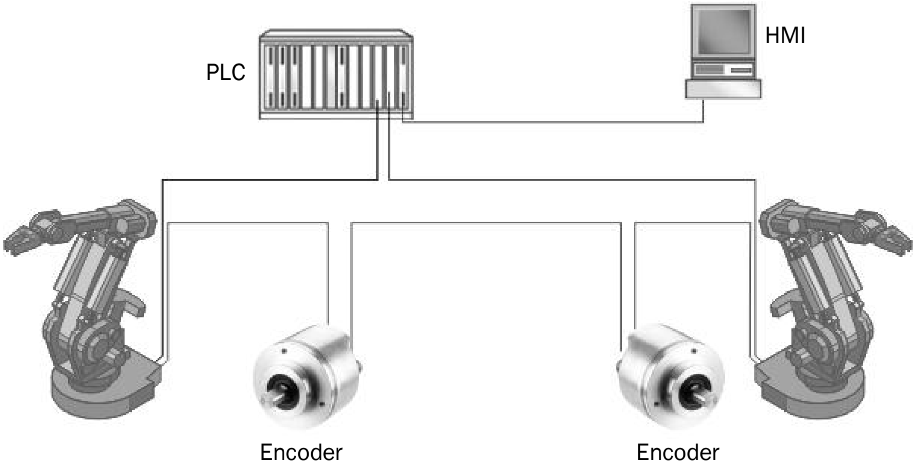


Figure 5: Example of an EtherNet/IP network in a Device Level Ring

The AFS60/AFM60 EtherNet/IP supports Device Level Ring.

3.3.2 EtherNet/IP communication

MAC address

Each AFS60/AFM60 EtherNet/IP has a factory-assigned worldwide unique MAC address for device identification. It is used for the identification of the Ethernet node. This 6 byte device identification can not be changed and comprises the following components:

- 3 bytes manufacturer ID
- 3 bytes device ID

TCP/IP and UDP/IP

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

For identification the IP address is required. A fixed address is assigned to the encoder using the address switches or the address is obtained from a DHCP server.

If the IP address is configured fix, only the least significant byte can be configured. 192.168.1.xxx is preset permanently.

Additionally the subnet mask (default = 255.255.255.0) and if required a gateway must be configured in the network.

For real-time communication between the controller and the encoder in EtherNet/IP **Implicit messaging** is used. With implicit messaging, a connection is established between two devices within the CIP to transfer, e.g., I/O data such as position, velocity etc. from the encoder to the controller (see also section 3.4.4 “Position Sensor Object” on page 28). Implicit messaging uses **UDP/IP** via port 2222. As a result a fast data rate is used.

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 transfer parameters from the controller to the encoder (see also section 3.4.3 “Assembly Object” on page 22).

Common Industrial Protocol (CIP)

EtherNet/IP uses the CIP on the process layer. Similarly as e.g. FTP is used for the transfer of files, this protocol is used for process control.

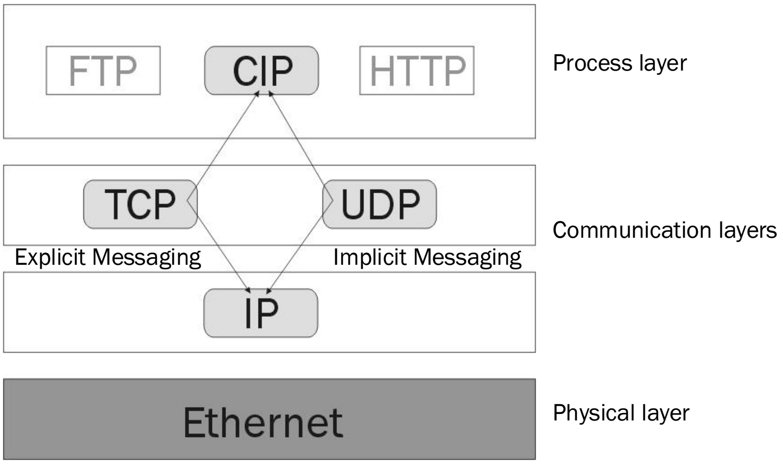


Figure 6: CIP and other services

The AFS60/AFM60 EtherNet/IP meets the requirements of the EtherNet/IP protocol according to IEC 61784-1 and those of the encoder profile 22h.

The encoder is an I/O adapter in the EtherNet/IP. It receives and sends explicit messages and implicit messages either cyclic or on request (polled).

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 of the source (source address) are contained in the Ethernet header.

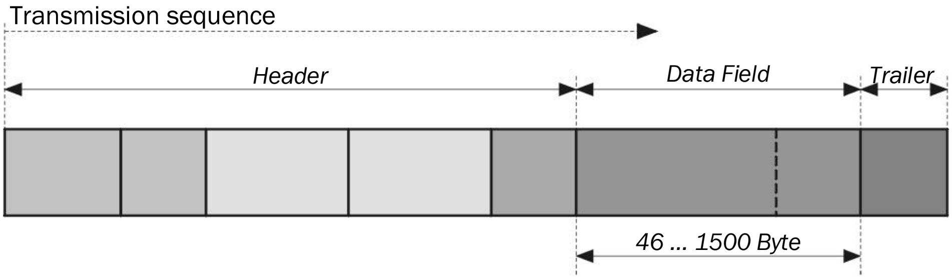


Figure 7: Ethernet FRAME

The Ethernet data field consists of several nested protocols:

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

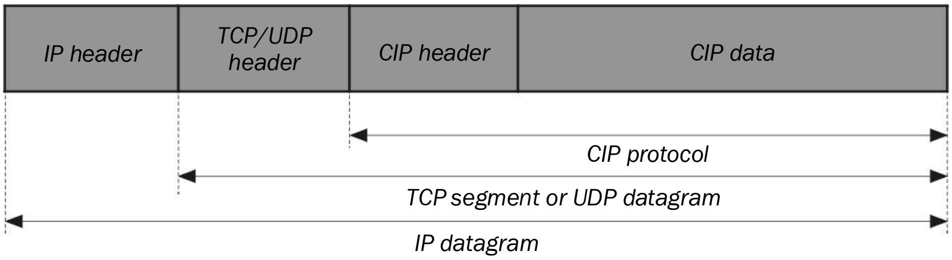


Figure 8: Ethernet data field

3.4 CIP object model

EtherNet/IP uses a so-called object model for network communication wherein all functions and data of a device are defined.

The most important terms are as follows:

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

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

Attribute The attributes represent the data a device provides over EtherNet/IP. These include the current values of, for example, a configuration or an input. Typical attributes are configuration or status information.

Service Services are used to access classes or the attributes of a class or to generate specific events. These services execute defined actions such as the reading of attributes.

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

Table 4: Example CIP object model

3.4.1 Supported classes

The AFS60/AFM60 EtherNet/IP supports the following classes of the 22h encoder profile:

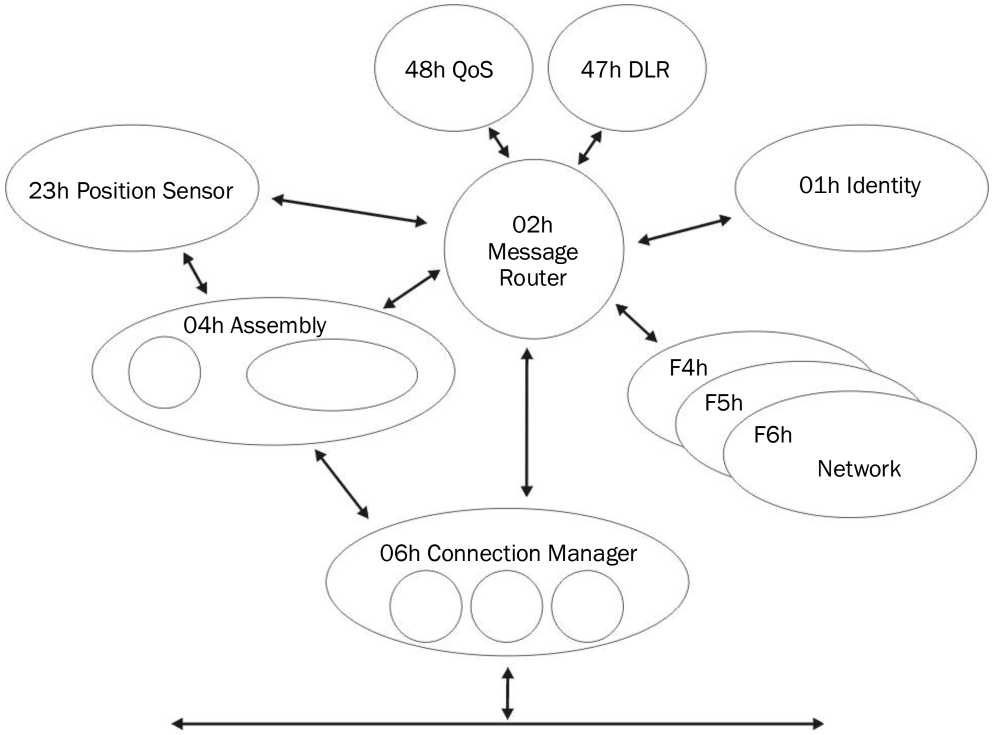


Figure 9: Supported classes

Class code	Class	Description	Access	Instances
01h	Identity Object	Includes all device specific data (e.g. ID, device type, device status etc.)	Get	1
02h	Message Router Object	Includes all supported class codes of the encoder and the maximum number of connections	Get	1
04h	Assembly Object	Assembles the data of several objects to one single object. Supplies (for example) the position value of the encoder	Get	7
06h	Connection Manager Object	Includes connection specific attributes for triggering, transport, connection type etc.	Get	1
23h	Position Sensor Object	Includes all attributes for the programming of the encoder parameters such as the scaling	Set/Get	1
F4h	Port Object	Includes the available ports, port name and node address	Get	1
F5h	TCP/IP Interface Object	Includes the attributes for TCP/IP such as IP address, subnet mask and gateway or acquisition of the IP address via DHCP or hardware switches	Set/Get	1

Class code	Class	Description	Access	Instances
F6h	Ethernet link object	Includes connection specific attributes such as transmission speed, interface status and the MAC address	Get	3
47h	Device Level Ring (DLR) Object	Includes status attributes and configuration attributes of the DLR protocol	Get	1
48h	Quality of Service (QoS) Object	Contains mechanisms for processing data streams with different priorities	Get	1

Table 5: Supported classes

3.4.2 Identity Object

The device information and device parameters are opened via the instances.

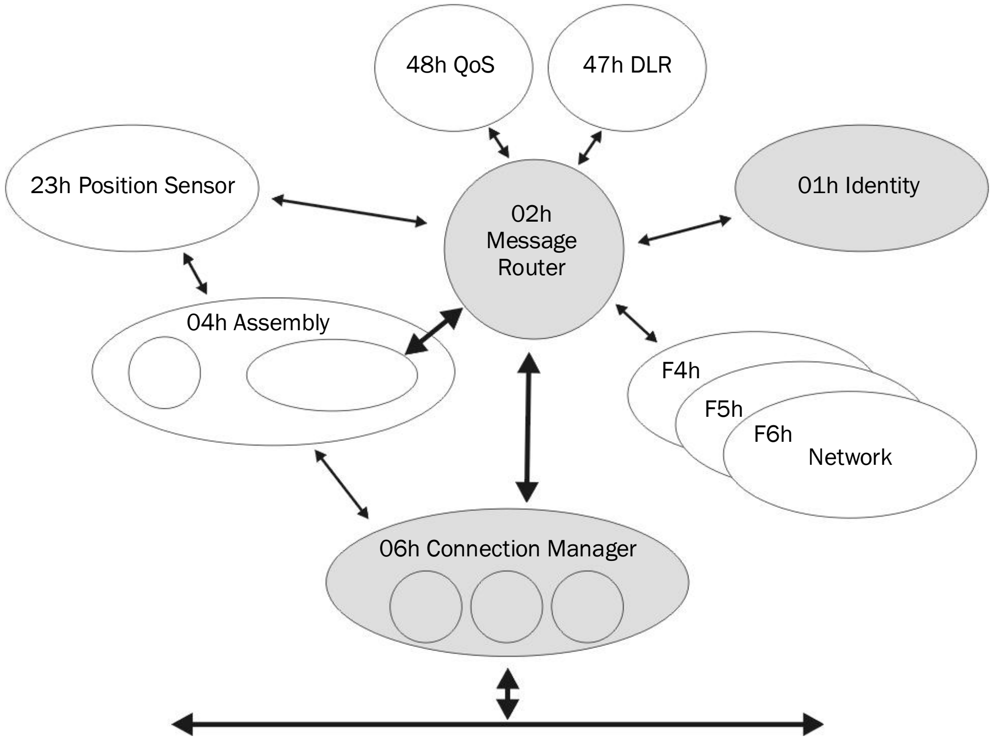


Figure 10: Connections for the Identity Object

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

Table 6: Class services of the Identity Object

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

Table 7: Class attributes of the Identity Object

NOTE

Class attribute 5 is not implemented.

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

Table 8: Instance Services 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	Vendor 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	First part of the revision number, e.g. 01 (depending on the release)	UINT	01h
	Get	Minor Revision	Last part of the revision number, e.g. 02 (depending on the release)	UINT	02h
05h	Get	Status	Device status flags	WORD	See Table 10

Attribute ID	Access	Name	Description	Data type	Default value
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	Firmware version in the FPGA (e. g. 1.2.0)	UDINT	00010200h

Table 9: Instance attributes of the Identity Object

Bit	Name	Description	Default value
0	Owned	0 = No connection to the master 1 = Connection to the master established	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	Vendor specific status bits	See Table 11
8	Minor Recoverable Status	0 = No error 1 = Recoverable error (device not in error status)	0
9	Minor Unrecoverable Status	0 = No error 1 = Recoverable error (device not in error status)	0
10	Major Recoverable Status	0 = No serious error 1 = Serious error that can be reset (device in error status)	0
11	Major Unrecoverable Status	0 = No serious error 1 = Serious error that cannot be reset (device in error status)	0
12 … 15	–	Reserved	0000

Table 10: Bits of the instance attribute "Status"

Possible combinations Bit 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	Serious error, bit 10 or bit 11 = 1
0110	At least one connection in the “Run” operating mode
0111	At least one connection exists, all in “Idle” operating mode
1000 … 1111	Reserved

Table 11: Bits 4 to 7 of the instance attribute “Status”

3.4.3 Assembly Object

The Assembly Object allows assembling of data attributes of other objects in one single object. The AFS60/AFM60 EtherNet/IP supports only static assemblies of attributes. For this reason the number of instances is fixed.

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

Table 12: Class services of the Assembly Object

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

Table 13: Class attributes of the Assembly Object

NOTE

Class attributes 4 and 5 are not implemented.

The encoder supports only “Input” and “Listen Only” connections.

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

Table 14: Instance Services 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 Velocity	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 Velocity	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 Velocity	32 32 32	12
110	3	Set/Get	Dummy instance for the configuration data of a “Listen-only” connection	0	0

Table 15: Instance attributes of the Assembly Object

NOTE

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

I/O Assembly

The I/O data are retrieved/output via instances.

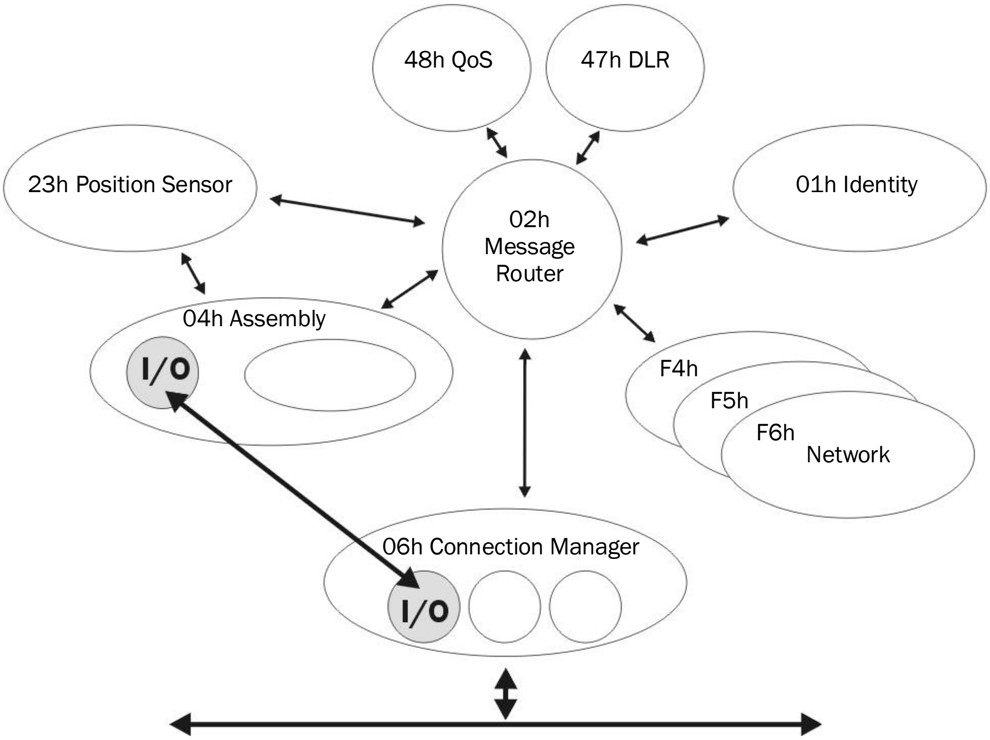


Figure 11: Connections for the I/O assembly

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	Velocity value (least significant byte)							
	5	Velocity value							
	6	Velocity value							
	7	Velocity 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 30 on page 103)							
	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 30 on page 103)							
	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	Velocity value (least significant byte)							
	9	Velocity value							
	10	Velocity value							
	11	Velocity value (most significant byte)							

Table 16: Data format of the attributes of the I/O assembly

Configuration Assembly

The encoder can be configured via the configuration assembly.

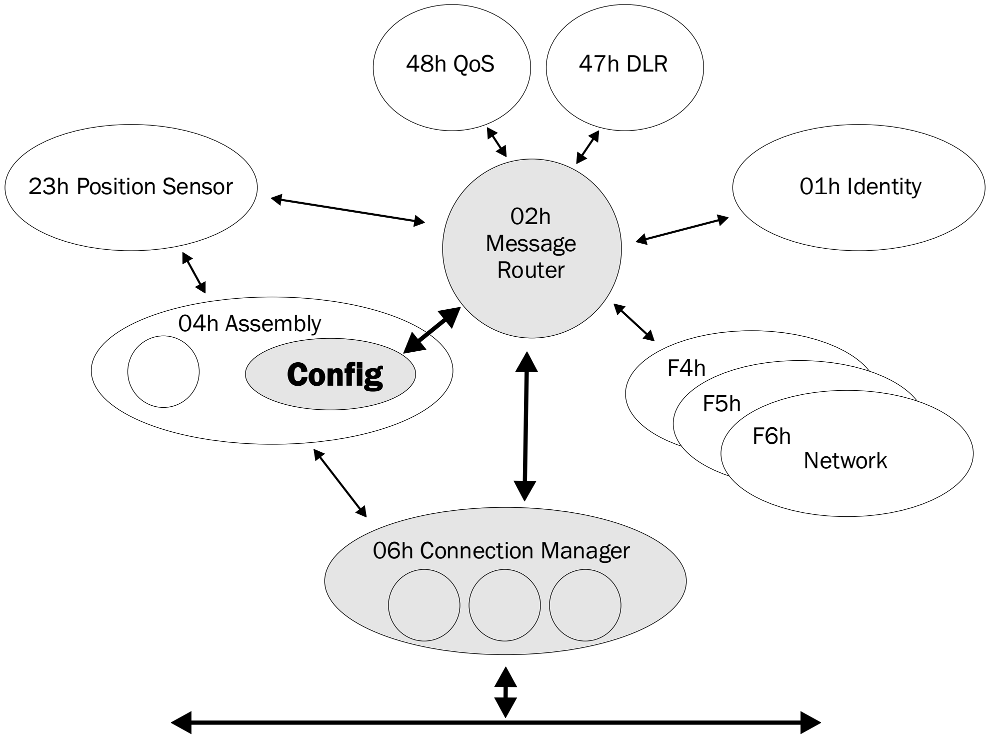


Figure 12: Connections for the configuration assembly

NOTE

- If you integrate the encoder as a generic module, then you can activate or not activate the configuration assembly **independent** of the I/O assembly instances.
- If you use the EDS file (electronic data sheet) for the encoder, then the configuration assembly is activated or not activated **depending** on the I/O assembly instances:
  - ○ active with instances 101, 102 and 103
  - ○ not active with instances 101WS, 102WS and 103WS
- If the configuration assembly is activated, then it is not allowed to be empty. Otherwise in some circumstances the control system may output an error.

<table>
<tr><th>Instance</th><th>Byte</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td rowspan="28"><b>100</b></td><td>0</td><td colspan="8">Not used</td></tr>
<tr><td>1</td><td colspan="8">Not used</td></tr>
<tr><td>2</td><td colspan="8">Not used</td></tr>
<tr><td>3</td><td colspan="8">Not used</td></tr>
<tr><td>4</td><td colspan="8">Steps per revolution CPR (least significant byte)</td></tr>
<tr><td>5</td><td colspan="8">CPR</td></tr>
<tr><td>6</td><td colspan="8">CPR</td></tr>
<tr><td>7</td><td colspan="8">CPR (most significant byte)</td></tr>
<tr><td>8</td><td colspan="8">Total resolution CMR (least significant byte)</td></tr>
<tr><td>9</td><td colspan="8">CMR</td></tr>
<tr><td>10</td><td colspan="8">CMR</td></tr>
<tr><td>11</td><td colspan="8">CMR (most significant byte)</td></tr>
<tr><td>12</td><td colspan="7">Not used</td><td>cw/
ccw [1]</td></tr>
<tr><td>13</td><td colspan="7">Not used</td><td>scf [2]</td></tr>
<tr><td>14</td><td colspan="7">Not used</td><td>raf [3]</td></tr>
<tr><td>15</td><td colspan="8">Not used</td></tr>
<tr><td>16</td><td colspan="8">Nominator for the number of revolutions CNR_N (least significant byte)</td></tr>
<tr><td>17</td><td colspan="8">CNR_N</td></tr>
<tr><td>18</td><td colspan="8">CNR_N</td></tr>
<tr><td>19</td><td colspan="8">CNR_N (most significant byte)</td></tr>
<tr><td>20</td><td colspan="8">Divisor for the number of revolutions CNR_D (least significant byte)</td></tr>
<tr><td>21</td><td colspan="8">CNR_D</td></tr>
<tr><td>22</td><td colspan="8">CNR_D</td></tr>
<tr><td>23</td><td colspan="8">CNR_D (most significant byte)</td></tr>
<tr><td>24</td><td colspan="8">Velocity measuring unit (least significant byte)</td></tr>
<tr><td>25</td><td colspan="8">Velocity measuring unit (most significant byte)</td></tr>
<tr><td>26</td><td colspan="8">Not used</td></tr>
<tr><td>27</td><td colspan="8">Not used</td></tr>
</table>

Table 17: Data format for the attributes for the configuration assembly

NOTE

- The structure of the configuration assembly is fixed.
- During the initialization of the encoder, it reads the data from the control system.
- The “Heartbeat connection point” for PLC input connections, that is for the encoder output, must be set to 198 (see Figure 30 on page 54).
- The “Heartbeat connection point” for listen-only connections must be set to 199.

[1] cw = clockwise.
ccw = counterclockwise.

[2] scf = scaling function.

[3] raf = round axis functionality.

3.4.4 Position Sensor Object

The Position Sensor Object contains all the attributes of the encoder. All parameters can be retrieved or set using explicit messages.

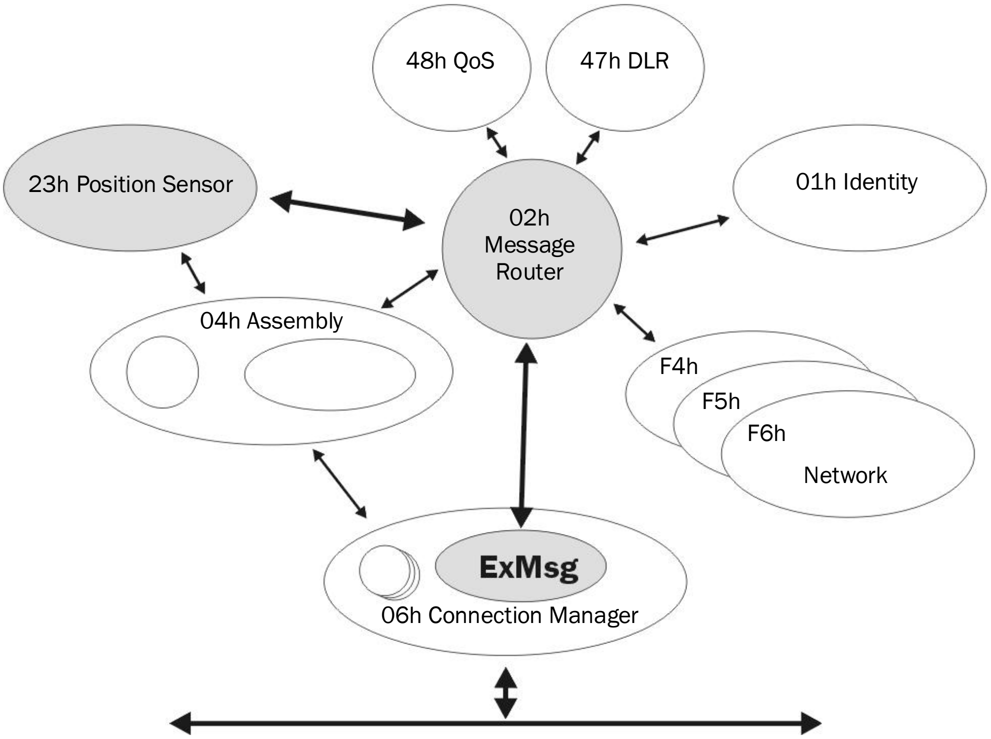


Figure 13: Connections for explicit messages to the Position Sensor Object

Service code	Service	Description
05h	Reset	Resets the encoder to the default factory settings
0Eh	Get_Attribute_Single	Returns the values of one attribute
15h	Restore	Restores all parameters last saved in non-volatile memory
16h	Save	Saves parameters in the non-volatile memory (see section 3.6.1 on page 37)

Table 18: Class services 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 within this class	UINT	0001h
3	Get	Number of object instances in this class	UINT	0001h
4	Get	Optional attribute list	STRUCT	–
5	Get	Optional services list	STRUCT	–
6	Get	Highest existing class attribute ID	UINT	0064h
7	Get	Highest implemented instance attribute	UINT	–
100	Get	Firmware version	Array	AFx_aa.bb. dd.mm.yy

Table 19: Class attributes of the Position Sensor Object

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

Table 20: Instance services of the Position Sensor Object

Attribute ID	Access	V/NV [4]	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 the 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 the position measurement 1001h = Steps	ENG UINT	(1001h)

[4] V = volatile, NV = non-volatile.

Attribute ID	Access	V/NV [4]	Name	Description	Data type	Min. Max. (default value)
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	Minimum 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 whether the limit set by the attributes 16h and 17h is dropped below/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 for the position [5]	DINT	00000000h 3FFFFFFFh (00000000h)
17h	Set	NV	Position high limit	Upper limit for the position [5]	DINT	00000000h 3FFFFFFFh (3FFFFFFFh)
18h	Get	V	Velocity Value	Current velocity. The format is defined by the attributes 19h and 1Ah.	DINT	00000000h XXXXXXXXh [6]
19h	Set	NV	Velocity Format	Velocity 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 velocity measurement	DUINT	(00000001h)

[5] Using the lower and upper limit for the position you can realize range monitoring. This is not an electronic cam.

[6] The maximum velocity is dependent on the mechanical interface used, “solid shaft” or “blind hollow shaft” (see data sheet).

Attribute ID	Access	V/NV [4]	Name	Description	Data type	Min. Max. (default value)
1Bh	Set	NV	Minimum Velocity Setpoint	Lower/upper limit for the velocity in turns/min [7]. If the velocity drops below/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	Accelera-tion value	Current acceleration. The format is defined by the attributes 1Eh and 1Fh.	DINT	00000000h FFFFFFFFh
1Eh	Set	NV	Accelera-tion 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	Accelera-tion resolution	Minimum resolution of the acceleration measurement	DUINT	(1)
20h	Set	NV	Minimum Accelera-tion Setpoint	Lower/upper limit for the acceleration in counts/ms^2 [8]. If the acceleration drops below/exceeds this value, the warning flag (attribute 2Fh) is set.	DINT	(C0000001h)
21h	Set	NV	Maximum accelera-tion setpoint		DINT	(3FFFFFFFh)
29h	Get	V	Operating Status	Operating status of the encoder Bit 0: Direction 0 = Counting up 1 = Downward counting 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)

[7] The unit changes with the velocity format (attribute ID 19h). The limits must then be converted correspondingly, e.g. 12,000 turns/min = 200 turns/s.

[8] The unit changes with the acceleration format (attribute ID 1Eh). The limits must then be converted correspondingly, e.g. 2 counts/ms^2 = 2,000,000 counts/s^2.

Attribute ID	Access	V/NV [4]	Name	Description	Data type	Min. Max. (default value)
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 31: Alarms on page 104)	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	–
2Fh	Get	V	Warnings	Bit field with flags for warnings (see Table 32: Warnings on page 105)	WORD	–
30h	Get	NV	Supported Warnings	Supported warnings	WORD	67C3h
31h	Get	V	Warning Flag	0 = No warning 1 = Warning	BOOL	–
32h	Get	NV	Operating Time	Saved operating time in 0.1 h = 6 min	UDINT	0
33h	Get	NV	Offset Value	Offset value is calculated on the initialization of 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	Lowest resolution for the temperature (°C/100 or °F/100)	UDINT	(00000064h)
67h	Set	NV	Minimum Temperature Setpoint	Lower/upper limit for the temperature in °C [9]. If the temperature drops below/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)

[9] The unit changes with the temperature value format (attribute ID 65h). The limits must then be converted correspondingly.

Attribute ID	Access	V/NV [4]	Name	Description	Data type	Min. Max. (default value)
69h	Get	V	Fault header	See Table 30 on page 103	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 … 31: Not used	DWORD	(00000500h)
6Bh	Get	NV	Encoder Motion Time	Saved motion time in seconds (is increased in case of movement)	UDINT	–
6Ch	Get	NV	Encoder Operating Time	Saved operating time in seconds (is increased as soon as the encoder is in operation)	UDINT	–
6Dh	Get	NV	Max. velocity	Highest velocity that the encoder has reached since start-up [10]	UDINT	–
6Eh	Get	NV	Max. acceleration	Highest acceleration that the encoder has reached since start-up [11]	UDINT	–
6Fh	Get	NV	Max. temp	Highest operating temperature reached 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 times the encoder has been commissioned (powered on)	UDINT	–
72h	Get	V	LED Current Value	Actual internal LED current of the sensor in µA	UINT	200 25,000 (0)
73h	Get	NV	Max. current value	Maximum internal LED current for the sensor in µA	UINT	200
74h	Get	NV	Min. Current Value	Minimum internal LED current in the sensors in µA	UINT	25,000

[10] The value is output in the format defined in attribute ID 19h.

[11] The value is output in the format defined in attribute ID 1Eh.

Attribute ID	Access	V/NV [4]	Name	Description	Data type	Min. Max. (default value)
75h	Get	V	Direction change counter	Number of changes in the direction of rotation (The counter increments if the encoder changes direction of rotation.)	UDINT	0
76h	Get	V	Revolution counter forward	Number of clockwise starts (The counter is increased if the encoder moves clockwise.)	UDINT	0
77h	Get	V	Revolution counter backwards	Number of counterclockwise starts (The counter is increased if the encoder moves counterclockwise.)	UDINT	0
78h	Get	V	Power Supply Voltage	Current operating voltage in mV	UINT	9,500 30,500 (24,000)
79h	Get	V	Max. power supply voltage	Maximum operating voltage in V (is saved in EEPROM)	UINT	0 33 (0)
7Ah	Get	V	Preset Offset Value	Offset value calculated from the preset value [12]	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	Nominator for the number of revolutions	UDINT	1 2,048 (2,048)
7Fh	Set	NV	Number of Revolutions, Divisor	Divisor for the number of revolutions	UDINT	1 65,535 (1)
80h	Set	NV	Velocity Filter Integration Time	Number of measured values from which an average value is formed	UDINT	0 128 (1)
81h	Set	NV	Velocity Filter Bandwidth	Bandwidth of the low pass filter in Hz 0 = Deactivated	UDINT	0 1000 (100)
82h	Set	NV	Acceleration filter integration time	Number of measured values from which an average value is formed	UDINT	0 128 (1)

[12] With normal scaling = physical position; for round axis functionality = physical position + range offset.

Attribute ID	Access	V/NV [4]	Name	Description	Data type	Min. Max. (default value)
83h	Set	NV	Acceleration filter bandwidth	Bandwidth of the low pass filter in Hz 0 = Deactivated	UDINT	0 1000 (100)
84h	Set	NV	Velocity Hysteresis	Hysteresis for the velocity limits (attributes 1Bh and 1Ch) The unit depends on attribute ID 19h.	UDINT	0 3FFFFFFF (0)
85h	Set	NV	Acceleration hysteresis	Hysteresis for the acceleration limits (attributes 20h and 21h) The unit depends on the attribute ID 1Eh.	UDINT	0 3FFFFFFF (0)
86h	Set	V	Motion time limit	Limit for the motion time in seconds	UDINT	00000000h FFFFFFFFh (630,720,000)
87h	Set	V	Power time limit	Limit for the operating time in seconds	UDINT	00000000h FFFFFFFFh (630,720,000)
88h	Set	V	Direction changes limit	Limit for the number of changes in the direction of rotation	UDINT	00000000h FFFFFFFFh (1,000,000)
89h	Set	V	Starts in cw limit	Limit for the number of clockwise starts	UDINT	00000000h FFFFFFFFh (1,000,000)
8Ah	Set	V	Starts in ccw limit	Limit for 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 30 on page 103)	Byte	(00h)

Table 21: Instance attributes of the Position Sensor Object

Filter for the velocity (attribute 80h and 81h) or the acceleration (attribute 82h and 83h)

The filters are used to smooth the raw velocity and acceleration values.

NOTE

The filters are applied in the following sequence:

- integration time filter for the velocity (80h) or acceleration (82h)
- low pass filter for the velocity (81h) or acceleration (83h)

The filter with the attribute 80h forms an average value from the measured velocity values. The filter with the attribute 82h forms an average value from the measured acceleration values:

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

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

- From the factory this is configured to 100 Hz. I.e. only velocity and acceleration values ≤ 100 Hz are taken into account.

3.5 Integration and configuration options

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

3.5.1 Integration in EtherNet/IP

The encoder can be integrated in EtherNet/IP:

- as Generic Modules (see section 5.4 on page 53):

  You enter all module settings manually.

- with the aid of an EDS file (see section 5.5 on page 56):

  The module settings for the encoder AFS60/AFM60 EtherNet/IP are already predefined.

3.5.2 Configuration

The following options are available to configure the encoder:

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

Case 1: On integration as a generic module

If you have integrated the encoder as a generic module, then you can configure it depending on the **Connection Parameters** entered.

- If the configuration assembly is **activated** in Connection Parameters, then you must use the configuration assembly for configuration (see section 5.4.1 on page 54).
  In addition you can configure the parameters that are not contained in the configuration assembly using the web server integrated in the encoder.
- If the configuration assembly is **not activated** in Connection Parameters, you can use the web server to configure all parameters (see chapter 6 on page 87).

NOTE

If the configuration assembly is active, all the parameters entered there overwrite the parameters that have been configured using the web server.

Case 2: On integration with the aid of the EDS file

If you have integrated the encoder with the aid of the EDS file, then you can configure it depending on the selected I/O assembly instances (see Table 15 on page 23).

- If you use the instances 101, 102 or 103, then the configuration parameters can be configured in the **Controller Tags**. In addition you can use the web server to configure the parameters that are not contained in the configuration assembly.
- If you use the instances 101WS, 102WS or 103WS, then you can use the web server to configure the parameters.

Case 3: On usage of the ladder routine for the configuration mapping

A ladder routine is available for mapping the configuration data for the AFS60/AFM60 EtherNet/IP (see section 5.6 on page 59).

If the ladder routine is used for mapping, and you use the instances 101WS, 102WS or 103WS (see Table 15 on page 23), then the encoder can be configured from the control system (in the **Controller Tags**) and also with the aid of the web server.

NOTE

In cases 1 and 2 the parameters are configured offline and written to the encoder and activated on changing to the online mode.

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

Parameter changes via the web server are applied immediately on the control system side and displayed. Parameter changes via the control system are applied immediately. However, to display them in the web browser you must refresh the related page.

WARNING
Before changing the configuration, check whether there is a hazard from the machine or system in which the encoder is integrated!

The ladder routine offers the possibility to change the parameter data during operation, i.e. **while the control system is in the online mode**.

The change to the configuration therefore has immediate effects on the data output from the encoder. This change could cause an unexpected reaction that may result in a hazard for persons or damage to the system or other items.

3.6 Configurable functions

3.6.1 Saving configuration and resetting

The configuration memory in the AFS60/AFM60 EtherNet/IP is divided into three. The following table shows the functions of the memory types.

Memory type	Function
Volatile memory	During operation the encoder operates with the values in the volatile memory. Modified parameters are initially written to the volatile memory. These data are lost on switching off.
Non-volatile memory	On switching on, the encoder loads the values from the non-volatile memory into the volatile memory.
Default factory settings	Contains the pre-set values from the factory.

Table 22: Configuration memory — functions of the different types of memory

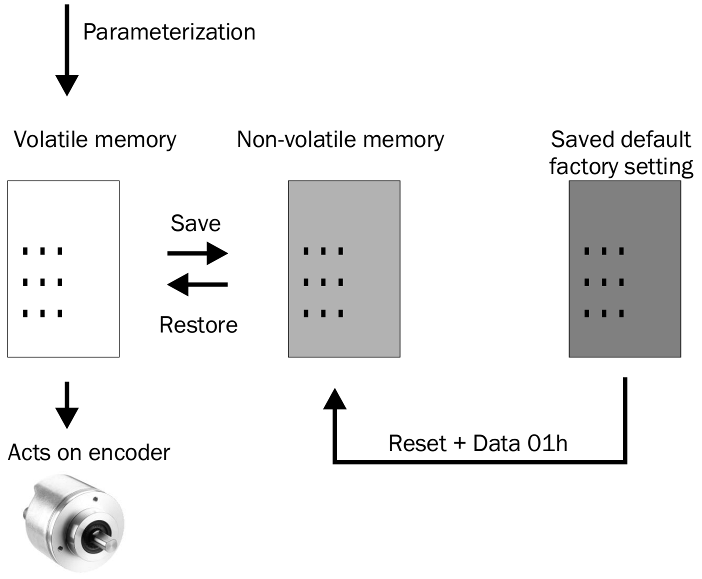


Figure 14: Configuration memory

Reset: Reset to the default factory settings

- Set the address switches to 888 (see Figure 18 on page 46).
- Press the preset push-button for longer than 5 seconds.

Or:

- Use the class service **Reset** (service code 05h) in the Position Sensor Object (23h) and set the data to 01h.

  The parameters for the Position Sensor Object are reset to the factory settings. Table 23 on page 39 shows which parameters are reset to which value.

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

Each time the encoder is switched on the values for the Position Sensor Object are read from the non-volatile memory.

- Use the class service **Restore** (service code 15h) in the Position Sensor Object if you want to read the parameters from the non-volatile memory during operation. The parameters that have been changed since switching on but not yet saved are lost.

Save: Save parameters in the non-volatile memory

- Use the class service **Save** (service code 16h) in the Position Sensor Object.

  The parameters are saved in the non-volatile memory. Table 23 on page 39 shows which parameters are saved.

Parameters that are saved or reset

Attribute ID in the position sensor object	Parameter	Default factory setting
0Ch	Code sequence	cw
0Eh	Scaling	Off
10h	Steps per revolution	262,144
11h	Total resolution	1,073,741,824

Attribute ID in the position sensor object	Parameter	Default factory setting
13h	Preset value	0
16h	Lower limit for the position	0
17h	Upper limit for the position	1,073,741,823
19h	Velocity unit	Turns/min
1Bh	Lower limit for the velocity	–12,000
1Ch	Upper limit for the velocity	12,000
1Eh	Acceleration unit	Counts/ms²
20h	Lower limit for the acceleration	–1,073,741,823
21h	Upper limit for the acceleration	1,073,741,823
65h	Temperature unit	°C
7Dh	Round axis functionality	Off
7Eh	Nominator for the number of revolutions	2,048
7Fh	Divisor for the number of revolutions	1
80h	Number of measured values from which an average value is formed	1
81h	Bandwidth of the low pass filter	100
82h	Number of measured values from which an average value is formed	1
83h	Bandwidth of the low pass filter	100
84h	Hysteresis for the velocity limits	0
85h	Hysteresis for the acceleration limits	0
86h	Limit for the motion time in seconds	630,720,000
87h	Limit for the operating time in seconds	630,720,000
88h	Limit for the number of changes in the direction of rotation	1,000,000
89h	Limit for the number of clockwise starts	1,000,000
8Ah	Limit for the number of counterclockwise starts	1,000,000

Table 23: Parameters that are saved or reset

NOTE

The following parameters are not reset:

- motion time
- operating time
- lower limit for the temperature
- upper limit for the temperature
- maximum voltage supply

3.6.2 IP address

For identification of the encoder in the EtherNet/IP, the IP address is required. This address is obtained for the encoder from a DHCP server (see section 5.2.2 on page 48) or a fixed address is set using address switches (see section 4.2.1 on page 46).

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

3.6.3 Slave Sign of Life

The AFS60/AFM60 EtherNet/IP supports Slave Sign of Life functionality.

It is transferred in bit 30 of the fault header. It is used so that the control system can determine whether the encoder is in operation, even if the position data do not change (e.g. at standstill).

The bit changes its value at the Update Cycle configured.

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 defined using attribute 6Ah in the Position Sensor Object (see Table 21 on page 35).

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

3.6.4 Code sequence

The code sequence defines the direction of rotation, viewed on the shaft, in which the position value increases.

- clockwise = increasing position value on clockwise revolution of the shaft
- counterclockwise = increasing position value on counterclockwise revolution of the shaft

3.6.5 Scaling

The scaling makes it possible to scale the steps per revolution and the total resolution.

NOTE

Only if the parameter **Scaling** (attribute ID 0Eh of the Position Sensor Object) is configured to **Enable**, the values entered for the steps per revolution and the total resolution are applied.

3.6.6 Steps per revolution

The resolution of the AFS60/AFM60 EtherNet/IP is max. 262,144 steps per revolution. The resolution can be scaled from 1 ... 262,144 as an integer.

NOTE

The parameter is not used if the round axis functionality (see section 3.6.10 on page 42) is activated.

3.6.7 Total resolution/measuring range

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

Steps per revolution	n	Total resolution
1,000	3	8,000
8,179	5	261,728
2,048	11	4,194,304

Table 24: Examples for total resolution

NOTE

This restriction is not relevant if the round axis functionality (see section 3.6.10 on page 42) is activated.

3.6.8 Preset function

The preset function is used to set the encoder to a predefined start position. With the aid 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 manner:

- with the aid of the preset pushbutton
- using an acyclic explicit message
  During this process the preset value is transferred as an attribute (13h) of the Position Sensor Object.
- with the aid of the integrated web server and the ladder routine

NOTE

▶ Only set a preset value when the encoder is at standstill.

WARNING

Before triggering the preset function, check whether there is a hazard from the machine or system in which the encoder is integrated!

The preset function results in an immediate change in the position value output by the encoder. This change could cause an unexpected movement that may result in a hazard for persons or damage to the system or other items.

3.6.9 Velocity measuring unit

Using this parameter you can define the units in which the velocity is transmitted.

Possible units are:

- counts/s [13]
- counts/ms [13]
- turns/s
- turns/min
- turns/h

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

3.6.10 Round axis functionality

NOTE

Only the multiturn encoder supports the round axis functionality.

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

The steps per revolution are not configured directly, instead the nominator and divisor for the number of revolutions are defined.

The following requirements must be met:

- attribute ID 0Eh, Scaling must be set to 1.
- attribute ID 11h, Total resolution must be set to between 1 ... 536,870,912.
- attribute ID 7Dh, Round axis functionality must be set to 1.
- attribute ID 7Eh, Nominator (CNR_N) must be set to 1 ... 2,048.
- attribute ID 7Fh, Divisor (CNR_D) must be set to between 1 ... 65,535.

Number of revolutions, divisor

The nominator can be scaled from 1 ... 2,048 as an integer. The default factory setting for the nominator is 2,048.

Number of revolutions, nominator

The divisor can be scaled from 1 ... 65,535 as an integer. The default factory setting for the divisor is 1.

Pay attention to the following restrictions:

- The total resolution of the round axis functionality is half the physical resolution (PhysRes) of the encoder = 536,870,912.
- the total resolution ≤ CNR_N ÷ CNR_D × PhysRes
- 1 ≤ nominator ≤ ½ × 4,096
- 1 ≤ divisor ≤ 65,535
- (CNR_N ÷ CNR_D) ≤ ½ × 4,096

[13] Depending on the resolution configured.
Example: Resolution = 2,000 steps; the encoder rotates 0.5 times per second = 1.000 counts/s or 1 counts/ms.

3.7 Controls and status indicators

The AFS60/AFM60 EtherNet/IP Absolute Encoder has five LEDs.

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

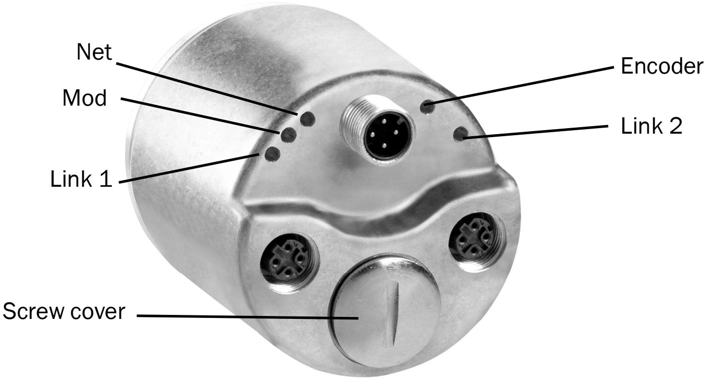


Figure 15: Position of the LEDs, the address switches and the preset pushbutton

The LEDs are multi-colored. Table 28 on page 101 and Table 29 on page 102 show the meaning of the signals.

There are the following controls under the screw cover:

- address switches
- preset pushbutton

4 Commissioning

This chapter provides information on the electrical installation, configuration and commissioning of the AFS60/AFM60 EtherNet/IP.

▶ Please read this chapter before mounting, installing and commissioning the device.

4.1 Electrical installation

WARNING
Switch the power supply off!

The machine/system could unintentionally start up while you are connecting the devices.

▶ Ensure that the entire machine/system is disconnected during the electrical installation.

For the electrical installation you will need male and female connectors (see data sheet for the AFS60/AFM60 EtherNet/IP).

4.1.1 Connections of the AFS60/AFM60 EtherNet/IP

The connections of the AFS60/AFM60 EtherNet/IP are on the back.

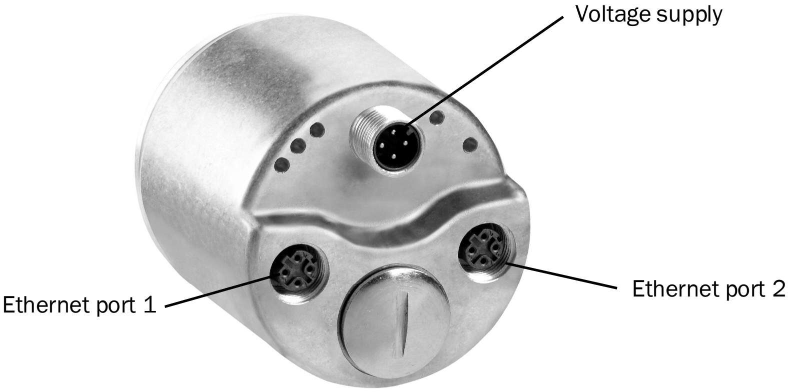


Figure 16: Position of the connections of the AFS60/AFM60 EtherNet/IP

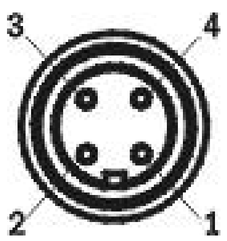
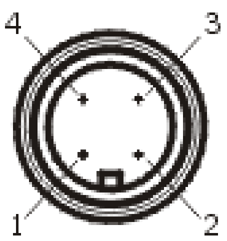
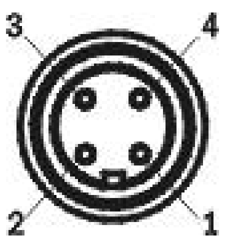
 Ethernet port 1 M12 × 4, female connector	 Voltage supply M12 × 4, plug	 Ethernet port 2 M12 × 4, female connector

Figure 17: Connections of the AFS60/AFM60 EtherNet/IP

NOTE

Two Ethernet connections are used if the AFS60/AFM60 EtherNet/IP is integrated in a DLR or a line topology (see Figure 5 on page 15).

Pin	Signal	Wire color [14]	Function
1	V_S	Brown	Supply voltage 10 … 30 V DC
2	–	White	Do not use
3	GND	Blue	0 V DC (ground)
4	–	Black	Do not use

Table 25: Pin assignment for the connection of the voltage supply

NOTE

Pin 2 and 4 are **not allowed to be assigned**, otherwise irreparable damage could be caused to the AFS60/AFM60 EtherNet/IP.

Pin	Signal	Wire color [14]	Function
1	TxD+	White/orange	Ethernet
2	RxD+	White/gray	Ethernet
3	TxD–	Orange	Ethernet
4	RxD–	Green	Ethernet

Table 26: Pin assignment for the Ethernet port 1 and 2 connections

NOTE

- ▶ **Connect the shield to the encoder housing!**
- ▶ Pay attention to the maximum cable lengths.
- ▶ Mount all cables with strain relief.

4.2 Settings on the hardware

There are the following controls for making settings under the screw cover:

- three address switches
- preset pushbutton

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

[14] On the usage of pre-wired cables.

4.2.1 Setting the IP address

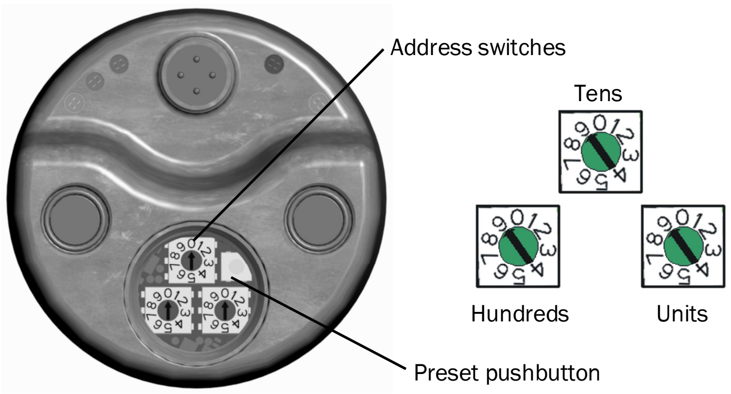


Figure 18: Address switch an preset pushbutton

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 preset permanently. • Subnet mask 255.255.255.0 is preset permanently. • Gateway address 0.0.0.0 is preset permanently.
000/999	On switching on, the encoder loads the IP address from the non-volatile memory.

Table 27: Address switches — Meaning of the values that can be set

Fixed IP address via address switches

- Set the hundreds for the address using the left address switch.
- Set the decades for the address using the center address switch.
- Set the units for the address using the right address switch.

Acquiring the IP address via DHCP

- Turn the encoder off.
- Set the address switches to 888.
- Switch back on the encoder.

The encoder now obtains its IP address from a DHCP server and saves this address in the non-volatile memory.

If necessary deactivate the DHCP function in the encoder (see section 5.2.3 on page 50).

Using the following procedure you can ensure that the encoder retains the IP address assigned via DHCP also after switching back on:

- Set the address switches to 000.
The encoder now loads the IP address from the non-volatile memory each time on switching on.

4.2.2 Triggering a preset value using the preset button

▶ To trigger the preset value, press the preset pushbutton [15].
The value from attribute 13h of the Position Sensor Object is used as a new position value (see Table 21 on page 35).

NOTE

- Only set a preset value when the encoder is at standstill.
- The preset value must lie within the measuring range configured.

WARNING

Before triggering the preset function, check whether there is a hazard from the machine or system in which the encoder is integrated!

The preset function results in an immediate change in the position value output by the encoder. This change could cause an unexpected movement that may result in a hazard for persons or damage to the system or other items.

[15] **Under no circumstances** press the Preset button for longer than 5 seconds, this action would reset the encoder to the factory settings.

5 Configuration with the aid of a PLC

The AFS60/AFM60 EtherNet/IP can be integrated into both an Allen-Bradley control system from Rockwell and into other systems with a control system that features an EtherNet/IP communication interface.

NOTE

- All software notes are displayed in English.
- All software notes are related to RSLogix 5000 software.
  For the following example project the Allen-Bradley control system "ControlLogix Controller 1756-L61" with "RSLogix 5000" is used. It is a prerequisite that the hardware has already been installed.

5.1 Default delivery status

The AFS60/AFM60 EtherNet/IP is supplied 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
- velocity measuring unit = turns/min
- round axis functionality = not activated
- nominator for the number of revolutions (round axis functionality) = 2,048
- divisor for the number of revolutions (round axis functionality) = 1
- position of the address switches = 999 (DHCP activated)

5.2 IP address of the encoder

5.2.1 Without DHCP server

If you have entered the IP address of the encoder via the address switches (see section 4.2.1 on page 46), then you must use this IP address in the control system.

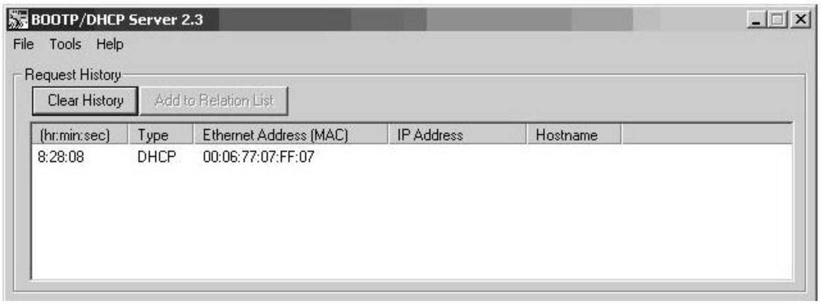
NOTE

In this way the address range is limited to 192.168.1.xxx. Only if the IP address is obtained via DHCP, any address range is possible.

5.2.2 IP address assignment via DHCP

If your control system has a DHCP server, then you can transfer an IP address to the encoder via this DHCP server.

▶ Start the **BOOTP/DHCP Server** (as a rule on the Start menu on your PC/notebook in **Rockwell Software**, **BOOTP-DHCP Server**, **BOOTP-DHCP Server**).



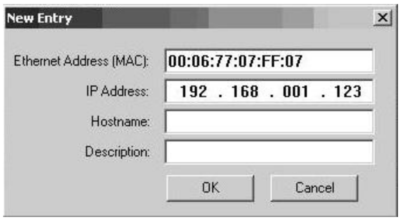
(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
8:28:08	DHCP	00:06:77:07:FF:07		

Figure 19: MAC address in the BOOTP/DHCP server

In the program window for the BOOTP/DHCP server the AFS60/AFM60 EtherNet/IP appears as a bus user with its MAC address, however without an IP address assigned.

☀ Green The Mod LED on the AFS60/AFM60 EtherNet/IP flashes green (the encoder does not yet have an IP address).

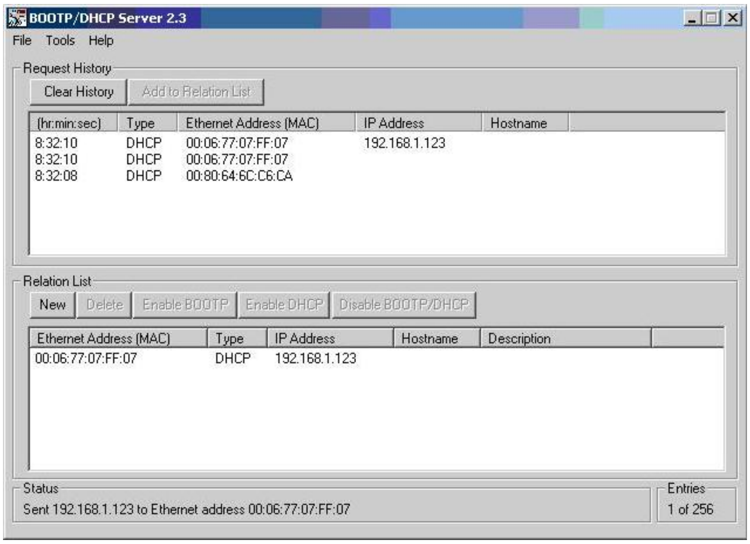
▶ Open the encoder in the BOOTP/DHCP server by double-clicking.



Ethernet Address (MAC): 00:06:77:07:FF:07
IP Address: 192 . 168 . 001 . 123
Hostname:
Description:

Figure 20: Entry of the IP address in the BOOTP/DHCP server

▶ In the **IP Address** field type a valid, spare address and click **OK**.
▶ Click on **Clear History**.



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
8:32:10	DHCP	00:06:77:07:FF:07	192.168.1.123	
8:32:10	DHCP	00:06:77:07:FF:07		
8:32:08	DHCP	00:80:64:6C:C6:CA		

Ethernet Address (MAC)	Type	IP Address	Hostname	Description
00:06:77:07:FF:07	DHCP	192.168.1.123		

Status: Sent 192.168.1.123 to Ethernet address 00:06:77:07:FF:07
Entries: 1 of 256

Figure 21: Integration of the IP address in the BOOTP/DHCP server

After a delay the encoder appears both in **Request History** and in **Relation List** with the IP address entered.

● Green The Mod LED on the AFS60/AFM60 EtherNet/IP illuminates green continuously (the encoder now has a valid IP address).

5.2.3 Freezing the IP address assigned

Using the following procedure you can ensure that the encoder retains the IP address assigned via DHCP also after switching back on:

- Deactivate the DHCP function in the encoder.

  Set attribute 3 of the TCP/IP Interface Object to 0. You can achieve this, e.g., in Rockwell **BOOTP/DHCP Server** by clicking the **Disable BOOTP/DHCP** button.

- Then change the address switches on the encoder to the position “000” (see section 4.2.1 on page 46).

  After switching back on, the encoder starts with the previously assigned IP address saved in the non-volatile memory.

5.2.4 Checking the integration in EtherNet/IP via RSLinx-Classic

With the aid of the tool **RSLinx Classic** you can again check whether the IP address set is detected by the control system.

- Start **RSLinx Classic** (as a rule on the Start menu on your PC/notebook in **Rockwell Software**, **RSLinx**, **RSLinx Classic**).
- Click on the **RSWho** button in the program.

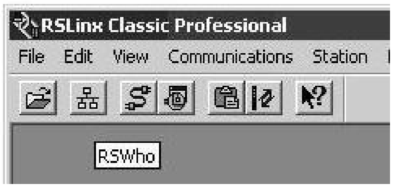
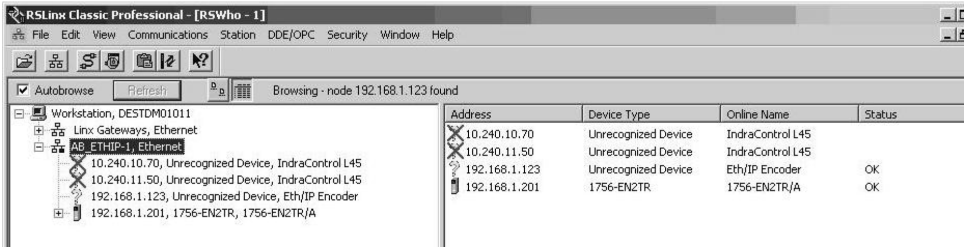


Figure 22: RSWho button in RSLinx Classic

- Then open the path AB_ETHIP-1, Ethernet.
  The encoder can be seen with its IP address.



Browsing - node 192.168.1.123 found

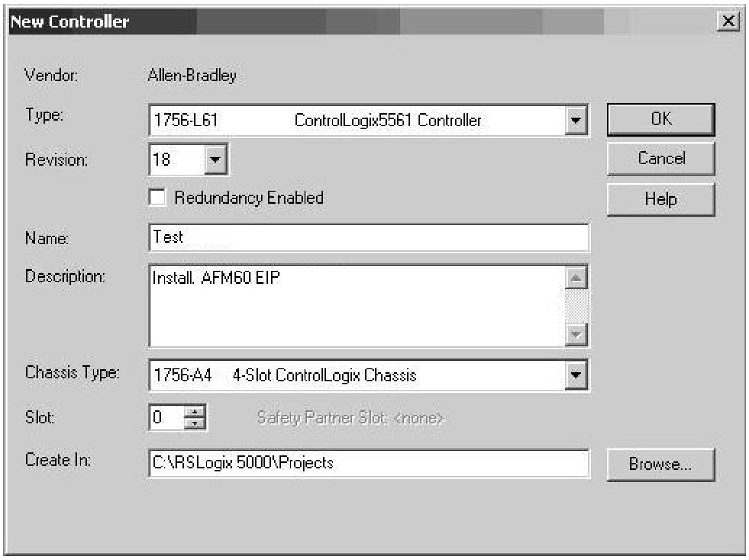
Workstation, DESTDM01011
Linx Gateways, Ethernet
AB_ETHIP-1, Ethernet
10.240.10.70, Unrecognized Device, IndraControl L45
10.240.11.50, Unrecognized Device, IndraControl L45
192.168.1.123, Unrecognized Device, Eth/IP Encoder
192.168.1.201, 1756-EN2TR, 1756-EN2TR/A

Address	Device Type	Online Name	Status
10.240.10.70	Unrecognized Device	IndraControl L45	
10.240.11.50	Unrecognized Device	IndraControl L45	
192.168.1.123	Unrecognized Device	Eth/IP Encoder	OK
192.168.1.201	1756-EN2TR	1756-EN2TR/A	OK

Figure 23: Encoder on the path AB_ETHIP-1 in RSLinx Classic

5.3 Creating a project in the controller software

- Start the control software **RSLogix 5000** (as a rule on the Start menu on your PC/notebook in **Rockwell Software**, **RSLogix 5000 Enterprise Series**, **RSLogix 5000**).
- On the **File** menu open a new project using the **New...** command.
- Configure the hardware.


New Controller

Vendor: Allen-Bradley

Type: 1756-L61 ControlLogix5561 Controller

Revision: 18

☐ Redundancy Enabled

Name: Test

Description: Install. AFM60 EIP

Chassis Type: 1756-A4 4-Slot ControlLogix Chassis

Slot: 0 Safety Partner Slot: <none>

Create In: C:\RSLogix 5000\Projects

Figure 24: Configuring the hardware

Example:

- **Type**: 1756-L61 ControlLogix5561 Controller (dependent on the controller)
- **Name**: Test (name can be selected as required)
- **Description**: Install. AFM60 EIP (can be selected as required)
- **Chassis Type**: 1756-A4 4-Slot ControlLogix Chassis (depending on the housing)
- **Create In**: storage location (can be selected as required)

▶ Click **OK**.
The **RSLogix 5000 [Name]** window will open.

NOTE

Type and Chassis Type must match your control system.

Adding communication interface

▶ In the **Controller Organizer** click **1756 Backplane, 1756-A4** using the right mouse button and select **New Module...**.

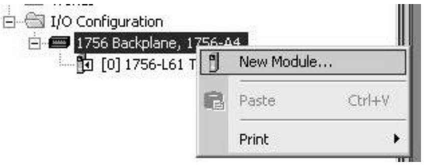
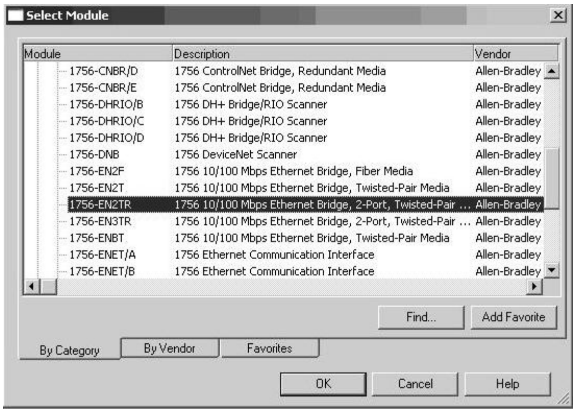


Figure 25: Adding communication interface

The **Select Module** dialog box opens.

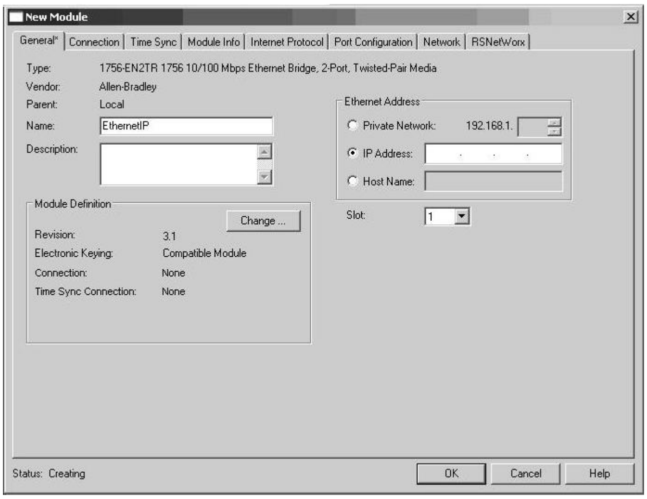
▶ In the **Select Module** dialog box select the **By Category** tab.
▶ In the tree in **Communications** select the module **1756-EN2TR**.



Module	Description	Vendor
1756-CNBR/D	1756 ControlNet Bridge, Redundant Media	Allen-Bradley
1756-CNBR/E	1756 ControlNet Bridge, Redundant Media	Allen-Bradley
1756-DHRIO/B	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DHRIO/C	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DHRIO/D	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DNB	1756 DeviceNet Scanner	Allen-Bradley
1756-EN2F	1756 10/100 Mbps Ethernet Bridge, Fiber Media	Allen-Bradley
1756-EN2T	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1756-EN2TR	1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-Pair ...	Allen-Bradley
1756-EN3TR	1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-Pair ...	Allen-Bradley
1756-ENBT	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1756-ENET/A	1756 Ethernet Communication Interface	Allen-Bradley
1756-ENET/B	1756 Ethernet Communication Interface	Allen-Bradley

Figure 26: Selecting communication interface

- ▶ Click **OK**.
  The **New Module** dialog box will open.
- ▶ On the **General** tab assign a name in the **Name** field, in the **IP Address** field the IP address, and select the **Slot**.



Type: 1756-EN2TR 1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-Pair Media
Vendor: Allen-Bradley
Parent: Local
Name: EthernetIP
Description:

Module Definition
Revision: 3.1
Electronic Keying: Compatible Module
Connection: None
Time Sync Connection: None

Ethernet Address
○ Private Network: 192.168.1.
● IP Address: . . .
○ Host Name:

Slot: 1

Status: Creating

Figure 27: Name of the communication interface

- ▶ Click **OK**.
  In **Controller Organizer** in **1756 Backplane, 1756-A4** the selected module **1756-EN2TR** [with name] appears along with the symbol for **Ethernet**.

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

- as Generic Modules (see section 5.4 on page 53)
- with the aid of an EDS file (see section 5.5 on page 56)
- with the aid of the function block (see section 5.7 on page 71)

5.4 Integration of the encoder as a generic module

- Using the right mouse button click the **Ethernet** symbol and select the **New Module...** command.

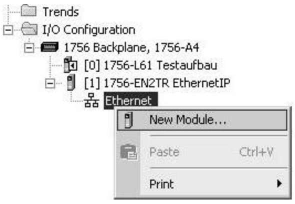
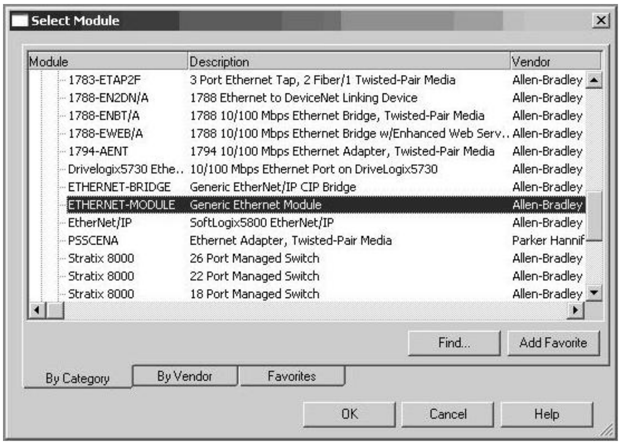


Figure 28: Integrating encoder

The **Select Module** dialog box opens.

- In the **Select Module** dialog box select the **By Category** tab.
- Open the **Communication** tree.
- In the tree in **Communications** select the module **ETHERNET-MODULE (Generic Ethernet Module)**.



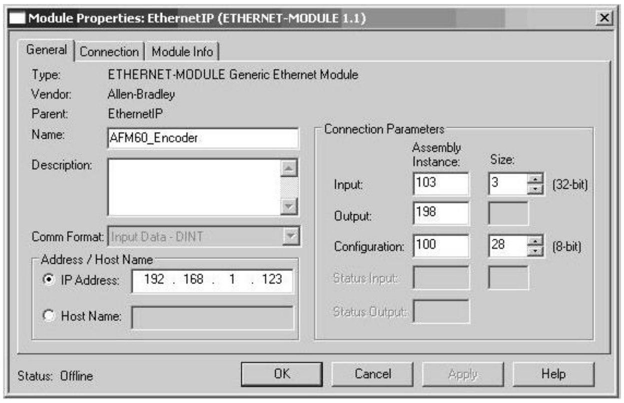
Module	Description	Vendor
1783-ETAP2F	3 Port Ethernet Tap, 2 Fiber/1 Twisted-Pair Media	Allen-Bradley
1788-EN2DN/A	1788 Ethernet to DeviceNet Linking Device	Allen-Bradley
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv..	Allen-Bradley
1794-AENT	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
Drivelogix5730 Ethe..	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley
PSSCENA	Ethernet Adapter, Twisted-Pair Media	Parker Hannif
Stratix 8000	26 Port Managed Switch	Allen-Bradley
Stratix 8000	22 Port Managed Switch	Allen-Bradley
Stratix 8000	18 Port Managed Switch	Allen-Bradley

Figure 29: Selecting module

- Click **OK**.
  The **Module Properties [module name]** dialog box will open.

5.4.1 Module settings

- ▶ In the **Modules Properties [module name]** dialog box enter the **IP address** assigned to the encoder (see section 5.2 on page 48).
- ▶ Enter the settings for **Input**, **Output**, as well as **Configuration**.



Module Properties: EthernetIP (ETHERNET-MODULE 1.1)

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: EthernetIP
Name: AFM60_Encoder
Description:
Comm Format: Input Data - DINT

Address / Host Name
● IP Address: 192 . 168 . 1 . 123
○ Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:	103	3	(32-bit)
Output:	198		
Configuration:	100	28	(8-bit)
Status Input:			
Status Output:			

Status: Offline

Figure 30: Entering module properties

Example:

- **Name**: AFM60_Encoder (name can be selected as required)
- **Comm Format**: Input Data – DINT
- **IP Address**: 192.168.1.123
- **Input**: Assembly Instance: 103; Size: 3
  In this way instance 103 of the Assembly Object is selected (see Table 15 on page 23). The size is 3 × 32 Bit (= 12 Byte)
- **Output**: Assembly Instance: 198 [16)]
- **Configuration**: Assembly Instance: 100; Size: 28
  In this way instance 100 of the Assembly Object is selected (see Table 15 on page 23). The size is 28 × 8 Bit (= 28 Byte).

NOTE

Instance 100 of the Assembly Object represents the configuration assembly. If this assembly is opened, it must never be empty. It is imperative you fill the configuration assembly with data first (see Table 17 on page 27). Otherwise in some circumstances the control system may output an error (see section 7.3.4 on page 105).

- ▶ Click **OK**.

Example data for a configuration assembly

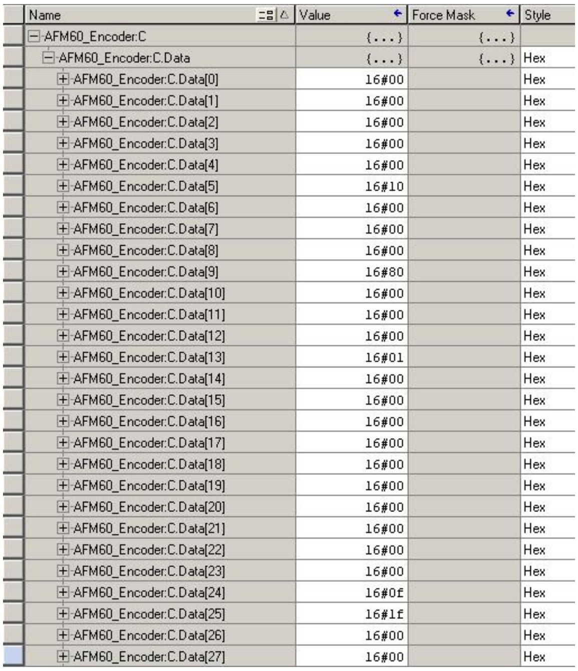
The data for the configuration assembly are transferred in the 28 bytes of instance 100 configured previously (see Table 17 on page 27).

You can see these data in **Controller Tags** in the **Name** column in the **AFM60_Encoder:C**, **AFM60_Encoder:C.Data** item.

NOTE

The low byte is displayed before the high byte.

[16)] As the encoder does not process an output assembly, the parameter output is set to 198 (Input only).



Name	Value	Force Mask	Style
AFM60_Encoder:C	{...}	{...}	
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

Figure 31: 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
- velocity format = 1F0Fh
  C.Data[24] 0Fh and **C.Data[25] 1Fh**

5.4.2 Download the configuration to the control system

▶ Load the configuration to the control system.

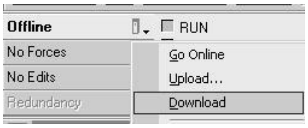


Figure 32: Loading configuration

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

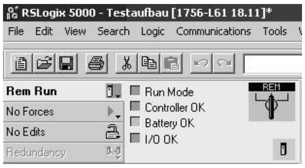
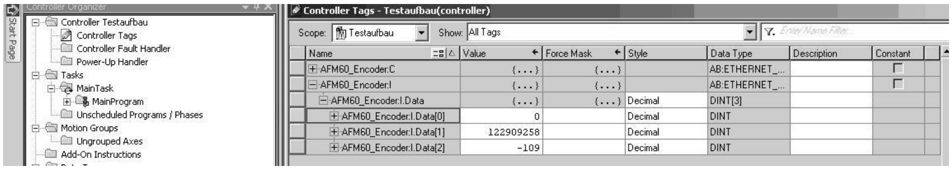

RSLogix 5000 - Testaufbau [1756-L61 18.11]*
File Edit View Search Logic Communications Tools
Rem Run
No Forces
No Edits
Redundancy
Run Mode
Controller OK
Battery OK
I/O OK
REM

Figure 33: Communication status

5.4.3 Checking the communication

To check the communication between control system and encoder, the data the control system receives from the encoder can be displayed.



Scope: Testaufbau Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description	Constant
AFM60_Encoder:C	{...}	{...}		AB:ETHERNET_...		□
AFM60_Encoder:I	{...}	{...}		AB:ETHERNET_...		□
AFM60_Encoder:I.Data	{...}	{...}	Decimal	DINT[3]		
AFM60_Encoder:I.Data[0]	0		Decimal	DINT		
AFM60_Encoder:I.Data[1]	122909258		Decimal	DINT		
AFM60_Encoder:I.Data[2]	-109		Decimal	DINT		

Figure 34: Checking the communication

- ▶ In the **Controller Organizer** open the **Controller Testaufbau** folder, **Controller Tags**.
- ▶ In the **Controller Tags** in the **Name** column open the **AFM60_Encoder:I**, **AFM60_Encoder:I.Data** item.

Displayed data in the example in Figure 34:

- **AFM60_Encoder:I.Data[0]**: Fault header: 0
- **AFM60_Encoder:I.Data[1]**: Position: 122909258
- **AFM60_Encoder:I.Data[2]**: Velocity: –109 turns/min

5.5 Integration and configuration with the aid of an EDS file

The EDS file (electronic data sheet) contains all the information related to the parameters as well as the operating modes of the AFS60/AFM60 EtherNet/IP. With the aid of the EDS file you can configure and place in operation the AFS60/AFM60 EtherNet/IP.

5.5.1 Prerequisites

- You are using an Allen-Bradley control system with control software "RSLogix 5000" from V22 (or another control system that facilitates integration with the aid of an EDS file).
- The encoder is integrated into the EtherNet/IP network (see section 5.2 on page 48).
- The EDS file is integrated in the control software via the Rockwell Hardware Installation Tool.

5.5.2 Establishing communication

- ▶ Using the right mouse button click the **Ethernet** symbol and select the **New Module...** command.

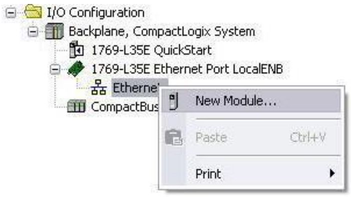
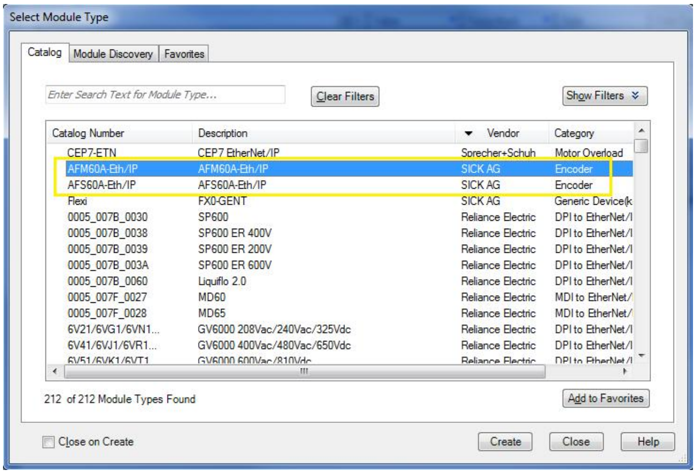


Figure 35: Integrating encoder using EDS

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

▶ Choose the corresponding encoder type on the **Catalog** tab.



Catalog Number	Description	Vendor	Category
CEP7-ETN	CEP7 EtherNet/IP	Sprecher+Schuh	Motor Overload
AFM60A-Eth/IP	AFM60A-Eth/IP	SICK AG	Encoder
AFS60A-Eth/IP	AFS60A-Eth/IP	SICK AG	Encoder
Flexi	FX0-GENT	SICK AG	Generic Device(k
0005_007B_0030	SP600	Reliance Electric	DPI to EtherNet/I
0005_007B_0038	SP600 ER 400V	Reliance Electric	DPI to EtherNet/I
0005_007B_0039	SP600 ER 200V	Reliance Electric	DPI to EtherNet/I
0005_007B_003A	SP600 ER 600V	Reliance Electric	DPI to EtherNet/I
0005_007B_0060	Liquiflo 2.0	Reliance Electric	DPI to EtherNet/I
0005_007F_0027	MD60	Reliance Electric	MDI to EtherNet/
0005_007F_0028	MD65	Reliance Electric	MDI to EtherNet/
6V21/6VG1/6VN1...	GV6000 208Vac/240Vac/325Vdc	Reliance Electric	DPI to EtherNet/I
6V41/6VJ1/6VR1...	GV6000 400Vac/480Vac/650Vdc	Reliance Electric	DPI to EtherNet/I
6V51/6VK1/6VT1	GV6000 600Vac/810Vdc	Reliance Electric	DPI to EtherNet/I

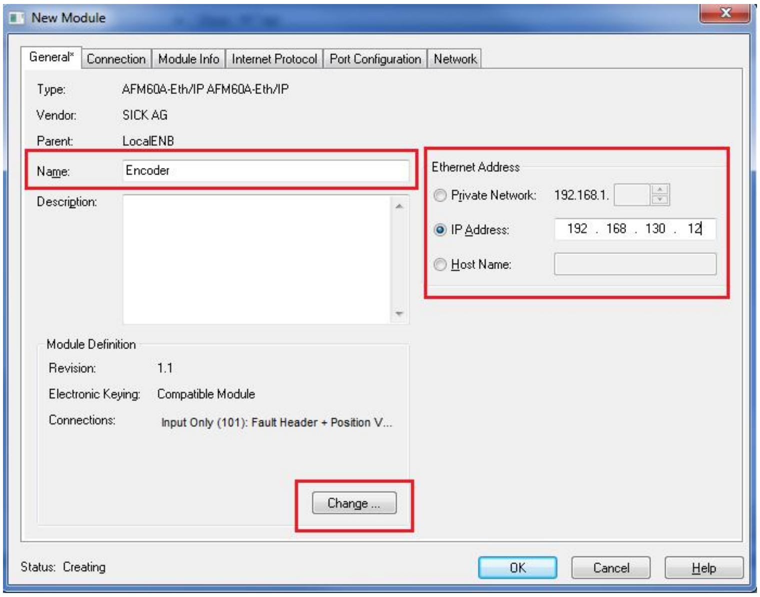
212 of 212 Module Types Found

Figure 36: Selecting module

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

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

▶ Click **OK**.
The **Module Properties [module name]** dialog box will open.



Type: AFM60A-Eth/IP AFM60A-Eth/IP
Vendor: SICK AG
Parent: LocalENB
Name: Encoder
Description:

Ethernet Address
○ Private Network: 192.168.1.
● IP Address: 192 . 168 . 130 . 12
○ Host Name:

Module Definition
Revision: 1.1
Electronic Keying: Compatible Module
Connections: Input Only (101): Fault Header + Position V...

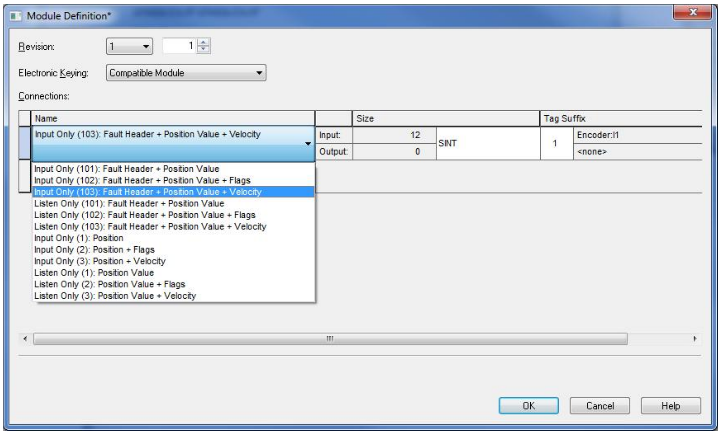
Status: Creating

Figure 37: Entering module properties

▶ In the **Name** field enter a name (can be selected as required) and enter the IP address defined for the encoder in the **IP Address** field (see section 5.2 on page 48).

In the Module Definition group box the default connection **Input Only (101)** is displayed in **Connections**. This is instance 101 of the Assembly Object (see Table 15 on page 23).

▶ If you want to change this instance, click **Change...**.



Module Definition*

Revision: 1 1

Electronic Keying: Compatible Module

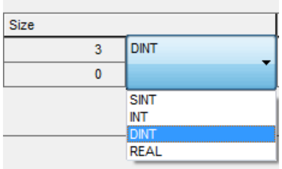
Connections:

Name		Size		Tag Suffix	
Input Only (103): Fault Header + Position Value + Velocity	Input:	12	SINT	1	Encoder:I1
	Output:	0			<none>

Input Only (101): Fault Header + Position Value
Input Only (102): Fault Header + Position Value + Flags
Input Only (103): Fault Header + Position Value + Velocity
Listen Only (101): Fault Header + Position Value
Listen Only (102): Fault Header + Position Value + Flags
Listen Only (103): Fault Header + Position Value + Velocity
Input Only (1): Position
Input Only (2): Position + Flags
Input Only (3): Position + Velocity
Listen Only (1): Position Value
Listen Only (2): Position Value + Flags
Listen Only (3): Position Value + Velocity
OK
Cancel
Help

Figure 38: Changing connections

▶ Select e.g. Input Only 103. This instance contains errors, the position value and the velocity for the encoder.



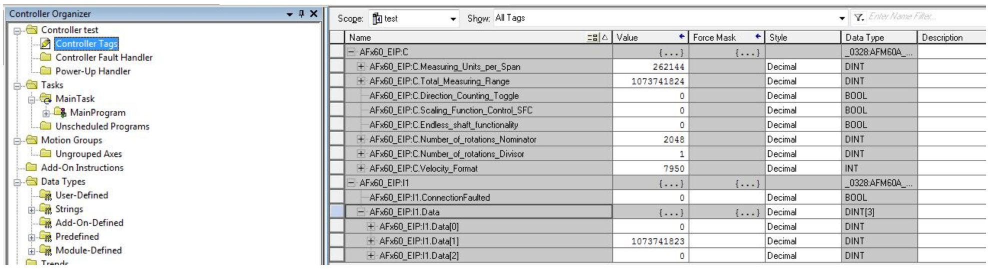
Size	
3	DINT
0	

Figure 39: Changing data format

▶ In **Size** choose the data format **DINT**.
▶ Then click on **OK**.

Checking the communication

To check the communication between control system and encoder, the data the control system receives from the encoder can be displayed.



Scope: test Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description
AFx60_EIP:C	{...}	{...}		_0328:AFM60A_...	
AFx60_EIP:C.Measuring_Units_per_Span	262144		Decimal	DINT	
AFx60_EIP:C.Total_Measuring_Range	1073741824		Decimal	DINT	
AFx60_EIP:C.Direction_Counting_Toggle	0		Decimal	BOOL	
AFx60_EIP:C.Scaling_Function_Control_SFC	0		Decimal	BOOL	
AFx60_EIP:C.Endless_shaft_functionality	0		Decimal	BOOL	
AFx60_EIP:C.Number_of_rotations_Nominator	2048		Decimal	DINT	
AFx60_EIP:C.Number_of_rotations_Divisor	1		Decimal	DINT	
AFx60_EIP:C.Velocity_Format	7950		Decimal	INT	
AFx60_EIP:I1	{...}	{...}		_0328:AFM60A_...	
AFx60_EIP:I1.ConnectionFaulted	0		Decimal	BOOL	
AFx60_EIP:I1.Data	{...}	{...}	Decimal	DINT[3]	
AFx60_EIP:I1.Data[0]	0		Decimal	DINT	
AFx60_EIP:I1.Data[1]	1073741823		Decimal	DINT	
AFx60_EIP:I1.Data[2]	0		Decimal	DINT	

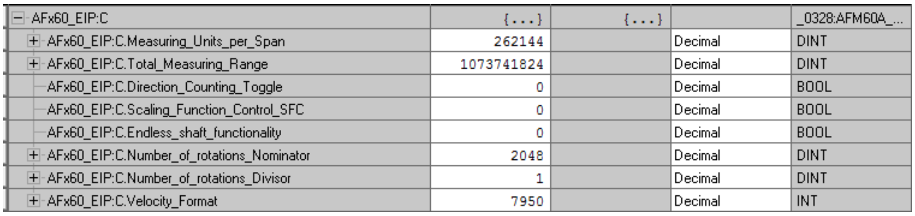
Figure 40: Checking the communication

▶ In the **Controller Organizer** open the **Controller test** folder, **Controller Tags**.
▶ In the **Controller Tags** in the **Name** column open the point **AFx60_EIP:I1**, **AFx60_EIP:I1.Data**.

Displayed data in the example:

- **AFx60_EIP:I1.Data[0]**: Fault header: 0
- **AFx60_EIP:I1.Data[1]**: Position: 1073741823
- **AFx60_EIP:I1.Data[2]**: Velocity: 0

5.5.3 Configuration



AFx60_EIP:C	{...}	{...}		_0328:AFM60A_...
AFx60_EIP:C.Measuring_Units_per_Span	262144		Decimal	DINT
AFx60_EIP:C.Total_Measuring_Range	1073741824		Decimal	DINT
AFx60_EIP:C.Direction_Counting_Toggle	0		Decimal	BOOL
AFx60_EIP:C.Scaling_Function_Control_SFC	0		Decimal	BOOL
AFx60_EIP:C.Endless_shaft_functionality	0		Decimal	BOOL
AFx60_EIP:C.Number_of_rotations_Nominator	2048		Decimal	DINT
AFx60_EIP:C.Number_of_rotations_Divisor	1		Decimal	DINT
AFx60_EIP:C.Velocity_Format	7950		Decimal	INT

Figure 41: Configuration of the encoder

- ▶ In the **Controller Tags** in the **Name** column open the point **AFx60_EIP:C**
- ▶ Enter the parameters for the encoder (see section 3.6 “Configurable functions” on page 37).

5.6 Installation of the ladder routine

Two so-called ladder routines are available to integrate the web server. The configuration data are mapped between the control system and web server with the aid of the ladder routine.

Use the following ladder routine depending on the instance selected:

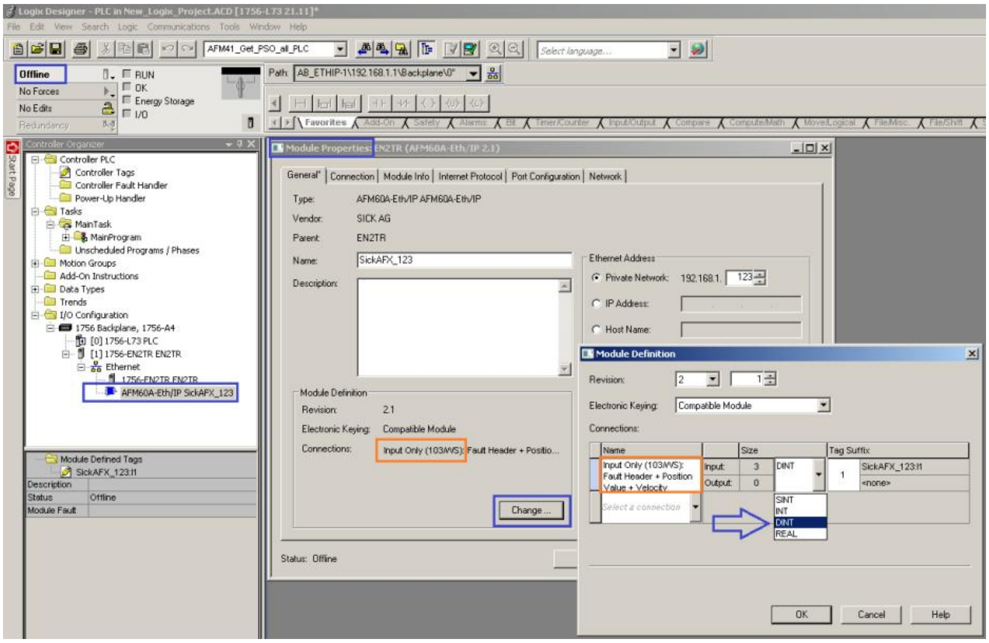
- SickAFx_A101WS_A103WS_FB_Enc1_GetSet.L5X for the instances 101WS and 103WS

  or

- SickAFx_A102WS_FB_Enc1_GetSet.L5X for the instance 102WS

Prerequisites for the installation of the ladder routine are:

- the installation file for the ladder routine that you can download from the encoder web server (see section 6.4.2 on page 97)
- the correct installation of the current EDS file (see section 5.5 on page 56)
- the selection of instance 101WS, 102WS or 103WS in the configuration of the encoder module



Module Properties: EN2TR (AFM60A-Eth/IP 2.1)

Type: AFM60A-Eth/IP AFM60A-Eth/IP

Vendor: SICK AG

Parent: EN2TR

Name: SickAFX_123

Ethernet Address: ● Private Network: 192.168.1. 123

Module Definition

Revision: 2.1

Electronic Keying: Compatible Module

Connections: Input Only (103WS): Fault Header + Positio...

Status: Offline

Module Definition

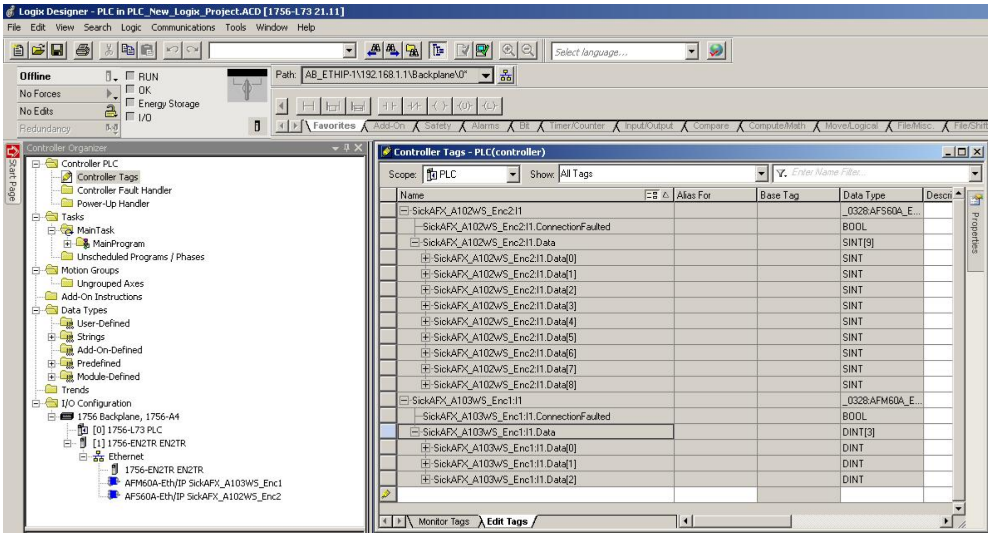
Revision: 2 1

Electronic Keying: Compatible Module

Name	Size			Tag Suffix
Input Only (103WS): Fault Header + Position Value + Velocity	Input	3	DINT	SickAFX_123:I1
	Output	0		<none>

Figure 42: Selection of language (in the example 103WS)

- a correctly configured project with the AFS60/AFM60 EtherNet/IP in the “RSLogix 5000”



Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]

Offline
No Forces
No Edits
Redundancy

□ RUN
□ OK
□ Energy Storage
□ I/O

Path: AB_ETHIP-1\192.168.1.1\Backplane\0*

Controller Organizer

- Controller PLC
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
  - MainTask
    - MainProgram
  - Unscheduled Programs / Phases
- Motion Groups
  - Ungrouped Axes
- Add-On Instructions
- Data Types
  - User-Defined
  - Strings
  - Add-On-Defined
  - Predefined
  - Module-Defined
- Trends
- I/O Configuration
  - 1756 Backplane, 1756-A4
    - [0] 1756-L73 PLC
    - [1] 1756-EN2TR EN2TR
      - Ethernet
        - 1756-EN2TR EN2TR
        - AFM60A-Eth/IP SickAFX_A103WS_Enc1
        - AFS60A-Eth/IP SickAFX_A102WS_Enc2

Controller Tags - PLC(controller)

Scope: PLC Show: All Tags Enter Name Filter...

Name	Alias For	Base Tag	Data Type	Descri
SickAFX_A102WS_Enc2:I1			_0328:AFS60A_E...	
SickAFX_A102WS_Enc2:I1.ConnectionFaulted			BOOL	
SickAFX_A102WS_Enc2:I1.Data			SINT[9]	
SickAFX_A102WS_Enc2:I1.Data[0]			SINT	
SickAFX_A102WS_Enc2:I1.Data[1]			SINT	
SickAFX_A102WS_Enc2:I1.Data[2]			SINT	
SickAFX_A102WS_Enc2:I1.Data[3]			SINT	
SickAFX_A102WS_Enc2:I1.Data[4]			SINT	
SickAFX_A102WS_Enc2:I1.Data[5]			SINT	
SickAFX_A102WS_Enc2:I1.Data[6]			SINT	
SickAFX_A102WS_Enc2:I1.Data[7]			SINT	
SickAFX_A102WS_Enc2:I1.Data[8]			SINT	
SickAFX_A103WS_Enc1:I1			_0328:AFM60A_E...	
SickAFX_A103WS_Enc1:I1.ConnectionFaulted			BOOL	
SickAFX_A103WS_Enc1:I1.Data			DINT[3]	
SickAFX_A103WS_Enc1:I1.Data[0]			DINT	
SickAFX_A103WS_Enc1:I1.Data[1]			DINT	
SickAFX_A103WS_Enc1:I1.Data[2]			DINT	

Figure 43: Correctly configured project with two encoder modules

The following steps must be undertaken:

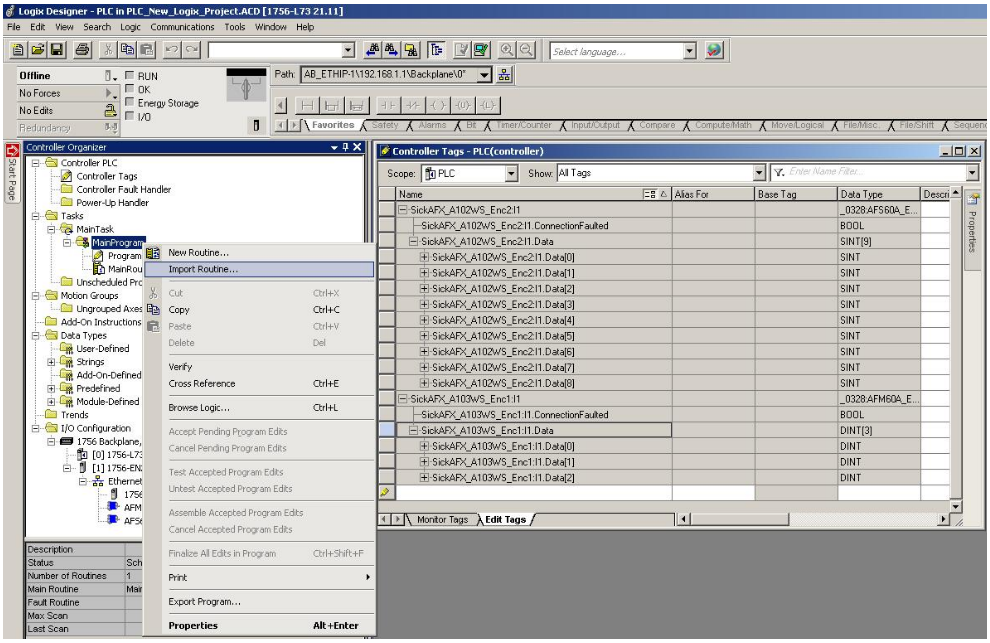
- The ladder routine must be imported and a few parameters must be configured during the import.
- The ladder routine must be integrated in the MainRoutine of your project as a SubRoutine.
- Then the encoder can be configured both from the control system (in the Controller Tags) and with the aid of the web server.

NOTE

If you use several encoders, you must import the routine several times and give it a dedicated unique so-called **Final Name** during the import. You must also uniquely name the **Tag References** for each encoder.

5.6.1 Importing the ladder routine

- From the **MainProgram** context menu select the **Import Routine...** command.



Name	Alias For	Base Tag	Data Type	Descri
SickAFX_A102WS_Enc2:I1			_0328:AFS60A_E...	
SickAFX_A102WS_Enc2:I1.ConnectionFaulted			BOOL	
SickAFX_A102WS_Enc2:I1.Data			SINT[9]	
SickAFX_A102WS_Enc2:I1.Data[0]			SINT	
SickAFX_A102WS_Enc2:I1.Data[1]			SINT	
SickAFX_A102WS_Enc2:I1.Data[2]			SINT	
SickAFX_A102WS_Enc2:I1.Data[3]			SINT	
SickAFX_A102WS_Enc2:I1.Data[4]			SINT	
SickAFX_A102WS_Enc2:I1.Data[5]			SINT	
SickAFX_A102WS_Enc2:I1.Data[6]			SINT	
SickAFX_A102WS_Enc2:I1.Data[7]			SINT	
SickAFX_A102WS_Enc2:I1.Data[8]			SINT	
SickAFX_A103WS_Enc1:I1			_0328:AFM60A_E...	
SickAFX_A103WS_Enc1:I1.ConnectionFaulted			BOOL	
SickAFX_A103WS_Enc1:I1.Data			DINT[3]	
SickAFX_A103WS_Enc1:I1.Data[0]			DINT	
SickAFX_A103WS_Enc1:I1.Data[1]			DINT	
SickAFX_A103WS_Enc1:I1.Data[2]			DINT	

Figure 44: Selection of the Import Routine... command

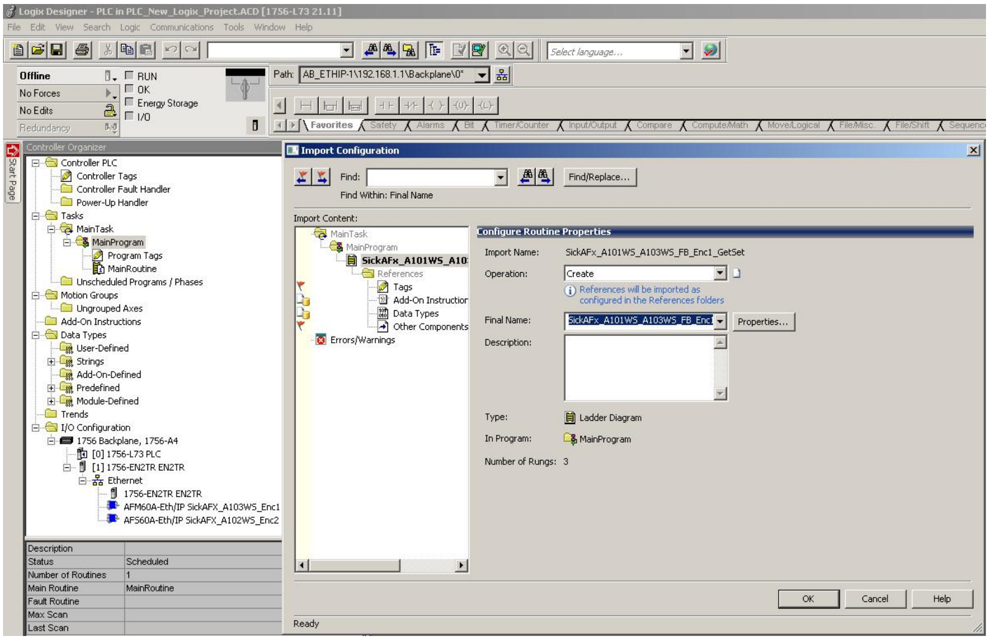
You must select the appropriate ladder routine depending on whether you use the instance 101WS and 103WS or the instance 102WS of the Assembly Object (see Table 15 on page 23).

- ▶ Select the file **SickAFx_A101WS_A103WS_FB_Enc1_GetSet.L5X** or the file **SickAFx_A102WS_FB_Enc1_GetSet.L5X** and click **Import...**.

  The **Import Configuration** dialog box will open.

NOTE

- ▶ Only click **OK** once all configuration steps for the import have been completed. If you inadvertently click OK, then you must restart the import as in Figure 43 on page 60.



Path: AB_ETHIP-1\192.168.1.1\Backplane\0*

Description	
Status	Scheduled
Number of Routines	1
Main Routine	MainRoutine
Fault Routine	
Max Scan	
Last Scan	

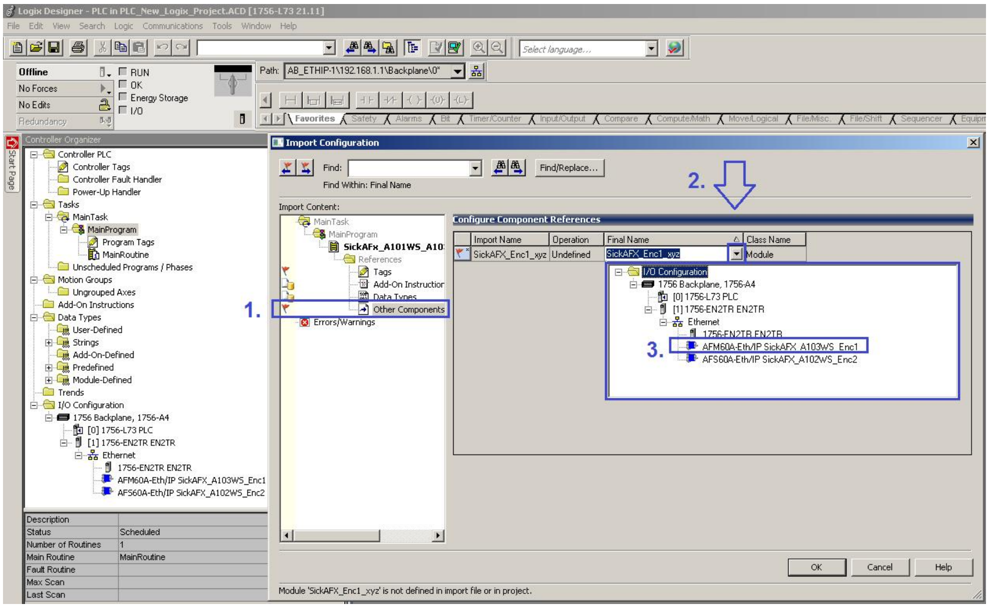
Find:
Find Within: Final Name

Import Name: SickAFx_A101WS_A103WS_FB_Enc1_GetSet
Operation: Create
References will be imported as configured in the References folders
Final Name: SickAFx_A101WS_A103WS_FB_Enc1
Description:
Type: Ladder Diagram
In Program: MainProgram
Number of Rungs: 3

Figure 45: Dialog Import Configuration

- ▶ If necessary change the name of the routine in the **Final Name** field.
  If you integrate several encoders into your project, then you must assign a unique final name to the routine for each encoder.

- Choose the point **Other Components**.
- In the **Final Name** column open the list box.
- Choose the encoder module for which you want to import the ladder routine.



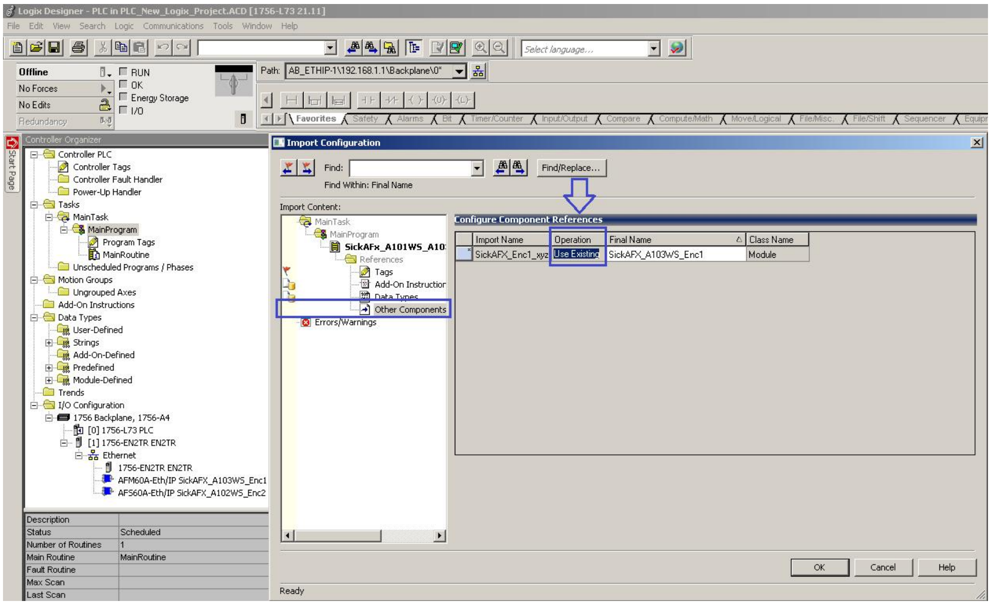
Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]
Import Configuration
Find:
Find Within: Final Name
Import Content:
Configure Component References
1.
2.
3.
Other Components
I/O Configuration
AFM60A-Eth/IP SickAFX_A103WS_Enc1
AFS60A-Eth/IP SickAFX_A102WS_Enc2
OK
Cancel
Help

Import Name	Operation	Final Name	Class Name
SickAFX_Enc1_xyz	Undefined	SickAFX_Enc1_xyz	Module

Module 'SickAFX_Enc1_xyz' is not defined in import file or in project.

Figure 46: Selection of the encoder

- In the **Operation** column choose the option **Use Existing**.



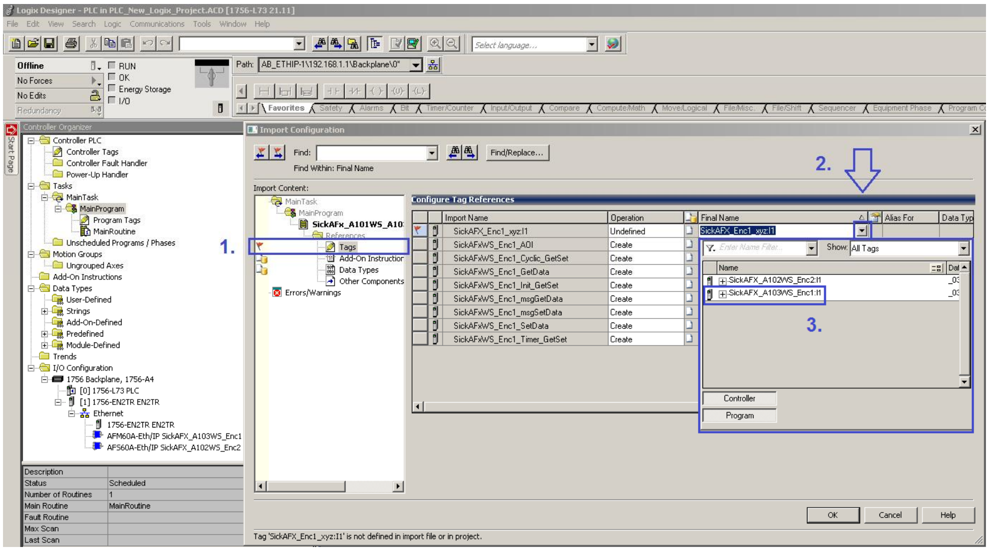
Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]
Import Configuration
Find:
Find Within: Final Name
Import Content:
Configure Component References
Other Components
OK
Cancel
Help

Import Name	Operation	Final Name	Class Name
SickAFX_Enc1_xyz	Use Existing	SickAFX_A103WS_Enc1	Module

Ready

Figure 47: Selection of operation for the component

- ▶ Choose in **Import Content** the point **Tags**.
- ▶ In the **Final Name** column open the list box.
- ▶ Choose the encoder module for which you want to modify the tags.



Path: AB_ETHIP-1\192.168.1.1\Backplane\0

Configure Tag References

Import Name	Operation	Final Name	Alias For	Data Typ
SickAFX_Enc1_xyz:I1	Undefined	SickAFX_Enc1_xyz:I1		
SickAFxWS_Enc1_AOI	Create			
SickAFxWS_Enc1_Cyclic_GetSet	Create			
SickAFxWS_Enc1_GetData	Create			
SickAFxWS_Enc1_Init_GetSet	Create			
SickAFxWS_Enc1_msgGetData	Create			
SickAFxWS_Enc1_msgSetData	Create			
SickAFxWS_Enc1_SetData	Create			
SickAFxWS_Enc1_Timer_GetSet	Create			

Show: All Tags

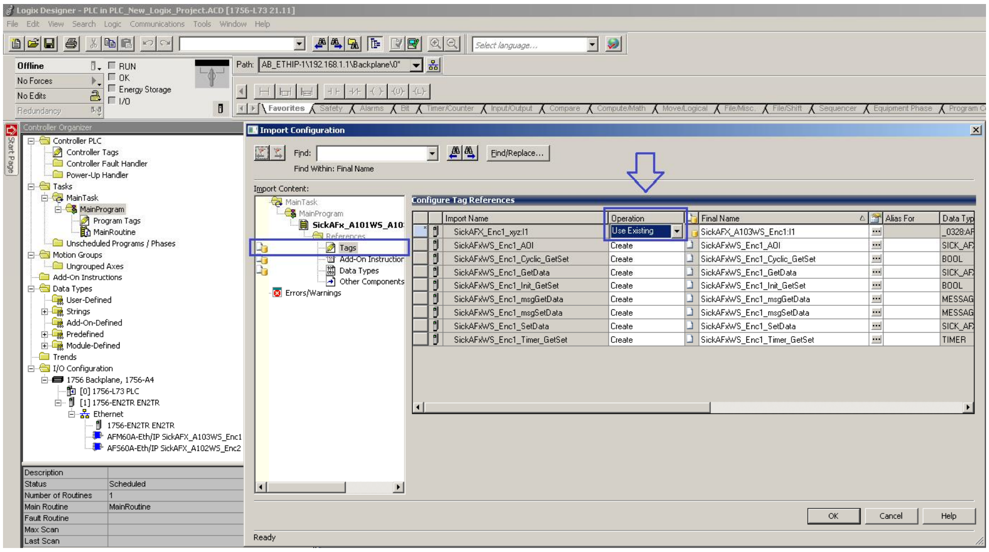
Name	Dat
SickAFX_A102WS_Enc2:I1	_03
SickAFX_A103WS_Enc1:I1	_03

Description / Status: Scheduled / Number of Routines: 1 / Main Routine: MainRoutine / Fault Routine / Max Scan / Last Scan

Tag 'SickAFX_Enc1_xyz:I1' is not defined in import file or in project.

Figure 48: Selection of the tags for the instance used

- ▶ In the **Operation** column choose the option **Use Existing**.



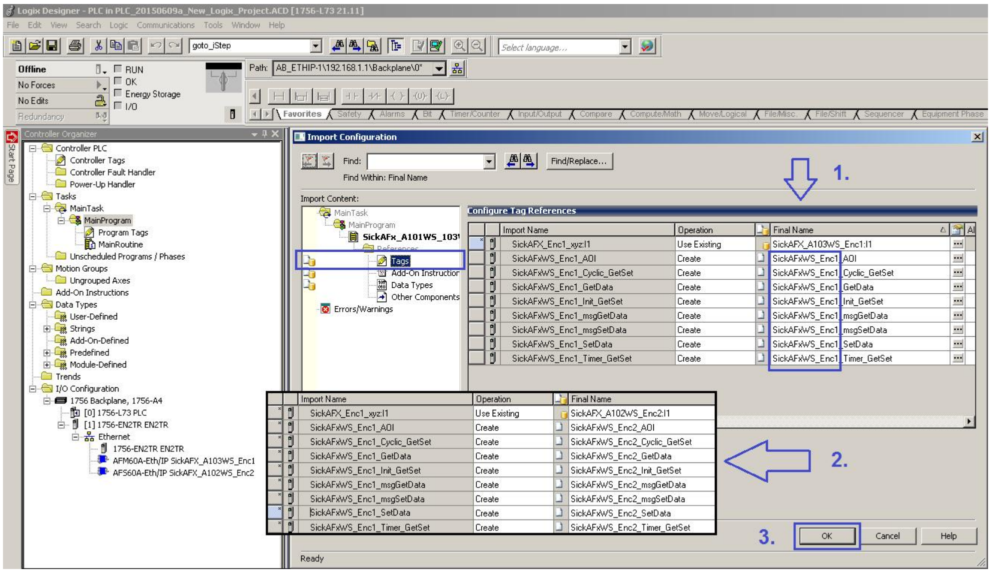
Configure Tag References

Import Name	Operation	Final Name	Alias For	Data Typ
SickAFX_Enc1_xyz:I1	Use Existing	SickAFX_A103WS_Enc1:I1		_0328:AF
SickAFxWS_Enc1_AOI	Create	SickAFxWS_Enc1_AOI		SICK_AF
SickAFxWS_Enc1_Cyclic_GetSet	Create	SickAFxWS_Enc1_Cyclic_GetSet		BOOL
SickAFxWS_Enc1_GetData	Create	SickAFxWS_Enc1_GetData		SICK_AF
SickAFxWS_Enc1_Init_GetSet	Create	SickAFxWS_Enc1_Init_GetSet		BOOL
SickAFxWS_Enc1_msgGetData	Create	SickAFxWS_Enc1_msgGetData		MESSAG
SickAFxWS_Enc1_msgSetData	Create	SickAFxWS_Enc1_msgSetData		MESSAG
SickAFxWS_Enc1_SetData	Create	SickAFxWS_Enc1_SetData		SICK_AF
SickAFxWS_Enc1_Timer_GetSet	Create	SickAFxWS_Enc1_Timer_GetSet		TIMER

Ready

Figure 49: Selection of the operation for the tag references

- ▶ If necessary, in the **Final Name** column change the name of the **Tags**.
  If you use several encoders in a project, then each final name is only allowed to be assigned once. For example change the names from "...Enc1..." to "...Enc2...".



Import Configuration

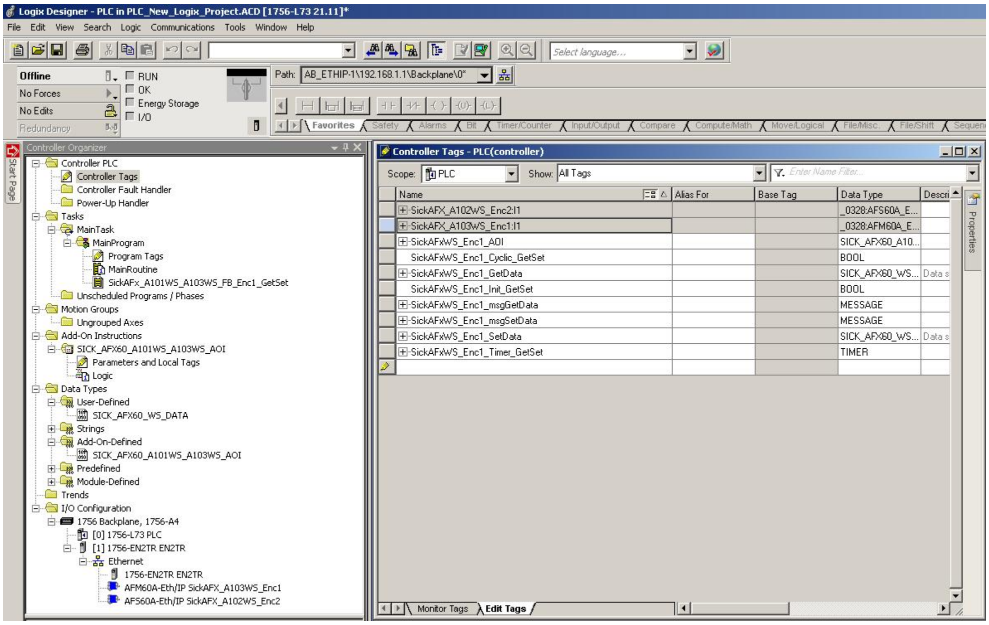
Configure Tag References

Import Name	Operation	Final Name
SickAFX_Enc1_xyz:I1	Use Existing	SickAFX_A103WS_Enc1:I1
SickAFxWS_Enc1_AOI	Create	SickAFxWS_Enc1_AOI
SickAFxWS_Enc1_Cyclic_GetSet	Create	SickAFxWS_Enc1_Cyclic_GetSet
SickAFxWS_Enc1_GetData	Create	SickAFxWS_Enc1_GetData
SickAFxWS_Enc1_Init_GetSet	Create	SickAFxWS_Enc1_Init_GetSet
SickAFxWS_Enc1_msgGetData	Create	SickAFxWS_Enc1_msgGetData
SickAFxWS_Enc1_msgSetData	Create	SickAFxWS_Enc1_msgSetData
SickAFxWS_Enc1_SetData	Create	SickAFxWS_Enc1_SetData
SickAFxWS_Enc1_Timer_GetSet	Create	SickAFxWS_Enc1_Timer_GetSet

Import Name	Operation	Final Name
SickAFX_Enc1_xyz:I1	Use Existing	SickAFX_A102WS_Enc2:I1
SickAFxWS_Enc1_AOI	Create	SickAFxWS_Enc2_AOI
SickAFxWS_Enc1_Cyclic_GetSet	Create	SickAFxWS_Enc2_Cyclic_GetSet
SickAFxWS_Enc1_GetData	Create	SickAFxWS_Enc2_GetData
SickAFxWS_Enc1_Init_GetSet	Create	SickAFxWS_Enc2_Init_GetSet
SickAFxWS_Enc1_msgGetData	Create	SickAFxWS_Enc2_msgGetData
SickAFxWS_Enc1_msgSetData	Create	SickAFxWS_Enc2_msgSetData
SickAFxWS_Enc1_SetData	Create	SickAFxWS_Enc2_SetData
SickAFxWS_Enc1_Timer_GetSet	Create	SickAFxWS_Enc2_Timer_GetSet

Figure 50: Changing the tag names

- ▶ Click **OK**.
- ▶ The ladder routine is imported.



Controller Tags - PLC(controller)

Scope: PLC Show: All Tags

Name	Alias For	Base Tag	Data Type	Descri
SickAFX_A102WS_Enc2:I1			_0328:AFS60A_E...	
SickAFX_A103WS_Enc1:I1			_0328:AFM60A_E...	
SickAFxWS_Enc1_AOI			SICK_AFX60_A10...	
SickAFxWS_Enc1_Cyclic_GetSet			BOOL	
SickAFxWS_Enc1_GetData			SICK_AFX60_WS...	Data s
SickAFxWS_Enc1_Init_GetSet			BOOL	
SickAFxWS_Enc1_msgGetData			MESSAGE	
SickAFxWS_Enc1_msgSetData			MESSAGE	
SickAFxWS_Enc1_SetData			SICK_AFX60_WS...	Data s
SickAFxWS_Enc1_Timer_GetSet			TIMER	

Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]*
Controller Organizer
SickAFx_A101WS_A103WS_FB_Enc1_GetSet
SICK_AFX60_A101WS_A103WS_AOI
SICK_AFX60_WS_DATA
AFM60A-Eth/IP SickAFX_A103WS_Enc1
AFS60A-Eth/IP SickAFX_A102WS_Enc2
Monitor Tags
Edit Tags

Figure 51: Project structure after the import

5.6.2 Integrating in the MainRoutine as a SubRoutine

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

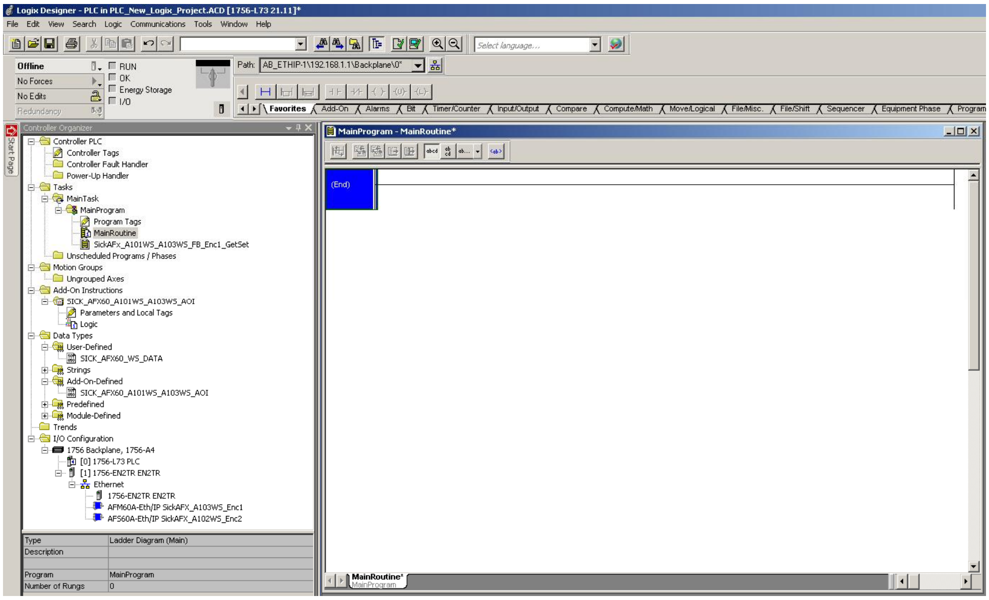
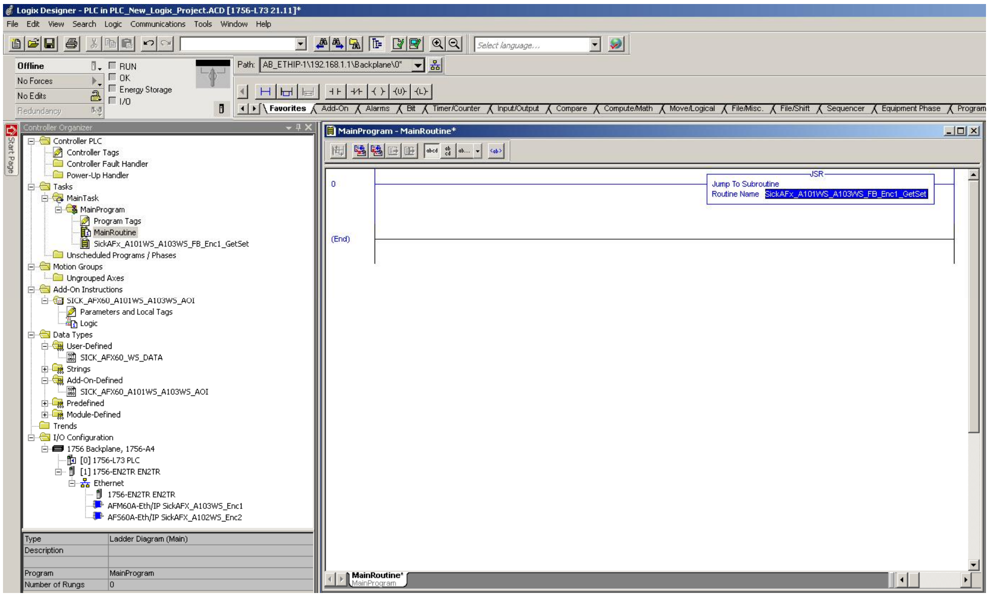

Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]*
File Edit View Search Logic Communications Tools Window Help
Offline
No Forces
No Edits
Redundancy
RUN
OK
Energy Storage
I/O
Path: AB_ETHIP-1\192.168.1.1\Backplane\0*
Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Sequencer Equipment Phase Program
Controller Organizer
Controller PLC
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Program Tags
MainRoutine
SickAFx_A101WS_A103WS_FB_Enc1_GetSet
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
SICK_AFX60_A101WS_A103WS_AOI
Parameters and Local Tags
Logic
Data Types
User-Defined
SICK_AFX60_WS_DATA
Strings
Add-On-Defined
SICK_AFX60_A101WS_A103WS_AOI
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A4
[0] 1756-L73 PLC
[1] 1756-EN2TR EN2TR
Ethernet
1756-EN2TR EN2TR
AFM60A-Eth/IP SickAFX_A103WS_Enc1
AFS60A-Eth/IP SickAFX_A102WS_Enc2
Type Ladder Diagram (Main)
Description
Program MainProgram
Number of Rungs 0
MainProgram - MainRoutine*
(End)
MainRoutine*
MainProgram

Figure 52: MainRoutine without SubRoutine

▶ Integrate, as shown in the example, the SickAFx ladder routine as a SubRoutine using the command **JSR** (Jump To Subroutine).


Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]*
File Edit View Search Logic Communications Tools Window Help
Offline
No Forces
No Edits
Redundancy
RUN
OK
Energy Storage
I/O
Path: AB_ETHIP-1\192.168.1.1\Backplane\0*
Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Sequencer Equipment Phase Program
Controller Organizer
Controller PLC
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Program Tags
MainRoutine
SickAFx_A101WS_A103WS_FB_Enc1_GetSet
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
SICK_AFX60_A101WS_A103WS_AOI
Parameters and Local Tags
Logic
Data Types
User-Defined
SICK_AFX60_WS_DATA
Strings
Add-On-Defined
SICK_AFX60_A101WS_A103WS_AOI
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A4
[0] 1756-L73 PLC
[1] 1756-EN2TR EN2TR
Ethernet
1756-EN2TR EN2TR
AFM60A-Eth/IP SickAFX_A103WS_Enc1
AFS60A-Eth/IP SickAFX_A102WS_Enc2
Type Ladder Diagram (Main)
Description
Program MainProgram
Number of Rungs 0
MainProgram - MainRoutine*
0
(End)
MainRoutine*
MainProgram

JSR
Jump To Subroutine
Routine Name SickAFx_A101WS_A103WS_FB_Enc1_GetSet

Figure 53: MainRoutine with SubRoutine

5.6.3 Using the SubRoutine

- ▶ Switch the control system to the online mode.

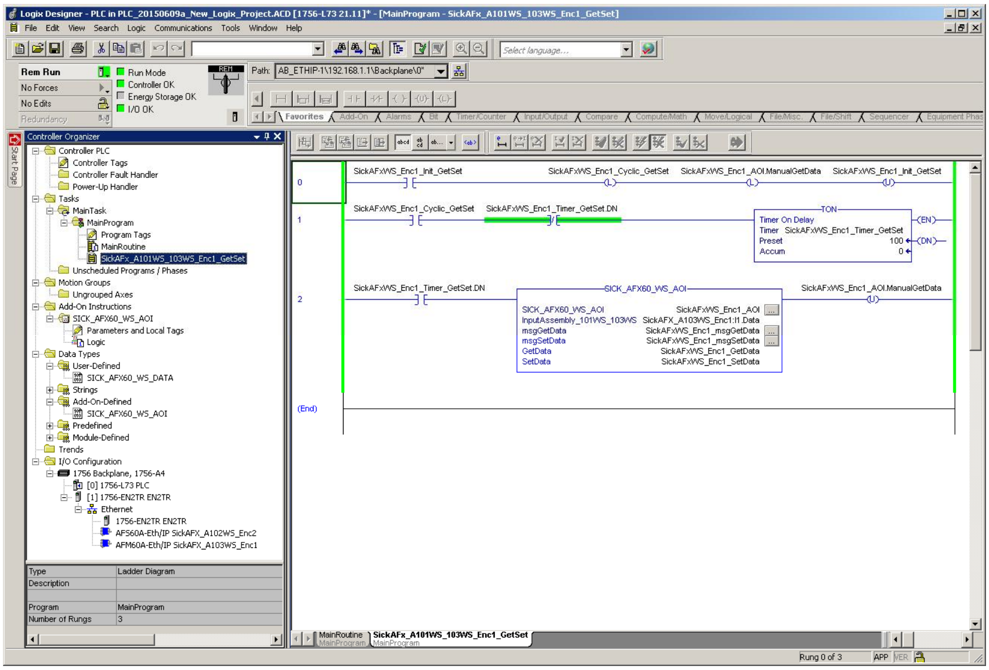

Logix Designer - PLC in PLC_20150609a_New_Logix_Project.ACD [1756-L73 21.11]* - [MainProgram - SickAFx_A101WS_103WS_Enc1_GetSet]
File Edit View Search Logic Communications Tools Window Help
Rem Run
No Forces
No Edits
Redundancy
Run Mode
Controller OK
Energy Storage OK
I/O OK
Path: AB_ETHIP-1\192.168.1.1\Backplane\0*
Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Sequencer Equipment Phase
Controller Organizer
Controller PLC
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Program Tags
MainRoutine
SickAFx_A101WS_103WS_Enc1_GetSet
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
SICK_AFX60_WS_AOI
Parameters and Local Tags
Logic
Data Types
User-Defined
SICK_AFX60_WS_DATA
Strings
Add-On-Defined
SICK_AFX60_WS_AOI
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A4
[0] 1756-L73 PLC
[1] 1756-EN2TR EN2TR
Ethernet
1756-EN2TR EN2TR
AFS60A-Eth/IP SickAFX_A102WS_Enc2
AFM60A-Eth/IP SickAFX_A103WS_Enc1
Type Ladder Diagram
Description
Program MainProgram
Number of Rungs 3
0
SickAFxWS_Enc1_Init_GetSet
SickAFxWS_Enc1_Cyclic_GetSet
SickAFxWS_Enc1_AOI.ManualGetData
SickAFxWS_Enc1_Init_GetSet
1
SickAFxWS_Enc1_Cyclic_GetSet
SickAFxWS_Enc1_Timer_GetSet.DN
TON
Timer On Delay
Timer SickAFxWS_Enc1_Timer_GetSet
Preset 100
Accum 0
EN
DN
2
SickAFxWS_Enc1_Timer_GetSet.DN
SICK_AFX60_WS_AOI
SICK_AFX60_WS_AOI SickAFxWS_Enc1_AOI
InputAssembly_101WS_103WS SickAFX_A103WS_Enc1:I1.Data
msgGetData SickAFxWS_Enc1_msgGetData
msgSetData SickAFxWS_Enc1_msgSetData
GetData SickAFxWS_Enc1_GetData
SetData SickAFxWS_Enc1_SetData
SickAFxWS_Enc1_AOI.ManualGetData
(End)
MainRoutine MainProgram
SickAFx_A101WS_103WS_Enc1_GetSet MainProgram
Rung 0 of 3
APP VER

Figure 54: Imported SickAFx ladder routine in the online mode

- ▶ Change in the MainProgram to **SickAFx_A101WS_A103WS_FB_Enc1_GetSet**.

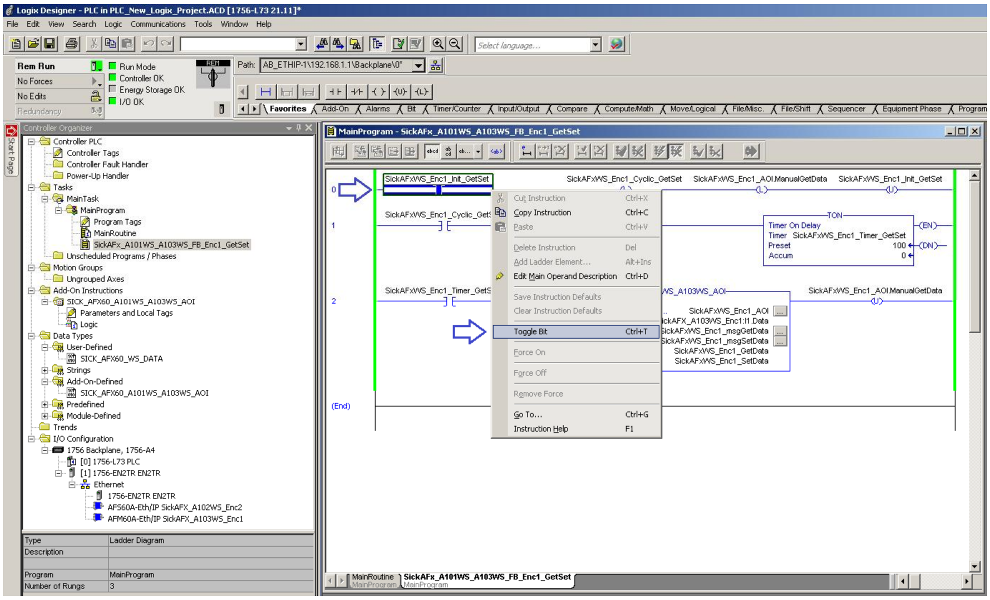

Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]*
File Edit View Search Logic Communications Tools Window Help
Rem Run
No Forces
No Edits
Redundancy
Run Mode
Controller OK
Energy Storage OK
I/O OK
Path: AB_ETHIP-1\192.168.1.1\Backplane\0*
Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Sequencer Equipment Phase Program
Controller Organizer
Controller PLC
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Program Tags
MainRoutine
SickAFx_A101WS_A103WS_FB_Enc1_GetSet
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
SICK_AFX60_A101WS_A103WS_AOI
Parameters and Local Tags
Logic
Data Types
User-Defined
SICK_AFX60_WS_DATA
Strings
Add-On-Defined
SICK_AFX60_A101WS_A103WS_AOI
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A4
[0] 1756-L73 PLC
[1] 1756-EN2TR EN2TR
Ethernet
1756-EN2TR EN2TR
AFS60A-Eth/IP SickAFX_A102WS_Enc2
AFM60A-Eth/IP SickAFX_A103WS_Enc1
Type Ladder Diagram
Description
Program MainProgram
Number of Rungs 3
MainProgram - SickAFx_A101WS_A103WS_FB_Enc1_GetSet
0
SickAFxWS_Enc1_Init_GetSet
SickAFxWS_Enc1_Cyclic_GetSet
SickAFxWS_Enc1_AOI.ManualGetData
SickAFxWS_Enc1_Init_GetSet
1
SickAFxWS_Enc1_Cyclic_GetS
TON
Timer On Delay
Timer SickAFxWS_Enc1_Timer_GetSet
Preset 100
Accum 0
EN
DN
2
SickAFxWS_Enc1_Timer_GetS
WS_A103WS_AOI
SickAFxWS_Enc1_AOI
SickAFX_A103WS_Enc1:I1.Data
SickAFxWS_Enc1_msgGetData
SickAFxWS_Enc1_msgSetData
SickAFxWS_Enc1_GetData
SickAFxWS_Enc1_SetData
SickAFxWS_Enc1_AOI.ManualGetData
(End)
Cut Instruction Ctrl+X
Copy Instruction Ctrl+C
Paste Ctrl+V
Delete Instruction Del
Add Ladder Element... Alt+Ins
Edit Main Operand Description Ctrl+D
Save Instruction Defaults
Clear Instruction Defaults
Toggle Bit Ctrl+T
Force On
Force Off
Remove Force
Go To... Ctrl+G
Instruction Help F1
MainRoutine MainProgram
SickAFx_A101WS_A103WS_FB_Enc1_GetSet MainProgram

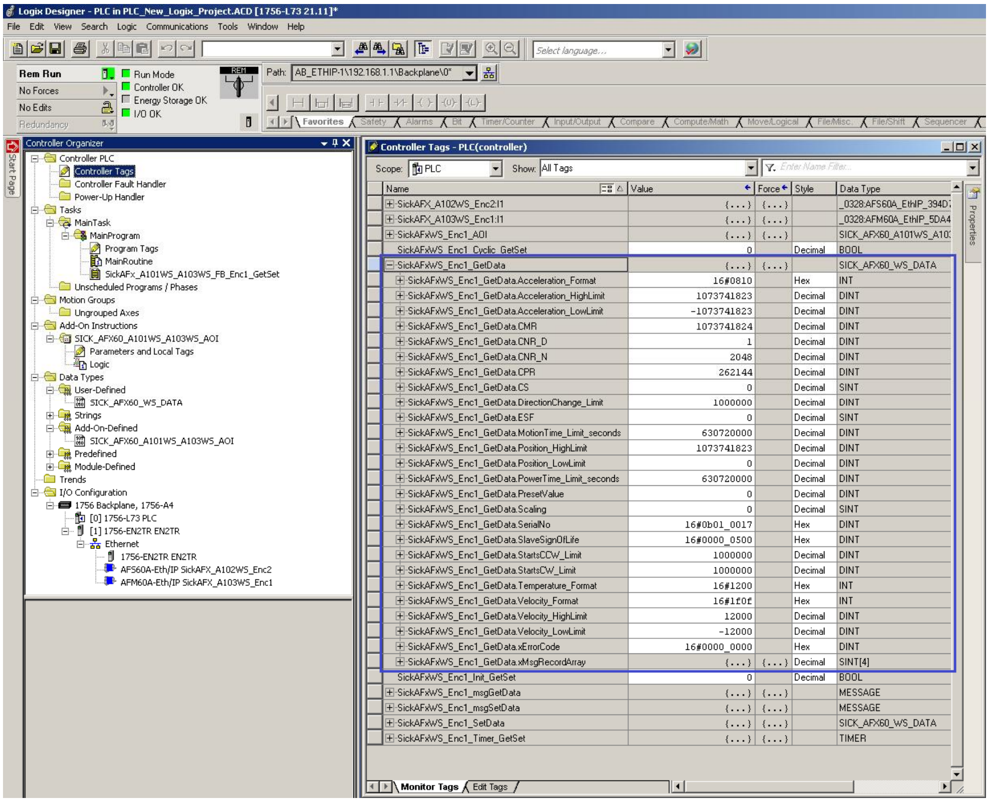
Figure 55: Initializing and starting the SubRoutine

- ▶ On the context menu for **SickAFxWS_Enc1_Init_GetSet** activate the command **Toggle Bit**.

In this way the connection is closed and the encoder can be configured both on the control system side and via the web server.

5.6.4 Reading and changing the parameters of the encoder

In **Controller Tags** you can read the parameters of the encoder in the node **SickAFxWS_Enc1_GetData**.

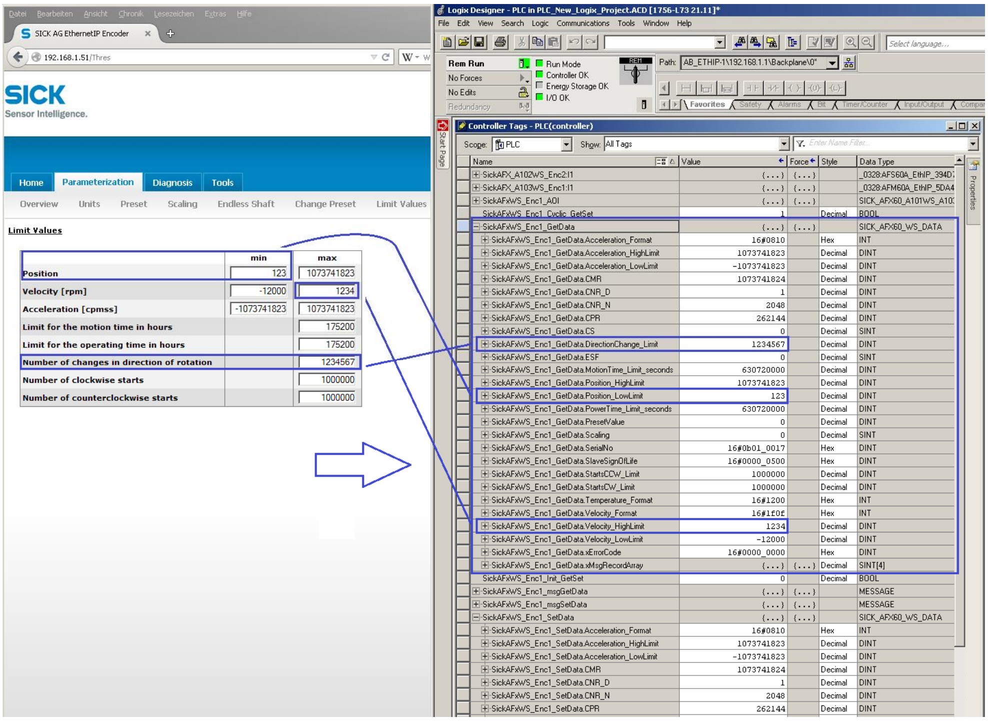

Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]*
File Edit View Search Logic Communications Tools Window Help
Select language...
Rem Run
No Forces
No Edits
Redundancy
Run Mode
Controller OK
Energy Storage OK
I/O OK
Path: AB_ETHIP-1\192.168.1.1\Backplane\0*
Favorites Safety Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Sequencer
Start Page
Controller Organizer
Controller PLC
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Program Tags
MainRoutine
SickAFx_A101WS_A103WS_FB_Enc1_GetSet
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
SICK_AFX60_A101WS_A103WS_AOI
Parameters and Local Tags
Logic
Data Types
User-Defined
SICK_AFX60_WS_DATA
Strings
Add-On-Defined
SICK_AFX60_A101WS_A103WS_AOI
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A4
[0] 1756-L73 PLC
[1] 1756-EN2TR EN2TR
Ethernet
1756-EN2TR EN2TR
AFS60A-Eth/IP SickAFX_A102WS_Enc2
AFM60A-Eth/IP SickAFX_A103WS_Enc1
Controller Tags - PLC(controller)
Properties
Monitor Tags
Edit Tags

Scope: PLC Show: All Tags Enter Name Filter...

Name	Value	Force	Style	Data Type
⊞ SickAFX_A102WS_Enc2:I1	{...}	{...}		_0328:AFS60A_EthIP_394D7
⊞ SickAFX_A103WS_Enc1:I1	{...}	{...}		_0328:AFM60A_EthIP_5DA4
⊞ SickAFxWS_Enc1_AOI	{...}	{...}		SICK_AFX60_A101WS_A10
SickAFxWS_Enc1_Cyclic_GetSet	0		Decimal	BOOL
⊟ SickAFxWS_Enc1_GetData	{...}	{...}		SICK_AFX60_WS_DATA
⊞ SickAFxWS_Enc1_GetData.Acceleration_Format	16#0810		Hex	INT
⊞ SickAFxWS_Enc1_GetData.Acceleration_HighLimit	1073741823		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.Acceleration_LowLimit	-1073741823		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.CMR	1073741824		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.CNR_D	1		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.CNR_N	2048		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.CPR	262144		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.CS	0		Decimal	SINT
⊞ SickAFxWS_Enc1_GetData.DirectionChange_Limit	1000000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.ESF	0		Decimal	SINT
⊞ SickAFxWS_Enc1_GetData.MotionTime_Limit_seconds	630720000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.Position_HighLimit	1073741823		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.Position_LowLimit	0		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.PowerTime_Limit_seconds	630720000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.PresetValue	0		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.Scaling	0		Decimal	SINT
⊞ SickAFxWS_Enc1_GetData.SerialNo	16#0b01_0017		Hex	DINT
⊞ SickAFxWS_Enc1_GetData.SlaveSignOfLife	16#0000_0500		Hex	DINT
⊞ SickAFxWS_Enc1_GetData.StartsCCW_Limit	1000000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.StartsCW_Limit	1000000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.Temperature_Format	16#1200		Hex	INT
⊞ SickAFxWS_Enc1_GetData.Velocity_Format	16#1f0f		Hex	INT
⊞ SickAFxWS_Enc1_GetData.Velocity_HighLimit	12000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.Velocity_LowLimit	-12000		Decimal	DINT
⊞ SickAFxWS_Enc1_GetData.xErrorCode	16#0000_0000		Hex	DINT
⊞ SickAFxWS_Enc1_GetData.xMsgRecordArray	{...}	{...}	Decimal	SINT[4]
SickAFxWS_Enc1_Init_GetSet	0		Decimal	BOOL
⊞ SickAFxWS_Enc1_msgGetData	{...}	{...}		MESSAGE
⊞ SickAFxWS_Enc1_msgSetData	{...}	{...}		MESSAGE
⊞ SickAFxWS_Enc1_SetData	{...}	{...}		SICK_AFX60_WS_DATA
⊞ SickAFxWS_Enc1_Timer_GetSet	{...}	{...}		TIMER

Figure 56: Reading the parameters in GetData

Parameters that you change in the web server are displayed in the control system.


Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
SICK AG EthernetIP Encoder
192.168.1.51/Thres
SICK
Sensor Intelligence.
Home
Parameterization
Diagnosis
Tools
Overview Units Preset Scaling Endless Shaft Change Preset Limit Values
Logix Designer - PLC in PLC_New_Logix_Project.ACD [1756-L73 21.11]*
File Edit View Search Logic Communications Tools Window Help
Select language...
Rem Run
No Forces
No Edits
Redundancy
Run Mode
Controller OK
Energy Storage OK
I/O OK
REM
Favorites Safety Alarms Bit Timer/Counter Input/Output Compar
Start Page
Properties
Controller Tags - PLC(controller)
Enter Name Filter...

Limit Values

	min	max
Position	123	1073741823
Velocity [rpm]	-12000	1234
Acceleration [cpmss]	-1073741823	1073741823
Limit for the motion time in hours		175200
Limit for the operating time in hours		175200
Number of changes in direction of rotation		1234567
Number of clockwise starts		1000000
Number of counterclockwise starts		1000000

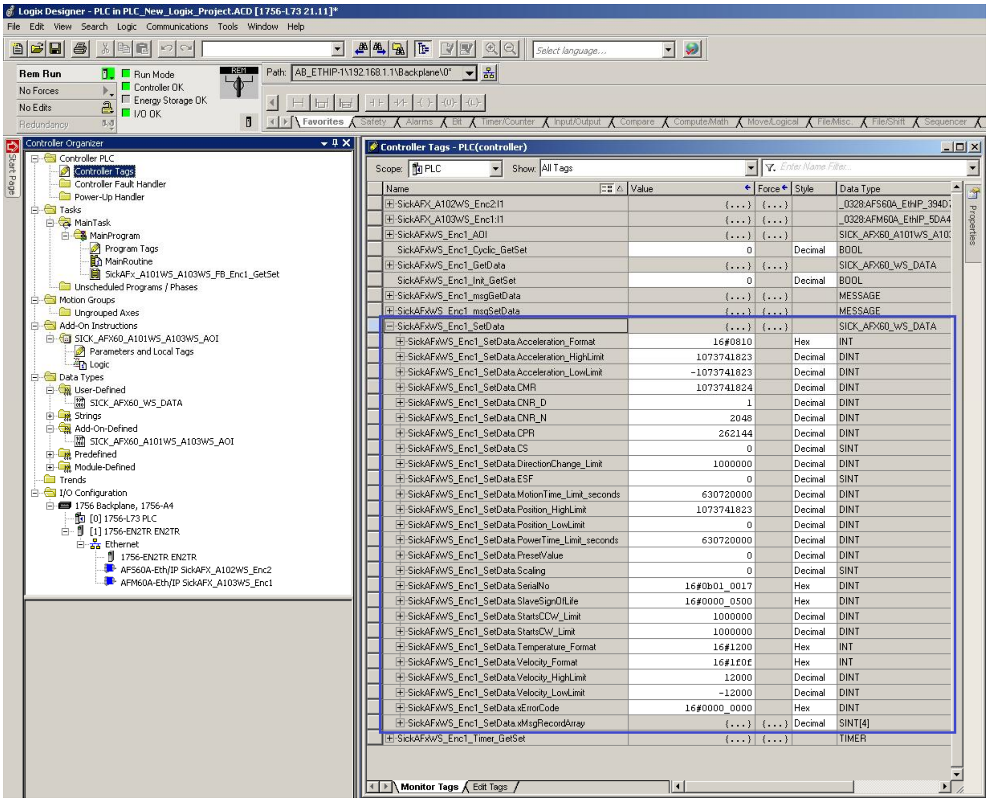
Path: AB_ETHIP-1\192.168.1.1\Backplane\0*

Scope: PLC Show: All Tags

Name	Value	Force	Style	Data Type
SickAFX_A102WS_Enc2:I1	{...}	{...}		_0328:AFS60A_EthIP_394D7
SickAFX_A103WS_Enc1:I1	{...}	{...}		_0328:AFM60A_EthIP_5DA4
SickAFxWS_Enc1_AOI	{...}	{...}		SICK_AFX60_A101WS_A10
SickAFxWS_Enc1_Cyclic_GetSet	1		Decimal	BOOL
SickAFxWS_Enc1_GetData	{...}	{...}		SICK_AFX60_WS_DATA
SickAFxWS_Enc1_GetData.Acceleration_Format	16#0810		Hex	INT
SickAFxWS_Enc1_GetData.Acceleration_HighLimit	1073741823		Decimal	DINT
SickAFxWS_Enc1_GetData.Acceleration_LowLimit	-1073741823		Decimal	DINT
SickAFxWS_Enc1_GetData.CMR	1073741824		Decimal	DINT
SickAFxWS_Enc1_GetData.CNR_D	1		Decimal	DINT
SickAFxWS_Enc1_GetData.CNR_N	2048		Decimal	DINT
SickAFxWS_Enc1_GetData.CPR	262144		Decimal	DINT
SickAFxWS_Enc1_GetData.CS	0		Decimal	SINT
SickAFxWS_Enc1_GetData.DirectionChange_Limit	1234567		Decimal	DINT
SickAFxWS_Enc1_GetData.ESF	0		Decimal	SINT
SickAFxWS_Enc1_GetData.MotionTime_Limit_seconds	630720000		Decimal	DINT
SickAFxWS_Enc1_GetData.Position_HighLimit	1073741823		Decimal	DINT
SickAFxWS_Enc1_GetData.Position_LowLimit	123		Decimal	DINT
SickAFxWS_Enc1_GetData.PowerTime_Limit_seconds	630720000		Decimal	DINT
SickAFxWS_Enc1_GetData.PresetValue	0		Decimal	DINT
SickAFxWS_Enc1_GetData.Scaling	0		Decimal	SINT
SickAFxWS_Enc1_GetData.SerialNo	16#0b01_0017		Hex	DINT
SickAFxWS_Enc1_GetData.SlaveSignOfLife	16#0000_0500		Hex	DINT
SickAFxWS_Enc1_GetData.StartsCCW_Limit	1000000		Decimal	DINT
SickAFxWS_Enc1_GetData.StartsCW_Limit	1000000		Decimal	DINT
SickAFxWS_Enc1_GetData.Temperature_Format	16#1200		Hex	INT
SickAFxWS_Enc1_GetData.Velocity_Format	16#1f0f		Hex	INT
SickAFxWS_Enc1_GetData.Velocity_HighLimit	1234		Decimal	DINT
SickAFxWS_Enc1_GetData.Velocity_LowLimit	-12000		Decimal	DINT
SickAFxWS_Enc1_GetData.xErrorCode	16#0000_0000		Hex	DINT
SickAFxWS_Enc1_GetData.xMsgRecordArray	{...}	{...}	Decimal	SINT[4]
SickAFxWS_Enc1_Init_GetSet	0		Decimal	BOOL
SickAFxWS_Enc1_msgGetData	{...}	{...}		MESSAGE
SickAFxWS_Enc1_msgSetData	{...}	{...}		MESSAGE
SickAFxWS_Enc1_SetData	{...}	{...}		SICK_AFX60_WS_DATA
SickAFxWS_Enc1_SetData.Acceleration_Format	16#0810		Hex	INT
SickAFxWS_Enc1_SetData.Acceleration_HighLimit	1073741823		Decimal	DINT
SickAFxWS_Enc1_SetData.Acceleration_LowLimit	-1073741823		Decimal	DINT
SickAFxWS_Enc1_SetData.CMR	1073741824		Decimal	DINT
SickAFxWS_Enc1_SetData.CNR_D	1		Decimal	DINT
SickAFxWS_Enc1_SetData.CNR_N	2048		Decimal	DINT
SickAFxWS_Enc1_SetData.CPR	262144		Decimal	DINT

Figure 57: Example for changing data in the web server and reading the parameters in the control system

In **Controller Tags** you can change the encoder parameters in the node **SickAFxWS_Enc1_SetData**.



Path: AB_ETHIP-1\192.168.1.1\Backplane\0*

Scope: PLC Show: All Tags

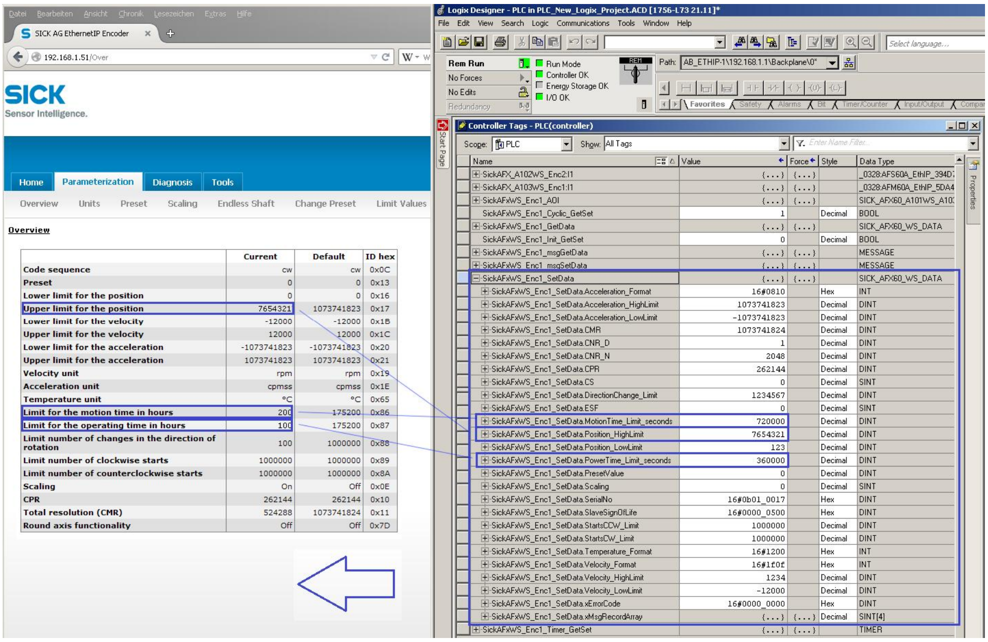
Name	Value	Force	Style	Data Type
SickAFX_A102WS_Enc2:I1	{...}	{...}		_0328:AFS60A_EthIP_394D7
SickAFX_A103WS_Enc1:I1	{...}	{...}		_0328:AFM60A_EthIP_5DA4
SickAFxWS_Enc1_AOI	{...}	{...}		SICK_AFX60_A101WS_A10
SickAFxWS_Enc1_Cyclic_GetSet	0		Decimal	BOOL
SickAFxWS_Enc1_GetData	{...}	{...}		SICK_AFX60_WS_DATA
SickAFxWS_Enc1_Init_GetSet	0		Decimal	BOOL
SickAFxWS_Enc1_msgGetData	{...}	{...}		MESSAGE
SickAFxWS_Enc1_msgSetData	{...}	{...}		MESSAGE
SickAFxWS_Enc1_SetData	{...}	{...}		SICK_AFX60_WS_DATA
SickAFxWS_Enc1_SetData.Acceleration_Format	16#0810		Hex	INT
SickAFxWS_Enc1_SetData.Acceleration_HighLimit	1073741823		Decimal	DINT
SickAFxWS_Enc1_SetData.Acceleration_LowLimit	-1073741823		Decimal	DINT
SickAFxWS_Enc1_SetData.CMR	1073741824		Decimal	DINT
SickAFxWS_Enc1_SetData.CNR_D	1		Decimal	DINT
SickAFxWS_Enc1_SetData.CNR_N	2048		Decimal	DINT
SickAFxWS_Enc1_SetData.CPR	262144		Decimal	DINT
SickAFxWS_Enc1_SetData.CS	0		Decimal	SINT
SickAFxWS_Enc1_SetData.DirectionChange_Limit	1000000		Decimal	DINT
SickAFxWS_Enc1_SetData.ESF	0		Decimal	SINT
SickAFxWS_Enc1_SetData.MotionTime_Limit_seconds	630720000		Decimal	DINT
SickAFxWS_Enc1_SetData.Position_HighLimit	1073741823		Decimal	DINT
SickAFxWS_Enc1_SetData.Position_LowLimit	0		Decimal	DINT
SickAFxWS_Enc1_SetData.PowerTime_Limit_seconds	630720000		Decimal	DINT
SickAFxWS_Enc1_SetData.PresetValue	0		Decimal	DINT
SickAFxWS_Enc1_SetData.Scaling	0		Decimal	SINT
SickAFxWS_Enc1_SetData.SerialNo	16#0b01_0017		Hex	DINT
SickAFxWS_Enc1_SetData.SlaveSignOfLife	16#0000_0500		Hex	DINT
SickAFxWS_Enc1_SetData.StartsCCW_Limit	1000000		Decimal	DINT
SickAFxWS_Enc1_SetData.StartsCW_Limit	1000000		Decimal	DINT
SickAFxWS_Enc1_SetData.Temperature_Format	16#1200		Hex	INT
SickAFxWS_Enc1_SetData.Velocity_Format	16#1f0f		Hex	INT
SickAFxWS_Enc1_SetData.Velocity_HighLimit	12000		Decimal	DINT
SickAFxWS_Enc1_SetData.Velocity_LowLimit	-12000		Decimal	DINT
SickAFxWS_Enc1_SetData.xErrorCode	16#0000_0000		Hex	DINT
SickAFxWS_Enc1_SetData.xMsgRecordArray	{...}	{...}	Decimal	SINT[4]
SickAFxWS_Enc1_Timer_GetSet	{...}	{...}		TIMER

Figure 58: Changing parameters in SetData

Parameters that you change in the control system are displayed in the web server on the Parameterization page.

NOTE

The web browser must be refreshed to display the modified data.



Path: AB_ETHIP-1\192.168.1.1\Backplane\0*

Overview

	Current	Default	ID hex
Code sequence	cw	cw	0x0C
Preset	0	0	0x13
Lower limit for the position	0	0	0x16
Upper limit for the position	7654321	1073741823	0x17
Lower limit for the velocity	-12000	-12000	0x1B
Upper limit for the velocity	12000	12000	0x1C
Lower limit for the acceleration	-1073741823	-1073741823	0x20
Upper limit for the acceleration	1073741823	1073741823	0x21
Velocity unit	rpm	rpm	0x19
Acceleration unit	cpmss	cpmss	0x1E
Temperature unit	°C	°C	0x65
Limit for the motion time in hours	200	175200	0x86
Limit for the operating time in hours	100	175200	0x87
Limit number of changes in the direction of rotation	100	1000000	0x88
Limit number of clockwise starts	1000000	1000000	0x89
Limit number of counterclockwise starts	1000000	1000000	0x8A
Scaling	On	Off	0x0E
CPR	262144	262144	0x10
Total resolution (CMR)	524288	1073741824	0x11
Round axis functionality	Off	Off	0x7D

Controller Tags - PLC(controller)

Scope: PLC Show: All Tags Enter Name Filter...

Name	Value	Force	Style	Data Type
⊞ SickAFX_A102WS_Enc2:I1	{...}	{...}		_0328:AFS60A_EthIP_394D7
⊞ SickAFX_A103WS_Enc1:I1	{...}	{...}		_0328:AFM60A_EthIP_5DA4
⊞ SickAFxWS_Enc1_AOI	{...}	{...}		SICK_AFX60_A101WS_A10
SickAFxWS_Enc1_Cyclic_GetSet	1		Decimal	BOOL
⊞ SickAFxWS_Enc1_GetData	{...}	{...}		SICK_AFX60_WS_DATA
SickAFxWS_Enc1_Init_GetSet	0		Decimal	BOOL
⊞ SickAFxWS_Enc1_msgGetData	{...}	{...}		MESSAGE
⊞ SickAFxWS_Enc1_msgSetData	{...}	{...}		MESSAGE
⊟ SickAFxWS_Enc1_SetData	{...}	{...}		SICK_AFX60_WS_DATA
⊞ SickAFxWS_Enc1_SetData.Acceleration_Format	16#0810		Hex	INT
⊞ SickAFxWS_Enc1_SetData.Acceleration_HighLimit	1073741823		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.Acceleration_LowLimit	-1073741823		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.CMR	1073741824		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.CNR_D	1		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.CNR_N	2048		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.CPR	262144		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.CS	0		Decimal	SINT
⊞ SickAFxWS_Enc1_SetData.DirectionChange_Limit	1234567		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.ESF	0		Decimal	SINT
⊞ SickAFxWS_Enc1_SetData.MotionTime_Limit_seconds	720000		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.Position_HighLimit	7654321		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.Position_LowLimit	123		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.PowerTime_Limit_seconds	360000		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.PresetValue	0		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.Scaling	0		Decimal	SINT
⊞ SickAFxWS_Enc1_SetData.SerialNo	16#0b01_0017		Hex	DINT
⊞ SickAFxWS_Enc1_SetData.SlaveSignOfLife	16#0000_0500		Hex	DINT
⊞ SickAFxWS_Enc1_SetData.StartsCCW_Limit	1000000		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.StartsCW_Limit	1000000		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.Temperature_Format	16#1200		Hex	INT
⊞ SickAFxWS_Enc1_SetData.Velocity_Format	16#1f0f		Hex	INT
⊞ SickAFxWS_Enc1_SetData.Velocity_HighLimit	1234		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.Velocity_LowLimit	-12000		Decimal	DINT
⊞ SickAFxWS_Enc1_SetData.xErrorCode	16#0000_0000		Hex	DINT
⊞ SickAFxWS_Enc1_SetData.xMsgRecordArray	{...}	{...}	Decimal	SINT[4]
⊞ SickAFxWS_Enc1_Timer_GetSet	{...}	{...}		TIMER

Figure 59: Example for changing data in the control system and reading in the web server

WARNING

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

As soon as you have entered the value and accepted the entry using the Enter key, the value is applied as a position value (see Figure 109 on page 91)!

5.7 Function block

For the communication between an Allen-Bradley control system and the AFS60/AFM60 EtherNet/IP you can use a function block.

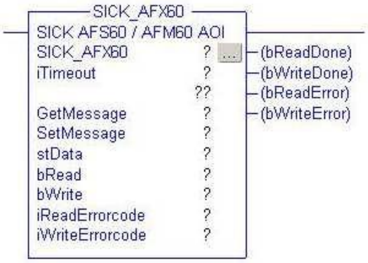
5.7.1 Prerequisites

- You will find the function block and the complete documentation on the SICK homepage in the Internet: “EthernetIP function block – EtherNet/IP function block for encoderspecific functions in RSLogix5000, included manual.”.
- The encoder must be integrated in the control system using an EDS file or as a generic module.

5.7.2 Importing and connecting

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

Then open the function block and connect it. Only with valid connection is it possible to read parameters from the encoder or to write to the encoder.



```
SICK_AFX60
SICK AFS60 / AFM60 AOI
SICK_AFX60          ? ...   (bReadDone)
iTimeout            ?       (bWriteDone)
                   ??       (bReadError)
GetMessage          ?       (bWriteError)
SetMessage          ?
stData              ?
bRead               ?
bWrite              ?
iReadErrorcode      ?
iWriteErrorcode     ?
```

Figure 60: Function block in the Rockwell control system

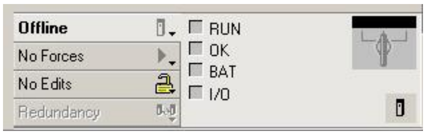
You will find a detailed description of how to connect in the operating instructions “AFS60/AFM60 EtherNet/IP Add-On Instruction”. These operating instructions are supplied with the function block as a PDF.

5.8 Program examples

The following examples show the configuration of two programs that read (temperature) and write (preset) acyclic data. For this purpose the programs are written in ladder logic with the aid of the software RSLogix 5000 from Rockwell Automation.

NOTE

During programming the control system must be in the offline mode.



Offline
No Forces
No Edits
Redundancy

□ RUN
□ OK
□ BAT
□ I/O

Figure 61: Control system in the offline mode

- ▶ First you must define and declare the variables for the program.
- ▶ Then add the program blocks to the ladder logic and assign the variables as appropriate.
- ▶ After that you must download the program to the control system.
- ▶ Finally, you can test the program.

5.8.1 Reading temperature

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

Defining and declaring variables

As the initial step the variables TEMP_Trigger, TEMP_OneShot, TEMP_Value and TEMP_Message must be defined and declared for the program.

First the variable TEMP_Trigger, which controls the reading process, is added.

- In the **Controller Organizer**, using the right mouse button click **Controller Tags** and select **New Tag**.

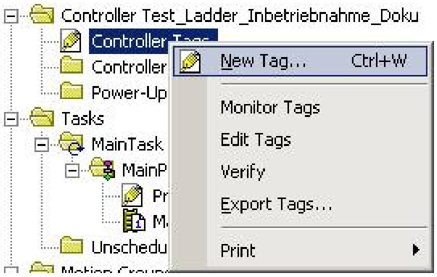
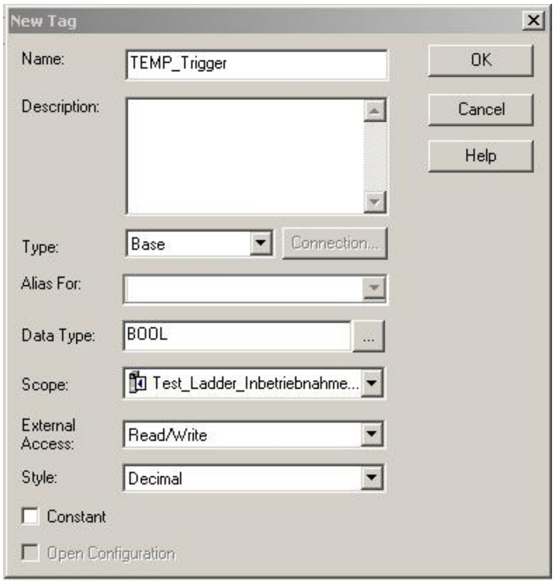


Figure 62: Adding a new variable

The **New Tag** dialog box opens.



Name: TEMP_Trigger

Description:

Type: Base

Alias For:

Data Type: BOOL

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style: Decimal

□ Constant

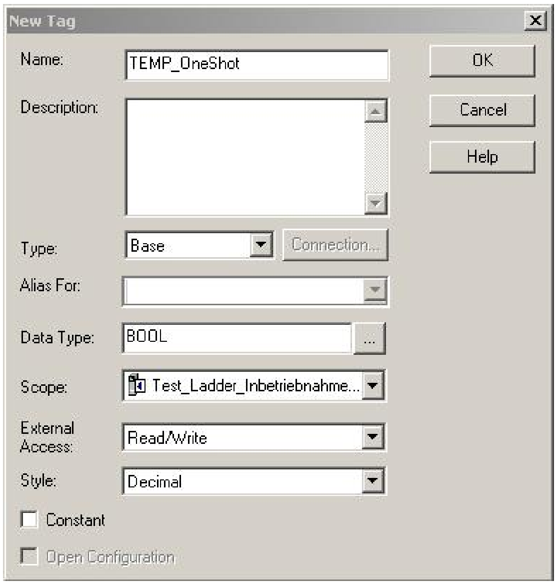
□ Open Configuration

Figure 63: Definition of the variable TEMP_Trigger

- In the **Name** field enter TEMP_Trigger, in the **Data Type** field select the data type BOOL and click **OK**.

To only trigger the action once, a further element, in this case an edge-sensitive element, must be defined and declared. This element ensures that the action is only triggered if an edge change from 0 to 1 occurs in the variable TEMP_Trigger.

- Select again **New Tag**.



Name: TEMP_OneShot

Description:

Type: Base

Alias For:

Data Type: BOOL

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style: Decimal

□ Constant

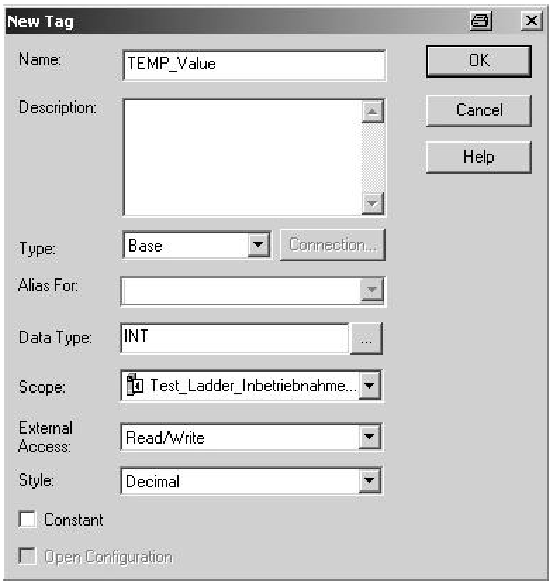
□ Open Configuration

Figure 64: Definition of the variable TEMP_OneShot

- ▶ In the **New Tag** dialog box enter TEMP_OneShot in the **Name** field, in the **Data Type** select the data type BOOL and click **OK**.

A further variable must be added that will then contain the temperature value later (see Table 21 on page 35, attribute ID 64h, temperature value).

- ▶ Select again **New Tag**.



Name: TEMP_Value

Description:

Type: Base

Alias For:

Data Type: INT

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style: Decimal

□ Constant

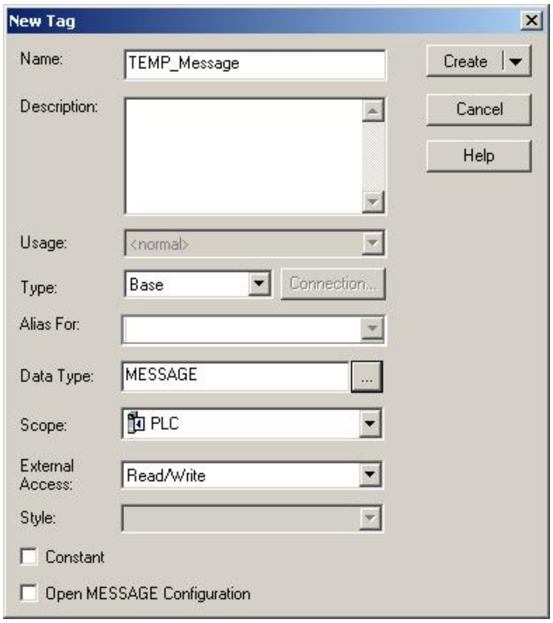
□ Open Configuration

Figure 65: Definition of the variable TEMP_Value

- ▶ In the **New Tag** dialog box enter TEMP_Value in the **Name** field, select in the **Data Type** field the data type INT and click **OK**.

Finally a further variable must be defined and declared that obtains the temperature value from the control system.

- ▶ Select again **New Tag**.



Name: TEMP_Message

Description:

Usage: <normal>

Type: Base

Alias For:

Data Type: MESSAGE

Scope: PLC

External Access: Read/Write

Style:

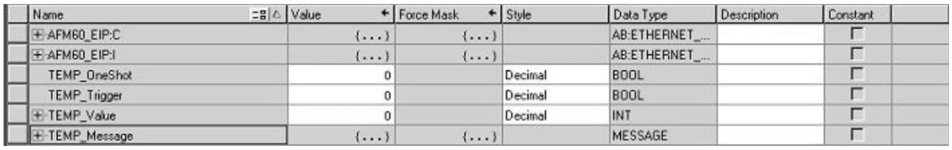
□ Constant

□ Open MESSAGE Configuration

Figure 66: Definition of the variable TEMP_Message

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

Figure 67 shows the resulting variable structure for reading the temperature acyclically.



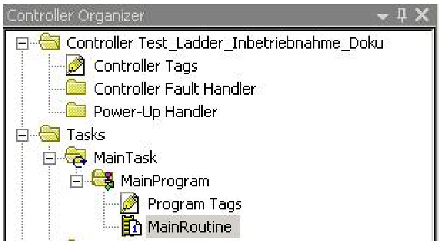
Name	Value	Force Mask	Style	Data Type	Description	Constant
⊞ AFM60_EIP:C	{...}	{...}		AB:ETHERNET_...		□
⊞ AFM60_EIP:I	{...}	{...}		AB:ETHERNET_...		□
TEMP_OneShot	0		Decimal	BOOL		□
TEMP_Trigger	0		Decimal	BOOL		□
⊞ TEMP_Value	0		Decimal	INT		□
⊞ TEMP_Message	{...}	{...}		MESSAGE		□

Figure 67: Variable structure for reading the temperature

Defining process sequence

After you have defined and declared the variables, the program blocks must be inserted in the ladder logic and the variables assigned as appropriate.

In **Tasks**, **Main Task**, **MainProgram** open the **MainRoutine** window.



```
Controller Test_Ladder_Inbetriebnahme_Doku
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
Tasks
    MainTask
        MainProgram
            Program Tags
            MainRoutine
```

Figure 68: Opening MainRoutine

For the first block an input is added that is to trigger the “read temperature” process.

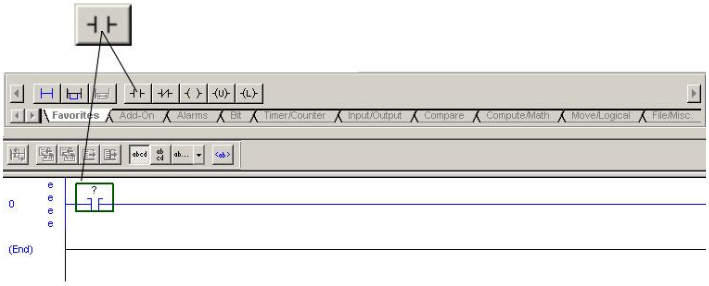
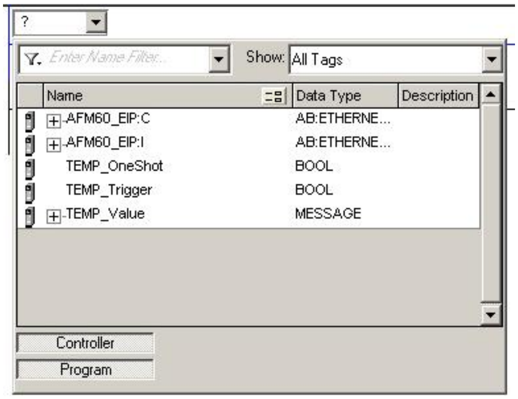


Figure 69: Adding ExamineOn block

- ▶ On the **Favorites** tab select the **ExamineOn** block and add it to the **MainRoutine**.

The related variable must be assigned to this input, in our example the variable TEMP_Trigger.



Name	Data Type	Description
AFM60_EIP:C	AB:ETHERNE...	
AFM60_EIP:I	AB:ETHERNE...	
TEMP_OneShot	BOOL	
TEMP_Trigger	BOOL	
TEMP_Value	MESSAGE	

Figure 70: Allocation of the variable TEMP_Trigger to ExamineOn

- ▶ Click on the **question mark**.
  A drop-down menu will open.
- ▶ Select the variable TEMP_Trigger.

The ONS block must be added for the edge sensitivity of the process sequence.

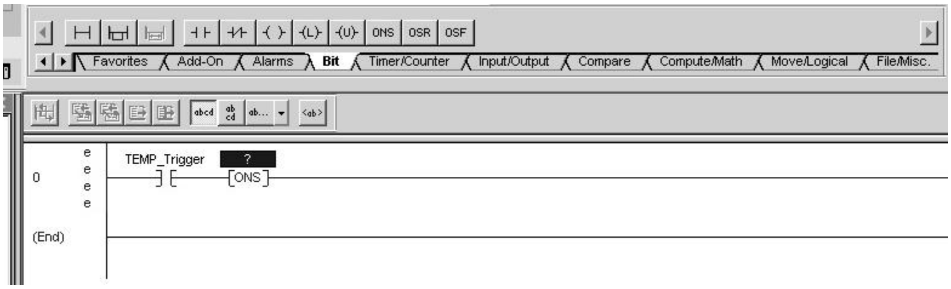
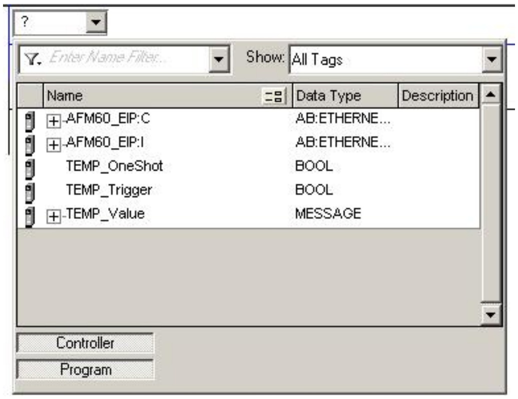


Figure 71: Adding ONS block

- ▶ On the **Bit** tab select the **ONS** block and add it to the **MainRoutine**.

A variable must also be assigned to this block.



Name	Data Type	Description
AFM60_EIP:C	AB:ETHERNE...	
AFM60_EIP:I	AB:ETHERNE...	
TEMP_OneShot	BOOL	
TEMP_Trigger	BOOL	
TEMP_Value	MESSAGE	

Figure 72: Allocation of the variable TEMP_OneShot to ONS

- ▶ Click on the **question mark**.
  A drop-down menu will open.
- ▶ Select the variable TEMP_OneShot.

In the next step the message must be configured to read the temperature value from the encoder.

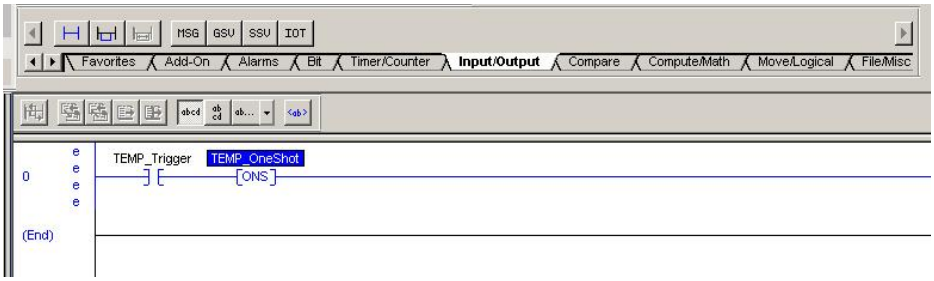
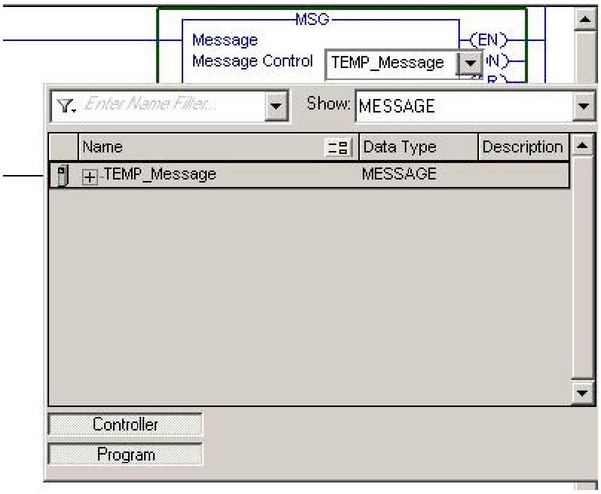


Figure 73: Adding MSG block

- ▶ On the **Input/Output** tab select the **MSG** block and add it to the **MainRoutine**.

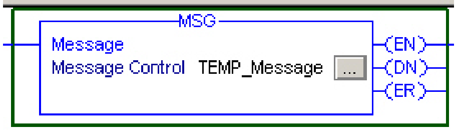


Name	Data Type	Description
TEMP_Message	MESSAGE	

Figure 74: Allocation of the variable TEMP_Message to MSG

- ▶ In the **Message Control** field select the variable TEMP_Message.

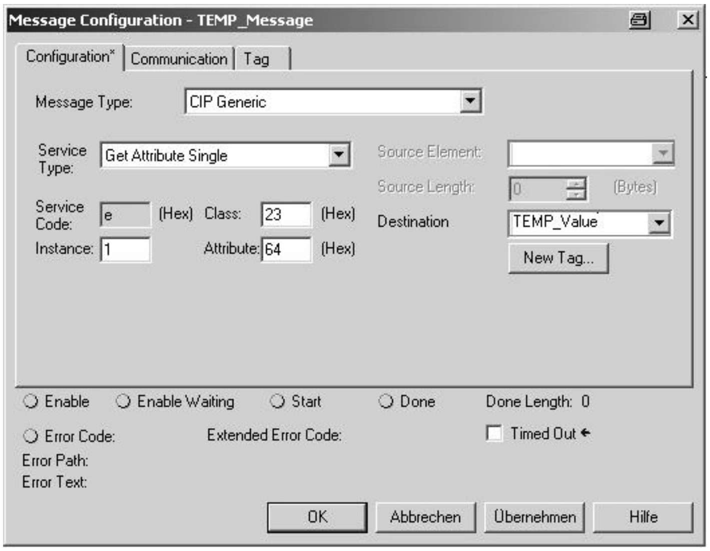
The MSG block must then be configured.



MSG
Message
Message Control TEMP_Message

Figure 75: Opening configuration dialog box for the MSG block

- ▶ For this purpose click the button with the three dots.
  The **Message Configuration** dialog box will open.



Message Configuration - TEMP_Message

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 23 (Hex)

Instance: 1 Attribute: 64 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination TEMP_Value

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

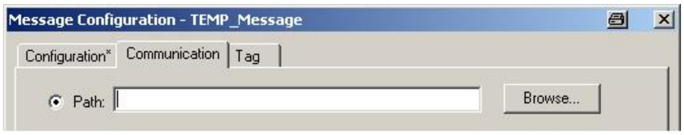
Figure 76: Configuration dialog box for the MSG block

- ▶ Configure the following parameters on the **Configuration** tab:
  - ○ **Service Type**: Get Attribute Single (see Table 18 on page 28)
  - ○ **Instance**: 1 (as only one device is connected to the control system)
  - ○ **Class**: 23(h) (Position Sensor Object, see Table 5 on page 19)
  - ○ **Attribute**: 64(h) (Temperature Value, see Table 21 on page 35)
  - ○ **Destination**: TEMP_Value

NOTE

TEMP_Value is the fourth variable added. The value for the temperature is written to this variable on executing the example program.

- ▶ Open the **Communication** tab.



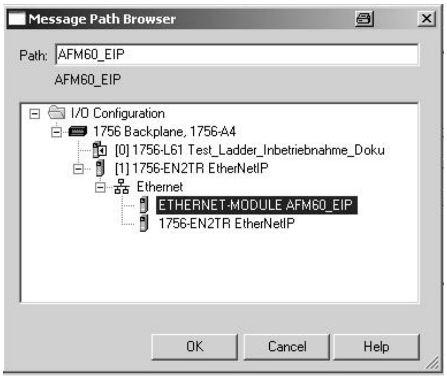
Message Configuration - TEMP_Message

● Path:

Figure 77: Communication tab

- ▶ Beside the **Path** field click the **Browse...** button.
  The **Message Path Browser** dialog box will open.

- ▶ Select the encoder connected.



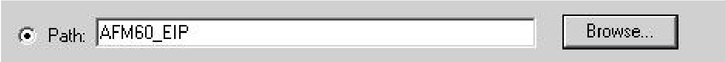
Message Path Browser

Path: AFM60_EIP

AFM60_EIP

```
⊟ I/O Configuration
  ⊟ 1756 Backplane, 1756-A4
      [0] 1756-L61 Test_Ladder_Inbetriebnahme_Doku
    ⊟ [1] 1756-EN2TR EtherNetIP
      ⊟ Ethernet
          ETHERNET-MODULE AFM60_EIP
          1756-EN2TR EtherNetIP
```

Figure 78: Selecting encoder



● Path: AFM60_EIP

Figure 79: Selected encoder

The encoder is applied in the **Path** field.

- ▶ Close the **Message Path Browser** dialog box using **OK**.

Transferring program to the control system

Finally the program is transferred to the control system.

- ▶ From the **Offline** menu select the **Download** command.

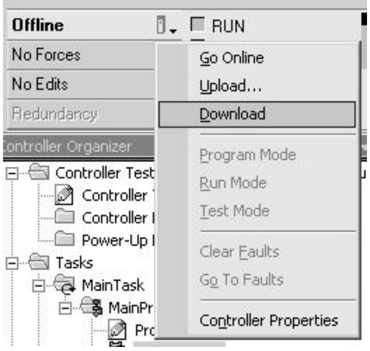
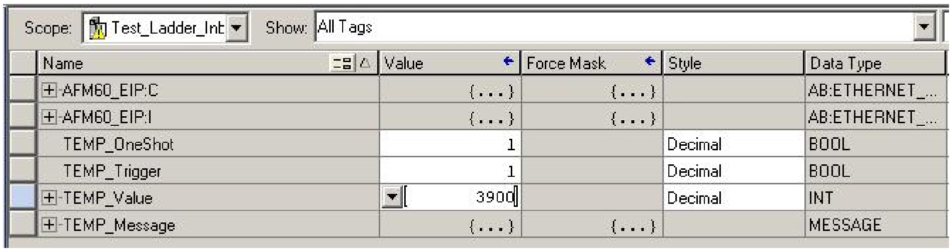


Figure 80: Transferring the program to the control system

- ▶ Accept the next message.

Testing program

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



Scope: Test_Ladder_Int Show: All Tags

Name	Value	Force Mask	Style	Data Type
⊞-AFM60_EIP:C	{...}	{...}		AB:ETHERNET_...
⊞-AFM60_EIP:I	{...}	{...}		AB:ETHERNET_...
TEMP_OneShot	1		Decimal	BOOL
TEMP_Trigger	1		Decimal	BOOL
⊞-TEMP_Value	3900		Decimal	INT
⊞-TEMP_Message	{...}	{...}		MESSAGE

Figure 81: 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

As the initial step the variables PRESET_Trigger, PRESET_OneShot, PRESET_Value and PRESET_Message must be defined and declared for the program.

First the variable PRESET_Trigger is added, this variable controls the process.

▶ In the **Controller Organizer**, using the right mouse button click **Controller Tags** and select **New Tag**.

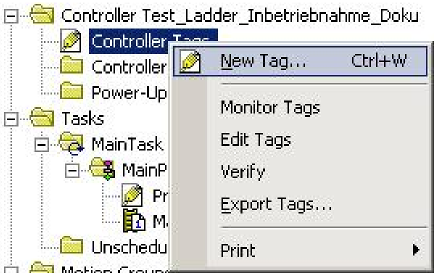
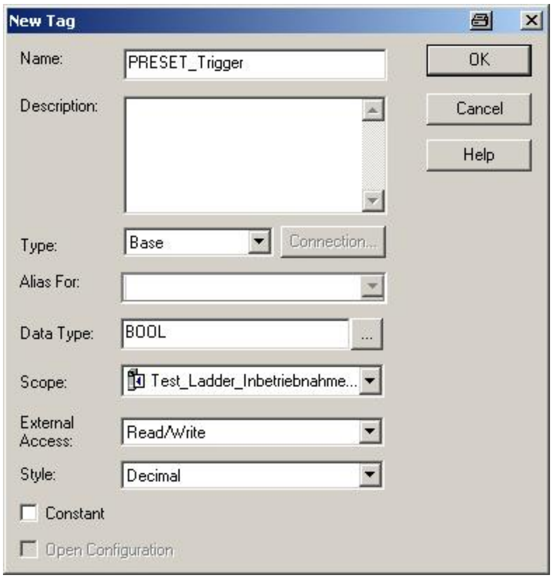


Figure 82: Adding a new variable

The **New Tag** dialog box opens.


New Tag

Name: PRESET_Trigger

Description:

Type: Base Connection...

Alias For:

Data Type: BOOL

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style: Decimal

□ Constant

□ Open Configuration

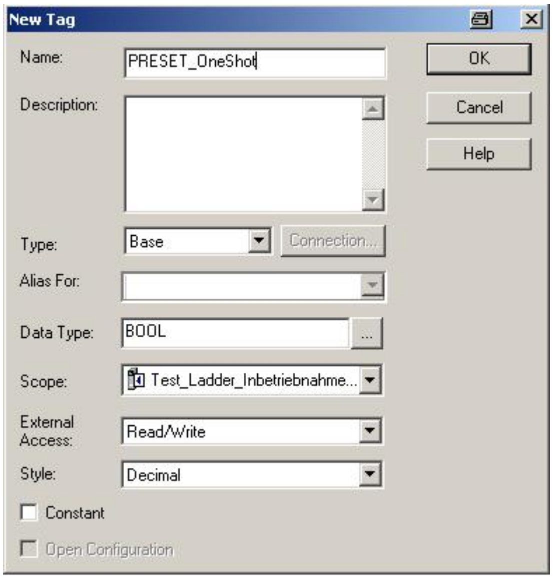
OK Cancel Help

Figure 83: Definition of the variable PRESET_Trigger

▶ In the **Name** field enter PRESET_Trigger, in the **Data Type** select the data type BOOL and click **OK**.

To only trigger the action once, a further element, in this case an edge-sensitive element, must be defined and declared. This element ensures that the action is only triggered if an edge change from 0 to 1 occurs in the variable PRESET_Trigger.

- ▶ Select again **New Tag**.



Name: PRESET_OneShot

Description:

Type: Base

Alias For:

Data Type: BOOL

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style: Decimal

☐ Constant

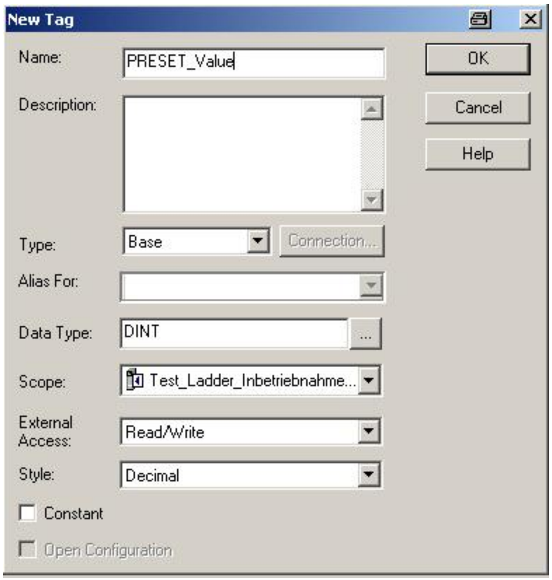
☐ Open Configuration

Figure 84: Definition of the variable PRESET_OneShot

- ▶ In the **New Tag** dialog box enter PRESET_OneShot in the **Name** field, select in the **Data Type** field the data type BOOL and click **OK**.

A further variable must be added that will then contain the preset value later (see Table 21 on page 35, attribute ID 13h, preset value).

- ▶ Select again **New Tag**.



Name: PRESET_Value

Description:

Type: Base

Alias For:

Data Type: DINT

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style: Decimal

☐ Constant

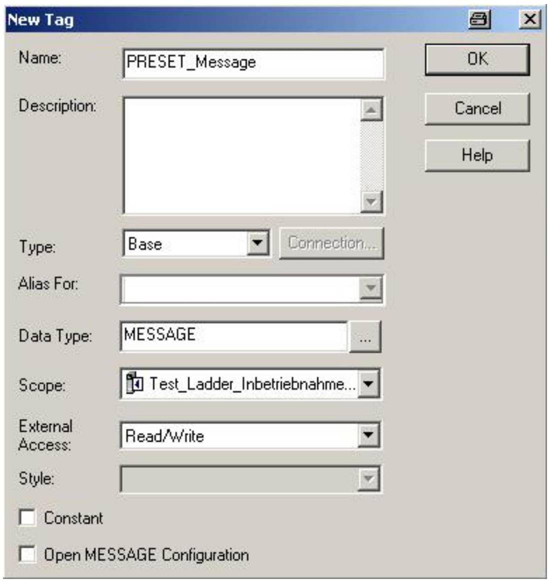
☐ Open Configuration

Figure 85: Definition of the variable PRESET_Value

- ▶ In the **New Tag** dialog box enter PRESET_Value in the **Name** field, select in the **Data Type** field the data type DINT and click **OK**.

Finally a further variable must be defined and declared that obtains the preset value from the control system.

▶ Select again **New Tag**.



New Tag

Name: PRESET_Message

Description:

Type: Base

Alias For:

Data Type: MESSAGE

Scope: Test_Ladder_Inbetriebnahme...

External Access: Read/Write

Style:

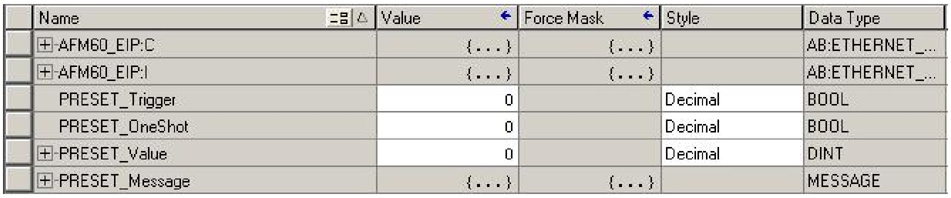
□ Constant

□ Open MESSAGE Configuration

Figure 86: Definition of the variable PRESET_Message

▶ In the **New Tag** dialog box enter PRESET_Message in the **Name** field, select in the **Data Type** field the data type MESSAGE and click **OK**.

Figure 87 shows the resulting variable structure for setting a preset value.



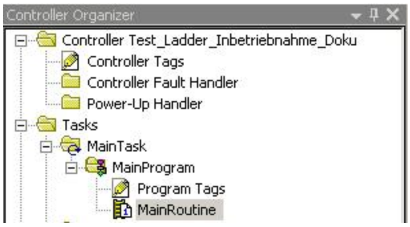
Name	Value	Force Mask	Style	Data Type
⊞ AFM60_EIP:C	{...}	{...}		AB:ETHERNET_...
⊞ AFM60_EIP:I	{...}	{...}		AB:ETHERNET_...
PRESET_Trigger	0		Decimal	BOOL
PRESET_OneShot	0		Decimal	BOOL
⊞ PRESET_Value	0		Decimal	DINT
⊞ PRESET_Message	{...}	{...}		MESSAGE

Figure 87: Variable structure for setting a preset value

Defining process sequence

After you have defined and declared the variables, the program blocks must be inserted in the ladder logic and the variables assigned as appropriate.

In **Tasks**, **Main Task**, **MainProgram** open the **MainRoutine** window.



Controller Organizer

- Controller Test_Ladder_Inbetriebnahme_Doku
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
  - MainTask
    - MainProgram
      - Program Tags
      - MainRoutine

Figure 88: Opening MainRoutine

If the process sequence for writing a preset value is to run in parallel with the previous example, then a new thread must be added.

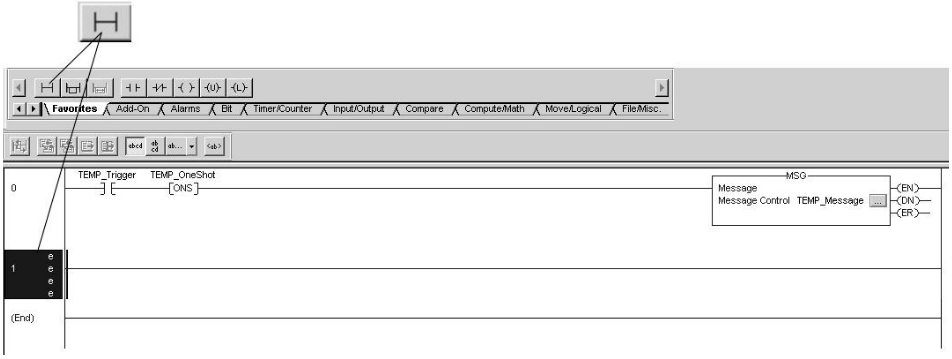


Figure 89: Adding Rung block

- ▶ On the **Favorites** tab select the **Rung** block and add it to the **MainRoutine**.

For the first block an input is added that is to trigger the “set preset value” process.

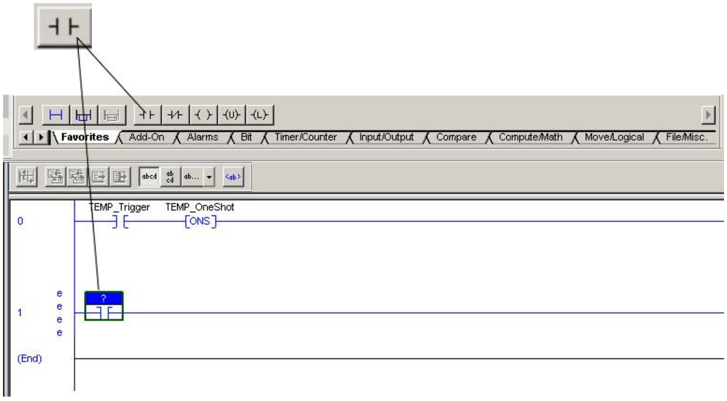
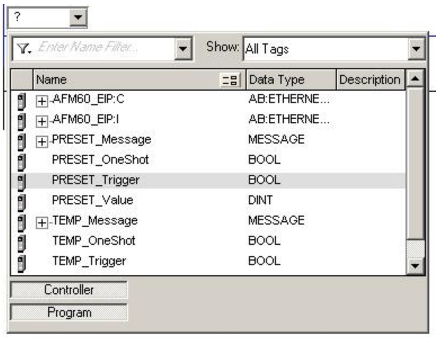


Figure 90: Adding ExamineOn block

- ▶ On the **Favorites** tab select the **ExamineOn** block and add it to the **MainRoutine**.

The related variable must be assigned to this input, in our example the variable PRESET_Trigger.



Name	Data Type	Description
⊞ AFM60_EIP:C	AB:ETHERNE...	
⊞ AFM60_EIP:I	AB:ETHERNE...	
⊞ PRESET_Message	MESSAGE	
PRESET_OneShot	BOOL	
PRESET_Trigger	BOOL	
PRESET_Value	DINT	
⊞ TEMP_Message	MESSAGE	
TEMP_OneShot	BOOL	
TEMP_Trigger	BOOL	

Figure 91: Allocation of the variable PRESET_Trigger to ExamineOn

- ▶ Click on the **question mark**.
  A drop-down menu will open.
- ▶ Select the variable PRESET_Trigger.

The ONS block must be added for the edge sensitivity of the process sequence.

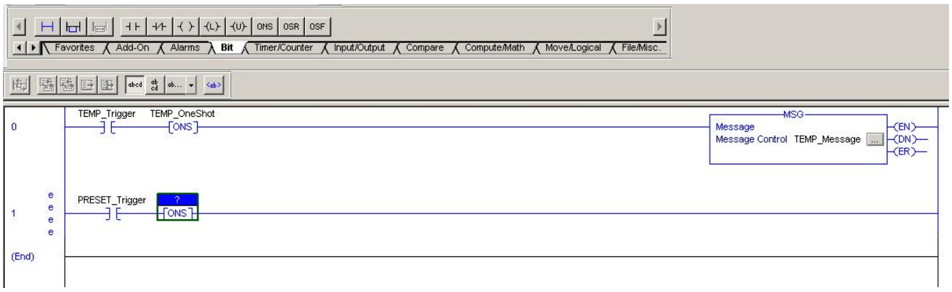
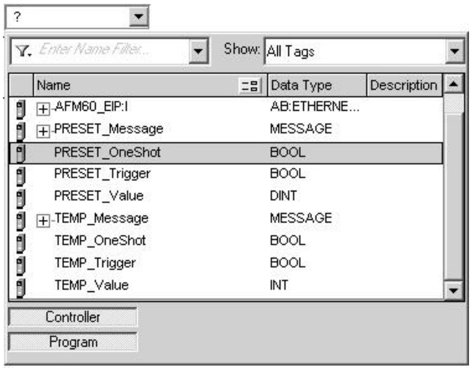


Figure 92: Adding ONS block

- ▶ On the **Bit** tab select the **ONS** block and add it to the **MainRoutine**.

A variable must also be assigned to this block.



Name	Data Type	Description
AFM60_EIP:I	AB:ETHERNE...	
PRESET_Message	MESSAGE	
PRESET_OneShot	BOOL	
PRESET_Trigger	BOOL	
PRESET_Value	DINT	
TEMP_Message	MESSAGE	
TEMP_OneShot	BOOL	
TEMP_Trigger	BOOL	
TEMP_Value	INT	

Figure 93: Allocation of the variable PRESET_OneShot to ONS

- ▶ Click on the **question mark**.
  A drop-down menu will open.
- ▶ Select the variable PRESET_OneShot.

In the next step the message must be configured to write the preset value to the encoder.

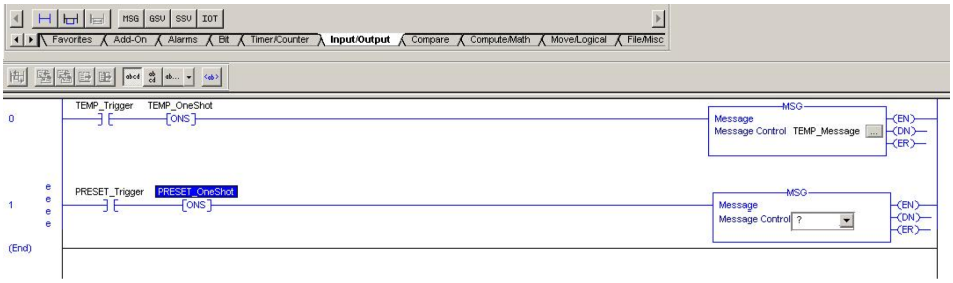
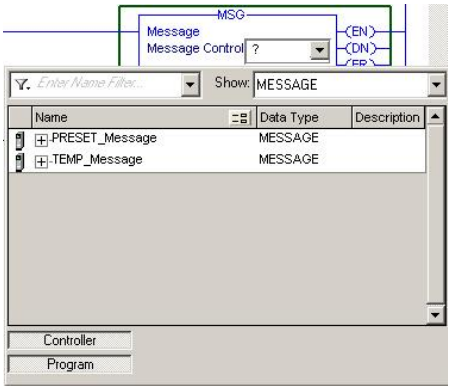


Figure 94: Adding MSG block

- ▶ On the **Input/Output** tab select the **MSG** block and add it to the **MainRoutine**.



MSG

Message

Message Control ?

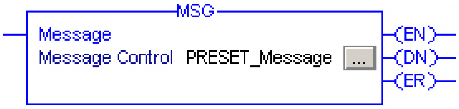
Enter Name Filter...

Show: MESSAGE

Name	Data Type	Description
PRESET_Message	MESSAGE	
TEMP_Message	MESSAGE	

Figure 95: Allocation of the variable PRESET_Message to MSG

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



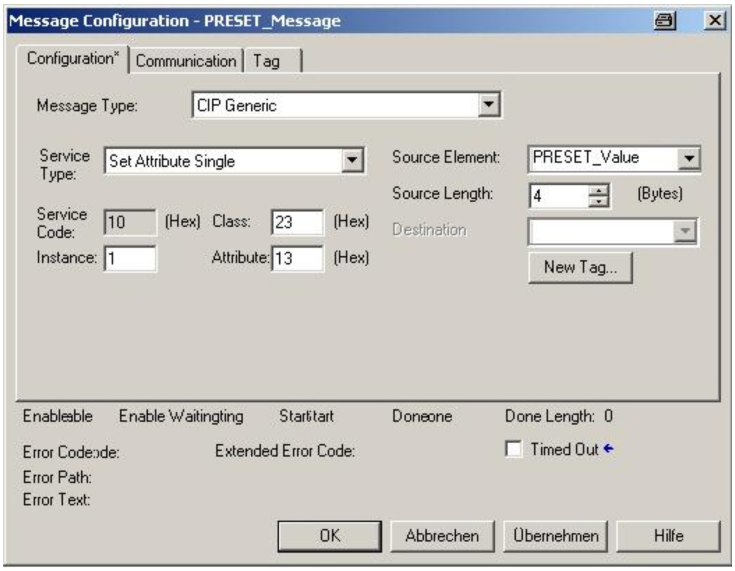
MSG

Message

Message Control PRESET_Message

Figure 96: Opening configuration dialog box for the MSG block

- ▶ For this purpose click the button with the three dots.
  The **Message Configuration** dialog box will open.



Message Configuration - PRESET_Message

Configuration* | Communication | Tag

Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex) Class: 23 (Hex)

Instance: 1 Attribute: 13 (Hex)

Source Element: PRESET_Value

Source Length: 4 (Bytes)

Destination

Enableable Enable Waitingting Starttart Doneone Done Length: 0

Error Codeode: Extended Error Code: □ Timed Out

Error Path:

Error Text:

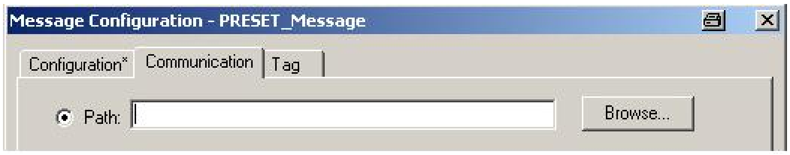
Figure 97: Configuration dialog box for the MSG block

- ▶ Configure the following parameters on the **Configuration** tab:
  - ○ **Service Type**: Set Attribute Single (see Table 18 on page 28)
  - ○ **Instance**: 1 (as only one device is connected to the control system)
  - ○ **Class**: 23(h) (Position Sensor Object, see Table 5 on page 19)
  - ○ **Attribute**: 13(h) (Preset Value, see Table 21 on page 35)
  - ○ **Source Element**: PRESET_Value
  - ○ **Source Length**: 4

NOTE

PRESET_Value is the fourth variable added. On executing the example program the preset value is taken from this variable and written to the attribute 13h of the Position Sensor Object.

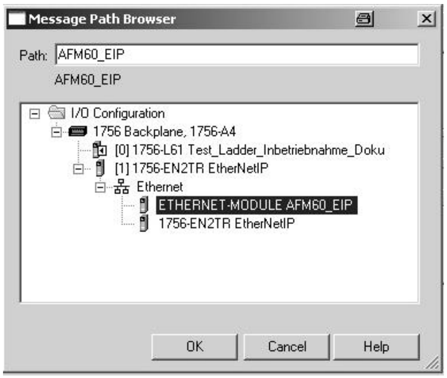
- ▶ Open the **Communication** tab.



Message Configuration - PRESET_Message

Figure 98: Communication tab

- ▶ Beside the **Path** field click the **Browse...** button. The **Message Path Browser** dialog box will open.
- ▶ Select the encoder connected.



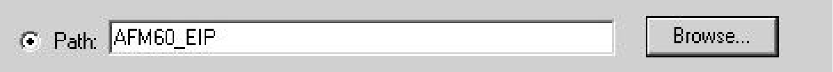
Message Path Browser

Path: AFM60_EIP

AFM60_EIP

```
⊟ I/O Configuration
  ⊟ 1756 Backplane, 1756-A4
      [0] 1756-L61 Test_Ladder_Inbetriebnahme_Doku
    ⊟ [1] 1756-EN2TR EtherNetIP
      ⊟ Ethernet
          ETHERNET-MODULE AFM60_EIP
          1756-EN2TR EtherNetIP
```

Figure 99: Selecting encoder



Path: AFM60_EIP

Figure 100: Selected encoder

The encoder is applied in the **Path** field.

- ▶ Close the **Message Path Browser** dialog box using **OK**.

Transferring program to the control system

Finally the program is transferred to the control system.

- ▶ From the **Offline** menu select the **Download** command.

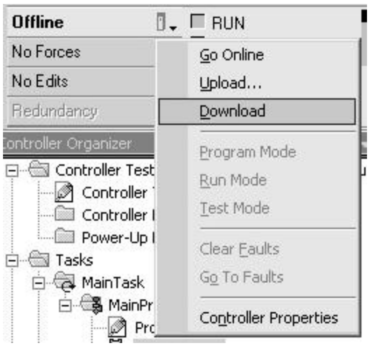
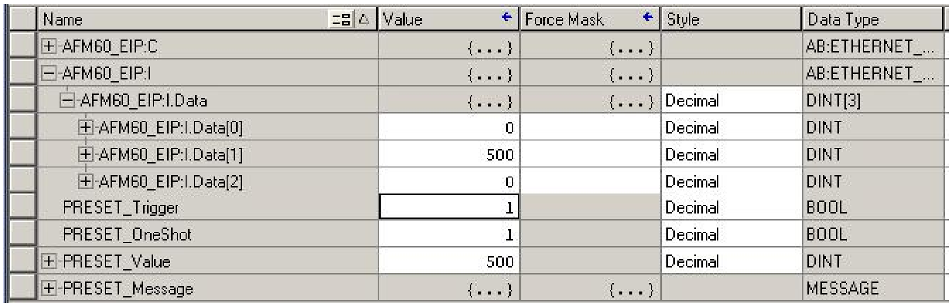


Figure 101: Transferring the program to the control system

- ▶ Accept the next message.

Testing program



Name	Value	Force Mask	Style	Data Type
⊞-AFM60_EIP:C	{...}	{...}		AB:ETHERNET_...
⊟-AFM60_EIP:I	{...}	{...}		AB:ETHERNET_...
⊟-AFM60_EIP:I.Data	{...}	{...}	Decimal	DINT[3]
⊞-AFM60_EIP:I.Data[0]	0		Decimal	DINT
⊞-AFM60_EIP:I.Data[1]	500		Decimal	DINT
⊞-AFM60_EIP:I.Data[2]	0		Decimal	DINT
PRESET_Trigger	1		Decimal	BOOL
PRESET_OneShot	1		Decimal	BOOL
⊞-PRESET_Value	500		Decimal	DINT
⊞-PRESET_Message	{...}	{...}		MESSAGE

Figure 102: Display of the preset value in PRESET_Value

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

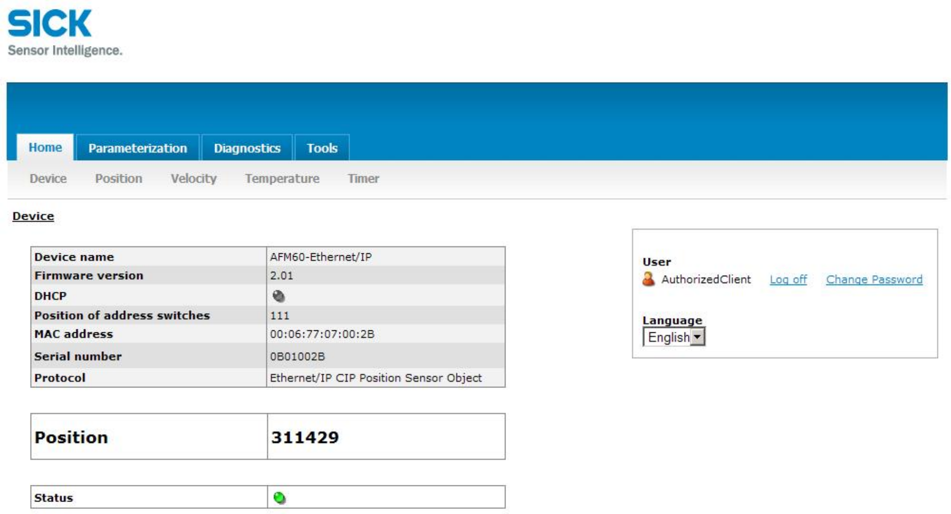
  In the position data **AFM60_EIP:I.Data[1]** the value now changes to 500.

6 Configuration with the aid of the integrated web server

A web server is integrated in the AFS60/AFM60 EtherNet/IP. Using the web server you can monitor the status of the encoder, configure the encoder parameters and undertake diagnostics.

NOTE

If you change parameters using the web server, then please pay attention to section 3.5 “Integration and configuration options” on page 36.



Device

Device name	AFM60-Ethernet/IP
Firmware version	2.01
DHCP	●
Position of address switches	111
MAC address	00:06:77:07:00:2B
Serial number	0B01002B
Protocol	Ethernet/IP CIP Position Sensor Object

Position	311429

Status	●

User
AuthorizedClient

Language
English

Figure 103: Web server user interface

Prerequisites

- 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 later, Google Chrome V38.0 and later, Firefox V33.0.2 and later.
- The IP address of the encoder must be known (see section 5.2 on page 48).

Language

The web server starts in English.

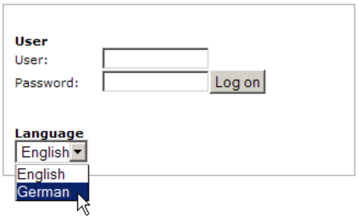


Figure 104: Selection of language

You can change the user interface language to German in the **Language** list box.

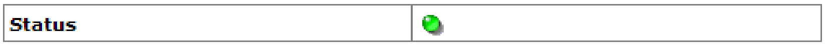
6.1 Home

NOTE

All values displayed are refreshed around once per second.

6.1.1 Device

This page lists the basic data on the encoder.



Status	🟢

Figure 105: LED symbol

An LED symbol also indicates the following status:

● **Green** Encoder in the operational status (ready for operation, no alarms, warnings or errors occurred)

☀ **Green** Incorrect scaling parameters

● **Red** The Alarm flag is set.

☀ **Red** The Warning flag is set.

You will find a detailed description of the alarms, warnings or errors that have occurred on the web server **Diagnostics** page (see section 6.3 on page 95).

6.1.2 Position

This page shows the following parameters from the Position Sensor Object (see Table 21 on page 35):

- current position value (attribute ID 0Ah)
- lower limit for the position (attribute ID 16h)
- upper limit for the position (attribute ID 17h)

You can change the limits as user “AuthorizedClient” (see section 6.2.7 on page 94).

6.1.3 Velocity

This page shows the following parameters from the Position Sensor Object (see Table 21 on page 35):

- current velocity (attribute ID 18h)
  The unit for the velocity is defined by the attributes 19h and 20h.
- lower limit for the velocity (attribute ID 1Bh)
- upper limit for the velocity (attribute ID 1Ch)

You can change the limits as user “AuthorizedClient” (see section 6.2.7 on page 94).

6.1.4 Temperature

This page shows the following parameters from the Position Sensor Object (see Table 21 on page 35):

- current temperature (attribute ID 64h)
  The temperature is indicated with ±5 ° accuracy.
- lower limit for the temperature (attribute ID 67h)
- upper limit for the temperature (attribute ID 68h)

You can change the limits as user “AuthorizedClient” (see section 6.2.7 on page 94).

6.1.5 Timer

This page shows the following parameters from the Position Sensor Object (see Table 21 on page 35):

- saved motion time in seconds (attribute ID 6Bh)
- saved operating time in seconds (attribute ID 6Ch)

You can change the limits as user “AuthorizedClient” (see section 6.2.7 on page 94).

6.2 Parameterization

With the aid of this page you can configure the encoder parameters. By configuring the parameters you can set the attributes of the Position Sensor Object (see Table 21 on page 35). The configuration options depend on whether you are logged in as a user.

After you have re-entered a parameter, press the Enter key. The parameter is saved in the volatile memory of the encoder.

NOTE

Only the parameter last changed is written to the volatile memory when you press the Enter key. If you want to change several values (e.g. the lower and the upper limit for the velocity), then press the Enter key after each data entry.

The following configuration options are available without logging in:

- overview
- units
- preset

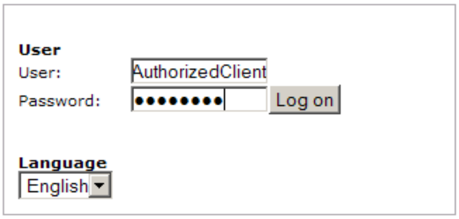
The following configuration options are available after logging in as user “AuthorizedClient”:

- scaling
- round axis functionality
- changing preset value
- limits
- reset

Login

You can login for configuration using the following access data:

- user: AuthorizedClient
- password: enc123



User

User: AuthorizedClient

Password: ••••••••

Language

English

Figure 106: Log on

Changing the password

NOTE

Change the password to prevent unauthorized access to the encoder.

- ▶ In **User** click the link **Change password**.

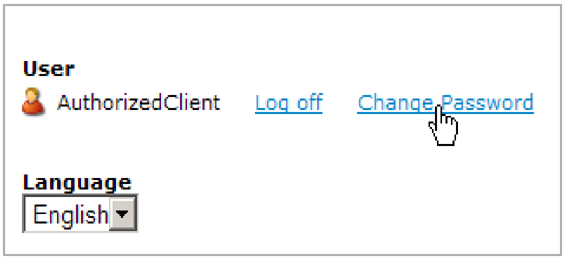


Figure 107: Changing the password

The **Change Password** dialog box will open.

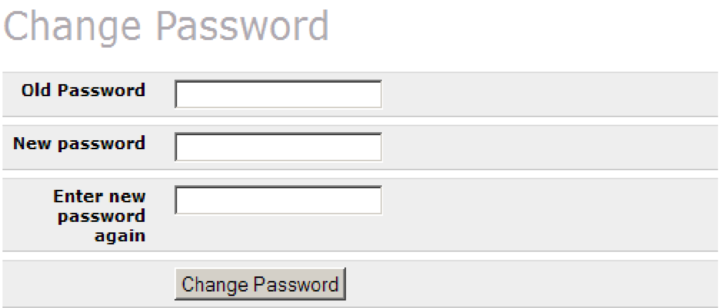


Figure 108: Changing the password

- ▶ Type the password used up to now in the **Old password** field.
- ▶ Type a new password in the **New password** field.
  Type at least 1 character and a maximum of 16 characters [17].
- ▶ Type the new password again in the **Enter new password again** field.
- ▶ Click on **Change password**.
  The new password is applied.

For technical reasons the password is transmitted unencrypted over the network. Therefore take measures to prevent password sniffing.

6.2.1 Overview

This page shows an extract from the attributes of the Position Sensor Object (see Table 21 on page 35).

- The **Current** column shows the parameters currently configured.
- The **Default** column shows the factory settings.
- The **ID hex** column shows the attribute IDs in the Position Sensor Object.

[17] All Unicode characters are permitted.

6.2.2 Units

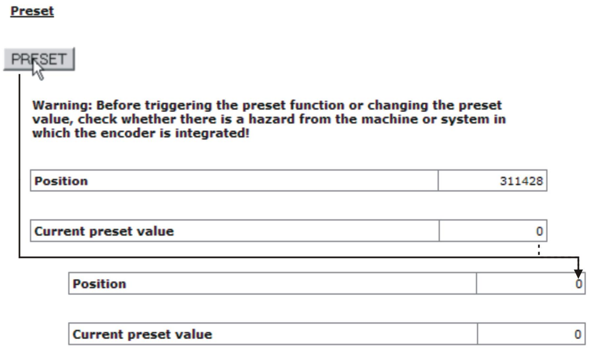
On this page you can configure the parameters for the units for direction, velocity, acceleration and temperature from the Position Sensor Object (see Table 21 on page 35).

- code sequence (attribute ID 0Ch)
  - clockwise
  - counterclockwise
- velocity 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 Triggering preset

WARNING
Before triggering the preset function, check whether there is a hazard from the machine or system in which the encoder is integrated!

This page shows the current position value for the encoder and the preset value (attribute ID 13h) from the Position Sensor Object.



Warning: Before triggering the preset function or changing the preset value, check whether there is a hazard from the machine or system in which the encoder is integrated!

Position	311428
Current preset value	0

Position	0
Current preset value	0

Figure 109: Triggering preset

▶ Click on **PRESET**.

The position value is set to the preset value.

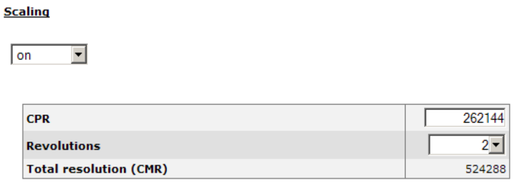
You can change the preset value as user “AuthorizedClient” (see Figure 106 on page 89).

6.2.4 Scaling

On this page you can configure the parameters for the scaling in the Position Sensor Object (see Table 21 on page 35).

- **Scaling** (attribute ID 0Eh)
  - ○ on
  - ○ off

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



Scaling

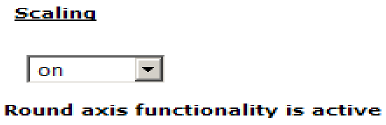
on

CPR	262144
Revolutions	2
Total resolution (CMR)	524288

Figure 110: Scaling

- **CPR**, number of steps per revolution (attribute ID 10h)
- **Revolutions**, number of revolutions of the total resolution (This is not a Position Sensor Object attribute.)

  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 **Total resolution (CMR)** field indicates the value of attribute ID 11h “Total Measuring Range, total resolution” in the Position Sensor Object (see Table 21 on page 35).



Scaling

on

Round axis functionality is active

Figure 111: Scaling with active round axis functionality

NOTE

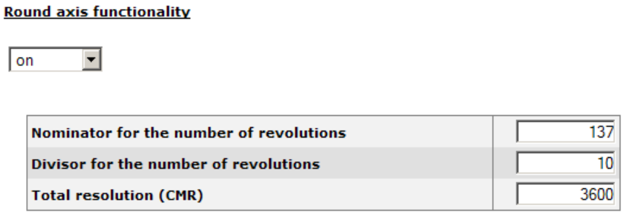
If the round axis functionality is activated, it is not possible to set any scaling.

6.2.5 Round axis functionality

You activate the round axis functionality and configure the parameters for nominator, divisor and the total resolution (see Table 21 on page 35).

- round axis functionality (attribute ID 7Dh)
  - ○ on
  - ○ off

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



Round axis functionality

on

Nominator for the number of revolutions	137
Divisor for the number of revolutions	10
Total resolution (CMR)	3600

Figure 112: Round axis functionality

- **Nominator for the number of revolutions** (attribute ID 7Eh)
- **Divisor for the number of revolutions** (attribute ID 7Fh)
- **Total resolution (CMR)** (attribute ID 11h)

The prerequisites and restrictions for the parameters are described in section 3.6.10 on page 42.

NOTE

If you activate the round axis functionality, then on the **Scaling** page the scaling is set to **on**. However, no scaling parameters are available (see Figure 111 on page 92).

6.2.6 Changing preset value

WARNING

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

As soon as you have entered the value and accepted the entry using the Enter key, the value is applied as a position value (see Figure 109 on page 91)!

On this page you can configure the preset value in the Position Sensor Object (attribute ID 13h, see Table 21 on page 35).

6.2.7 Limits

On this page you can configure the limits for the position, velocity, acceleration and temperature:

- lower limit for the position (attribute ID 16h)
- upper limit for the position (attribute ID 17h)

NOTE

Using the lower and upper limit for the position you can realize range monitoring. This is not an electronic cam.

- lower limit for the velocity (attribute ID 1Bh)
- upper limit for the velocity (attribute ID 1Ch)
- lower limit for the acceleration (attribute ID 20h)
- upper limit for the acceleration (attribute ID 21h)

If these limits are exceeded, the consequence will be the following:

- The Warning flag (attribute ID 31h) in the Position Sensor Object is set (see Table 21 on page 35).
- On the **Device** page the status LED flashes (see section 6.1.1 on page 88).
- The warning text is displayed on the **Status** page (see section 6.3.1 on page 95).

In addition other limits that are not included in the Position Sensor Object can be set:

- limit for the motion time in hours [18)]
- limit for the operating time in hours [18)]
- limit for the number of changes in the direction of rotation
- limit for the number of clockwise starts
- limit for the number of counterclockwise starts

6.2.8 Reset

On this page you can run the class service **Reset** and restart the encoder.

NOTE

After the restart the language will be reset to English and the user logged out.

Save parameters in the non-volatile memory

▶ Click on **–S–**.

The function uses the class service **Save** (service code 16h) in the Position Sensor Object.

The parameters are saved in the non-volatile memory, the encoder is restarted.

[18)] The motion time and the operating time are always calculated from the initial commissioning of the encoder. On configuring the limit, note that encoder may already have a certain amount of motion time or operating time.

Reset to the default factory settings

▶ Click on **–D–**.

The function uses the class service **Reset** (service code 05h) in the Position Sensor Object (data = 01h).

The parameters are reset to the factory settings, the encoder restarted.

Restart

▶ Click on **–R–**.

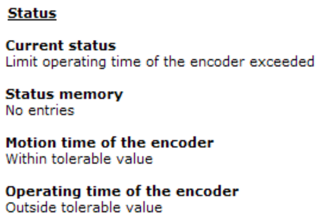
The encoder is restarted.

6.3 Diagnostics

On the diagnostics pages you will find detailed information on possible alarms, warnings and errors.

6.3.1 Status

The page shows a description of the error if a warning or an alarm has occurred.



Status

Current status
Limit operating time of the encoder exceeded

Status memory
No entries

Motion time of the encoder
Within tolerable value

Operating time of the encoder
Outside tolerable value

Figure 113: Diagnostics status

- **Current status**
  The last three messages since switching on [19] are displayed.
- **Status memory**
  The texts for warnings, alarms and errors from the fault header are displayed (see Table 30 on page 103). If a warning, alarm or error has not yet occurred, the text displayed is **No entries**.
- **Motion time of the encoder**
  Indicates whether the motion time is within the tolerated values (see section 6.2.7 on page 94).
- **Operating time of the encoder**
  Indicates whether the operating time is within the tolerated values (see section 6.2.7 on page 94).

[19] The memory is empty after switching off and on again.

6.3.2 Velocity

This page shows the following values on the velocity from the Position Sensor Object (see Table 21 on page 35):

- velocity unit (attribute ID 19h)
- current velocity (attribute ID 18h)
- highest velocity that the encoder has reached since start-up (attribute ID 6Dh)
- lower limit for the velocity (attribute ID 1Bh)
- upper limit for the velocity (attribute ID 1Ch)

6.3.3 Temperature

This page shows the following values on the temperature from the Position Sensor Object (see Table 21 on page 35):

- 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 for the temperature (attribute ID 67h)
- upper limit for the temperature (attribute ID 68h)

6.3.4 Time

This page shows the following values on the encoder motion time and operating time from the Position Sensor Object (see Table 21 on page 35):

- saved motion time in seconds (attribute ID 6Bh)
- limit for the motion time in hours (see section 6.2.7 on page 94)
- saved operating time in seconds (attribute ID 6Ch)
- limit for the operating time in hours (see section 6.2.7 on page 94)

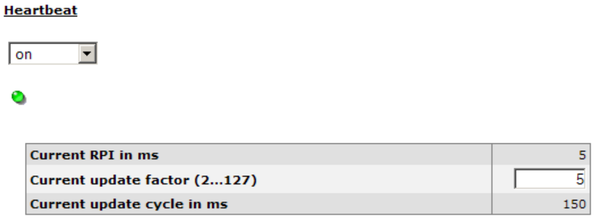
6.3.5 Cycles

This page shows the following values on the encoder cycles from the Position Sensor Object (see Table 21 on page 35):

- 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 for the number of changes in the direction of rotation (see section 6.2.7 on page 94)
- limit for the number of clockwise starts (see section 6.2.7 on page 94)
- limit for the number of counterclockwise starts (see section 6.2.7 on page 94)

6.3.6 Heartbeat

The AFS60/AFM60 EtherNet/IP supports Slave Sign of Life functionality (see section 3.6.3 on page 40).



Heartbeat

on

●

Current RPI in ms	5
Current update factor (2...127)	5
Current update cycle in ms	150

Figure 114: Heartbeat

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

An LED symbol indicates the heartbeat:

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

NOTE

As the website is refreshed every second, the change between the status cannot be displayed in real-time.

The **Current RPI in ms** column indicates the RPI.

Define the update factor in the **Current update factor (2 ... 127)** field.

The **Current update cycle in ms** column indicates the heartbeat.

6.4 Tools

6.4.1 EDS

The EDS files for integrating the encoder in the PLC are saved 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 the multiturn encoder as well as their icon.

6.4.2 Ladder routine

The configuration data are mapped between the control system and the web server with the aid of the ladder routine (see section 3.5.2 on page 36). The ladder routine is saved in the encoder.

You must download the appropriate ladder routine depending on whether you use the instance 101WS and 103WS or the instance 102WS of the Assembly Object (see Table 15 on page 23).

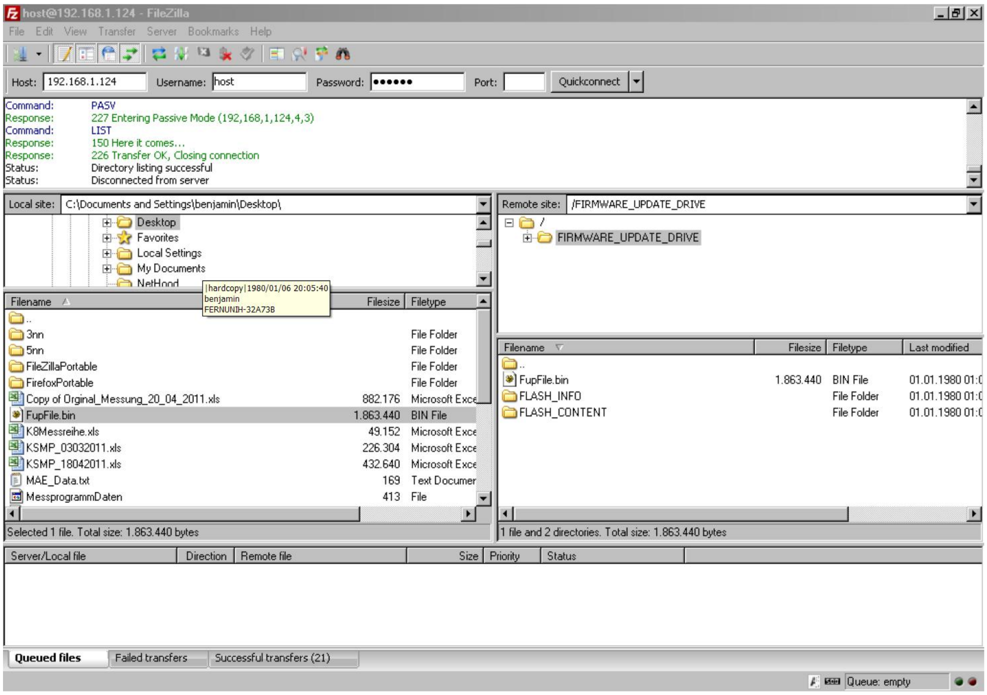
▶ Choose the ladder routine to suit the instance used. Click **Download Ladder-Routine ...** to download the file as a RAR archive.

6.4.3 Update

You can update the firmware using FTP.

▶ If you are connected to the encoder using the web server, close the web browser.
▶ Start your FTP client and enter the IP address of the encoder.
▶ Use the following login data:

- ○ user name = host
- ○ password = enc123



Host: 192.168.1.124 Username: host Password: •••••• Port:

```
Command:    PASV
Response:   227 Entering Passive Mode (192,168,1,124,4,3)
Command:    LIST
Response:   150 Here it comes...
Response:   226 Transfer OK, Closing connection
Status:     Directory listing successful
Status:     Disconnected from server
```

Local site: C:\Documents and Settings\benjamin\Desktop\

- Desktop
- Favorites
- Local Settings
- My Documents
- NetHood

|hardcopy|1980/01/06 20:05:40
benjamin
FERNUNIH-32A73B

Filename	Filesize	Filetype
..		
3nn		File Folder
5nn		File Folder
FileZillaPortable		File Folder
FirefoxPortable		File Folder
Copy of Orginal_Messung_20_04_2011.xls	882.176	Microsoft Exce
FupFile.bin	1.863.440	BIN File
K8Messreihe.xls	49.152	Microsoft Exce
KSMP_03032011.xls	226.304	Microsoft Exce
KSMP_18042011.xls	432.640	Microsoft Exce
MAE_Data.txt	169	Text Documer
MessprogrammDaten	413	File

Selected 1 file. Total size: 1.863.440 bytes

Remote site: /FIRMWARE_UPDATE_DRIVE

- /
  - FIRMWARE_UPDATE_DRIVE

Filename	Filesize	Filetype	Last modified
..			
FupFile.bin	1.863.440	BIN File	01.01.1980 01:(
FLASH_INFO		File Folder	01.01.1980 01:(
FLASH_CONTENT		File Folder	01.01.1980 01:(

1 file and 2 directories. Total size: 1.863.440 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status

Figure 115: Example for the firmware update

- ▶ Open the folder FIRMWARE_UPDATE_DRIVE.
- ▶ Copy the update file (*.bin) to this folder.

The firmware update takes approx. 3 minutes.

- During the firmware update the Encoder LED initially flashes red.
- Then the Encoder LED illuminates red.

After the firmware update the encoder restarts.

- The Encoder LED then illuminates green.

NOTE

Make sure that the encoder is continuously supplied with power during the firmware update. If the power is interrupted the encoder will either be reset to the state prior to the update or, in the worst case, will no longer respond.

6.4.4 Address switches

This page shows the possible settings for the address switches (see Table 27 on page 46).

6.4.5 Fault header information

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

6.5 Test notes

⚠ **WARNING**

Commissioning requires a thorough check by authorized personnel!

Before you operate a system equipped with the AFS60/AFM60 EtherNet/IP for the first time, make sure that the system is first checked and released by authorized personnel. Please read the notes in chapter 2 “On safety” on page 9.

7 Fault diagnosis

This chapter describes how to identify and rectify errors and malfunctions of the AFS60/AFM60 EtherNet/IP Absolute Encoder.

7.1 In the event of faults or errors

⚠ **WARNING**
Cease operation if the cause of the malfunction has not been clearly identified!

Stop the machine if you cannot clearly identify or allocate the error and if you cannot safely rectify the malfunction.

7.2 SICK STEGMANN support

If you cannot remedy an error with the help of the information provided in this chapter, please contact your local SICK STEGMANN representative.

7.3 Diagnostics

7.3.1 Error and status indications on the LEDs

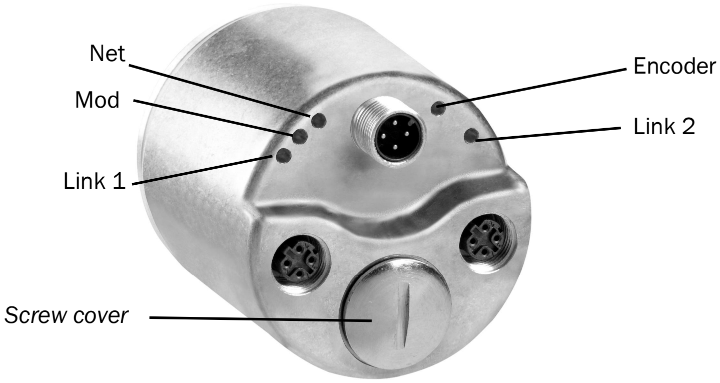


Figure 116: Position of the LEDs

Status LEDs Mod, Net and Encoder

LED Mod shows the device status, LED Net shows the status of the CIP connection and LED Encoder shows the status of the internal measuring device in the AFS60/AFM60 EtherNet/IP.

Display	Description
	LED Mod
○ Off	No operating voltage
● Green	Device in operation
☀ Green	Stand-by/device not configured, no IP address assigned
☀ Red	Warning, but device still operational or Firmware update in progress
● Red	Error, device not operational
☀ Red/green	Self-test at power-on
	LED Net
○ Off	No operating voltage or No IP address
☀ Green	No connection The device has an IP address but no CIP connection.
● Green	The device has an IP address and a CIP connection.
☀ Red	Warning, connection timeout Cleared by reset or a new connection
● Red	Error IP address has been assigned to another device already.
☀ Red/green	Self-test at power-on
	LED Encoder
○ Off	No operating voltage or No IP address
☀ Green	Warning Incorrect parameter
● Green	Device in operation
☀ Red	Warning, but device still operational or Firmware update in progress
● Red	Error Encoder error or Reboot after firmware update in progress
☀ Red/green	Self-test at power-on

Table 28: Meaning of the status LEDs Mod, Net and Encoder

Ethernet Link LEDs Link 1 and 2

The Ethernet Link LEDs Link 1 and Link 2 display the status of the physical connection on the Ethernet interface.

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

Table 29: Meaning of the LEDs Link 1 and Link 2

7.3.2 Self-test via EtherNet/IP

To check the sensors and the most important functions of the encoder, a self-test is available.

NOTE

The self-test is only allowed to be undertaken with the encoder at standstill.

The self-test can be triggered via the diagnostics bit of attribute ID 0Dh in the Position Sensor Object (see Table 21 on page 35). If an error occurs, bit 27 in the fault header is set (see Table 30 on page 103).

After 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 using implicit messages and also explicit messages.

If connections are established via the I/O assembly, the fault header can be read using the instances 101, 102 and 103 as well as the instances 101WS, 102WS and 103WS (see Table 16 on page 25).

Alarms and warnings for the encoder can be read via the Position Sensor Object (see Table 21 on page 35) with the aid of the attributes.

For errors, alarms and warnings the following applies:
Bit status = 0: no error, alarm or warning
Bit status = 1: error, alarm or warning present

Fault header

Byte	Bit	Description
0	0	Operating temperature of the microcontroller outside the permissible range
	1	Operating temperature of the encoder outside the permissible range
	2	Permissible internal LED current in the sensors exceeded
	3	Supply voltage outside the permissible range
	4	Frequency error, maximum velocity has been exceeded
	5	The upper/lower limit for the velocity configured using the attribute ID 1Bh and 1Ch has been dropped below/exceeded (see Table 21 on page 35).
	6	The upper/lower limit for the acceleration configured using the attribute IDs 20h and 21h has been dropped below/exceeded (see Table 21 on page 35).
	7	The upper/lower limit for the position configured using the attribute IDs 16h and 17h has been dropped below/exceeded (see Table 21 on page 35).
1	8	Position error (amplitude error of the singleturn measurement)
	9	Position error (amplitude error of the multiturn measurement)
	10	Position error (vector error Sin² + Cos² of the singleturn measurement)
	11	Position error (vector error Sin² + Cos² of the multiturn measurement)
	12 … 14	Reserved
	15	One parameter was changed.
2	16	Singleturn position error (error in the sensor)
	17	Multiturn position error (synchronization MA single)
	18	Multiturn position error (synchronization quad single)
	19	Multiturn position error (internal interface)
	20	Multiturn position error (FRAM)
	21	Limit for the number of changes in the direction of rotation exceeded
	22	Limit for the number of clockwise starts exceeded
	23	Limit for the number of counterclockwise starts exceeded
3	24	Memory error (EEPROM Checksumme)
	25	Memory error (EEPROM IRQ)
	26	Error on start-up
	27	Error during self-test
	28	Limit for the motion time of the encoder has been exceeded
	29	So-called “Sanity-check flag”. The flag is set if the encoder has detected an incorrect velocity or a position error. Is reset on switching back on.
	30	Slave Sign of Life. Active, if attribute ID 0Dh is set (see Table 21 on page 35). The bit changes its value at the update cycle configured.
	31	Limit for the operating time of the encoder has been exceeded

Table 30: Fault header

Alarms

If, for example, the internal self-test detects that the position value has been incorrectly calculated or an incorrect configuration value has been transferred to the encoder, the alarm flag is set, (attribute 46, see Table 21 on page 35).

WARNING

It is imperative to evaluate the alarms in your application!

In case of a serious error, incorrect position values may be output. This change could cause an unexpected movement that may result in a hazard for persons or damage to the system or other items.

● Red

In addition the Mod LED illuminates red continuously.

The alarm type is coded in a bit field of attributes 44 and 45.

Bit	Description
0	Position error
1	Error during self-test
2 … 11	Reserved
12	Incorrect checksum (vendor specific)
4	Error on system start-up (vendor specific)
14 … 15	Reserved

Table 31: Alarms

Warnings

If, for example, the velocity or temperature drop below/exceed the limit values, the warning flag is set (attribute ID 31h, see Table 21 on page 35).

☀ Red

In addition the Mod LED flashes red.

The warning type is coded in a bit field of attribute IDs 2Fh and 30h.

NOTE

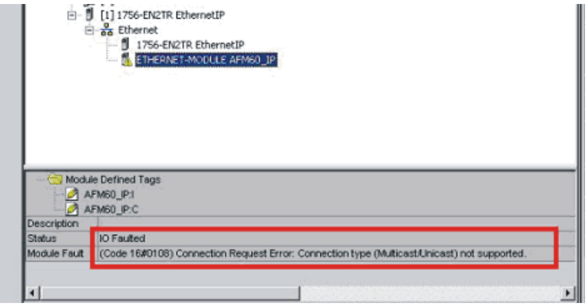
The position value will continue to be correctly calculated, the encoder is therefore still ready for operation.

Bit	Description
0	Maximum velocity exceeded
1	Permissible internal LED current in the sensors exceeded
2 ... 5	Not supported
6	The lower limit for the velocity configured with attribute 1Bh has been dropped below.
7	The upper limit for the velocity configured with attribute 1Ch has been exceeded.
8	The lower limit for the acceleration configured with attribute 20h has been dropped below.
9	The upper limit for the acceleration configured with attribute 21h has been exceeded.
10	The lower/upper limit for the position configured with attribute 16h and 17h has been dropped below/exceeded.
11 ... 12	Reserved
13 [20]	The lower/upper limit for the temperature configured with attribute 67h and 68h has been dropped below/exceeded.
14 [20]	The operating voltage has dropped below/exceeded the minimum/maximum operating voltage.

Table 32: Warnings

7.3.4 Error messages from the Allen-Bradley control system

If the encoder is integrated into an Allen-Bradley control system, some error messages may occur that have message text from which the cause is not immediately obvious.



Description

Status: IO Faulted

Module Fault: (Code 16#0108) Connection Request Error: Connection type (Multicast/Unicast) not supported.

Figure 117: Example of an error message in RSLogix

[20] Vendor specific warning.

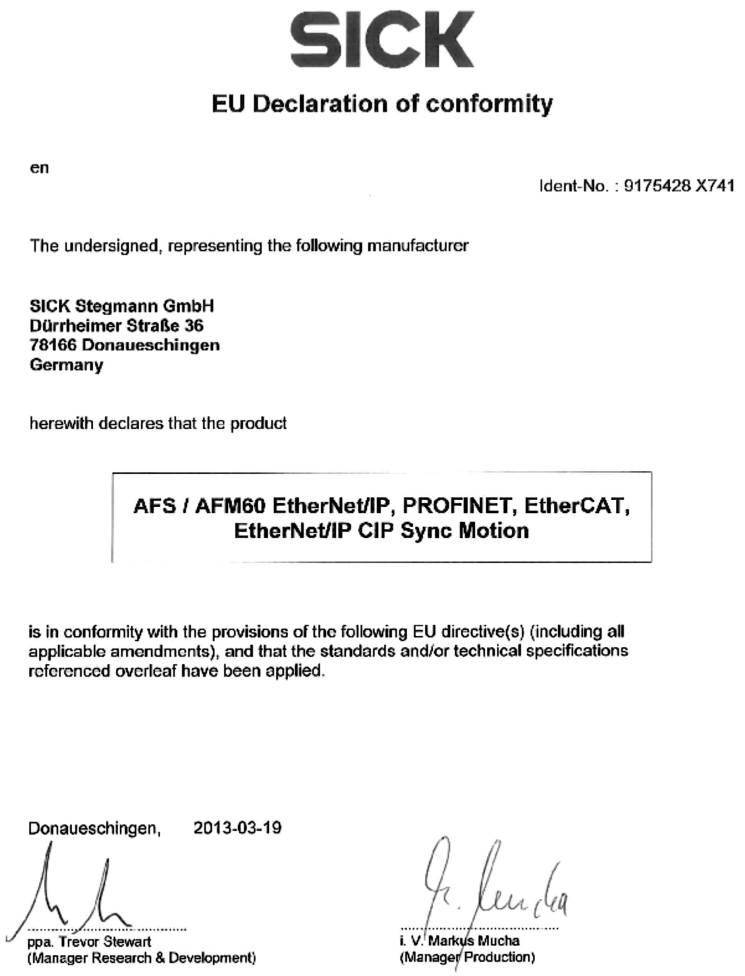
The following error messages stem from the RSLogix 5000 software.

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 yes, check whether the configuration data are correctly and fully configured in this assembly (see Figure 31 on page 55).
16#0114	Electronic Keying Mismatched: Electronic keying product code and/or vendor ID mismatched.	▶ Check whether the wrong EDS file has been selected (e.g. singleturn instead of multiturn or vice versa, see section 5.5 on page 56).
16#0127	Connection Request Error: Invalid output size.	▶ Check whether the correct communication format for the control system is used. The default value in the control system is “Data-DINT”. The encoder requires the communication format: “Input Data-DINT”.
16#0204	Connection Request Error: Connection timed out.	▶ Check the supply voltage on the encoder. ▶ Check the Ethernet cables for the encoder for open circuit. ▶ Check whether the IP address of the encoder matches the IP address saved in the control system. Possible causes: ○ The address switches are not engaged correctly (see Figure 18 on page 46). ○ The encoder has lost the IP address assigned to it after switching back on (see section 5.2.3 on page 50).

Table 33: Error messages from the RSLogix 5000 software

8 Annex

8.1 EU declaration of conformity



SICK

EU Declaration of conformity

en

Ident-No. : 9175428 X741

The undersigned, representing the following manufacturer

SICK Stegmann GmbH
Dürrheimer Straße 36
78166 Donaueschingen
Germany

herewith declares that the product

AFS / AFM60 EtherNet/IP, PROFINET, EtherCAT,
EtherNet/IP CIP Sync Motion

is in conformity with the provisions of the following EU directive(s) (including all applicable amendments), and that the standards and/or technical specifications referenced overleaf have been applied.

Donaueschingen, 2013-03-19

ppa. Trevor Stewart
(Manager Research & Development)

i. V. Markus Mucha
(Manager Production)

Figure 118: EU declaration of conformity

NOTE

You can obtain the complete EU declaration of conformity via the SICK homepage on the Internet at: www.sick.com

9 List of figures

10 List of tables

8014213/YFU7/2015-08-03/en · REIPA/XX (2015-08) · USmod 4c int44

Australia
Phone +61 3 9457 0600
1800 334 802 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0)22 36 62 28 8-0
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 marketing@sick.com.br

Canada
Phone +1 905 771 14 44
E-Mail information@sick.com

Czech Republic
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

Chile
Phone +56 2 2274 7430
E-Mail info@schadler.com

China
Phone +86 4000 121 000
E-Mail info.china@sick.net.cn

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

Finland
Phone +358-9-2515 800
E-Mail sick@sick.fi

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

Gemany
Phone +49 211 5301-301
E-Mail info@sick.de

Great Britain
Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

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

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

India
Phone +91–22–4033 8333
E-Mail info@sick-india.com

Israel
Phone +972-4-6881000
E-Mail info@sick-sensors.com

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

Japan
Phone +81 (0)3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603 808070425
E-Mail enquiry.my@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 22 837 40 50
E-Mail info@sick.pl

Romania
Phone +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7-495-775-05-30
E-Mail info@sick.ru

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

Slovakia
Phone +421 482 901201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 (0)1-47 69 990
E-Mail office@sick.si

South Africa
Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321
E-Mail info@sickkorea.net

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 2645 0009
E-Mail tawiwat@sicksgp.com.sg

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 info@sick.ae

USA/Mexico
Phone +1(952) 941-6780
1 (800) 325-7425 – tollfree
E-Mail info@sick.com

Vietnam
Phone +84 8 62920204
E-Mail Ngo.Duy.Linh@sicksgp.com.sg

More representatives and agencies
at **www.sick.com**

SICK
Sensor Intelligence.